Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Report on Case Studies

of

"DHIS2, openMRS, openIMIS, and FHIR HL7"

[Course Code: COMP 401]

[For the partial fulfillment of 4th year/1st Semester in Computer Engineering]

Submitted by

Sabin Thapa

Roll No.: 54

Registration No.: 024358-18

Submitted to

Rabindra Bista, Ph.D.

Associate Professor

Department of Computer Science and Engineering

Submission Date: November 22, 2022

Table of Contents

Introduction

In this era of technology, software tools are yet another new technology competing for the attention of health personnel. Medical software is rapidly evolving from a record-keeping tool to a communication system and plays the role of a medical device or a clinical service. Healthcare is not an easy subject to cover from just about any angle. One of the issues with such software is the strict privacy laws that govern the industry, which makes it challenging for healthcare software to not just be created but maintained over time. Since Electronic Medical Record (EMR) and Electronic Health Record (EHR) have been adopted in our healthcare world, most healthcare providers wish to get software that is less expensive, flexible, easy to use, and certified. For this reason, there exists a number of free and open-source software for EMR. However, it is not easy to choose one of the available open-source software, especially when the area of exploration is healthcare.

Among the various open-source software available for use in healthcare, *OpenMRS, DHIS2, openIMIS, and FHIR HL7* are some very good options. These open-source software reduce disparities because they are, almost without exception, free and accessible to all, domestically and around the world. Health is one of the most important sectors to determine how well a nation has progressed. The health status of the citizens of the country determines its position in the global context. Health data, therefore, is very important. In developing and underdeveloped countries, health data are yet to be digitized – people have to carry all the documents related to hospital visits, medical prescriptions, medical reports, etc. By some years, the document collected becomes so huge, it becomes impossible to find a specific document and/or to carry the document to every hospital one visits. Therefore, digitizing the healthcare world is one of the most important factors every nation should look up to.

In this case study, the open-source tools, ***DHIS2, openMRS, openIMIS****, and* ***FHIR HL7*** are discussed. This case study focuses on the use cases of these software tools and their significance in the healthcare world. Being able to make use of these tools that are freely available would help yield a product that would benefit not only a health institution but the entire country itself, which would ultimately benefit the entire world. Therefore, the study of these tools is likely to help us gain insights into them as well as help us understand their significance and the ways to use and implement them.

Chapter 1: DHIS2

DHIS2 stands for District Health Information System 2. It is a tool for collecting, validating, analyzing, and presenting aggregate and patient-based statistical data, tailored to integrated health information management activities. Rather than a pre-configured database application, it is a generic tool with an open meta-data model and a flexible user interface that allows the user to design the contents of a specific information system without the need for programming. DHIS2 is a modular web-based software package with free, open-source Java frameworks (*What Is DHIS2?*, 2022).

DHIS2 is a free and open-source health management data platform developed by the Health Information Systems Program (HISP) and it is supported by the University of Oslo's Department of Informatics. It is a web-based platform most commonly used as a health management information system (HMIS). It is currently being used by multiple organizations, including the European Union, and other governments worldwide. On a national scale, a total of 73 low and middle-income countries are deploying DHIS2. Since DHIS2's release in 2006, NGOs and national governments in more than 100 countries have deployed DHIS2 for health-related projects which include patient health monitoring, improving disease surveillance, and speeding up health data access (*About DHIS2*, n.d.).

1.1. Objective

DHIS2 is mainly used to aggregate statistical data collection, validation, analysis, management, and presentation of patient-based statistical data. In addition, DHIS2 has the following objectives:

- To monitor patient health, improve disease surveillance, map disease outbreaks, and speed up health data access for health facilities and government organizations.
- To provide information for management and performance improvement of primary and secondary healthcare services provided by the public health sector at the district level.
- To provide a comprehensive data management solution based on data warehousing principles and a modular structure that can easily be customized to the different requirements of management information systems, supporting analysis at different levels of the organizational hierarchy.
- To provide a data analytics and management platform that boasts great visualization features and the ability to create analysis from live data in seconds.

1.2. Features

DHIS2 is the world's leading Health Management Information System and it includes features for monitoring and following up with individual people or entities over time – and online and offline data entry via the DHIS2 web portal, mobile Android app, SMS, or direct import. Some of the key features of DHIS2 are discussed below:

1. **Analytics and Data Management**

   With a flexible data model that has been field-tested for more than 20 years, DHIS2 lets us manage and analyze aggregate and individual-level data. It is customizable to meet a variety of use cases and can be configured as a data warehouse to make use of data from a variety of sources. With advanced features for dynamic data visualization, like maps, charts, pivot tables, and dashboards, it is very easy to bring meaning to the data.

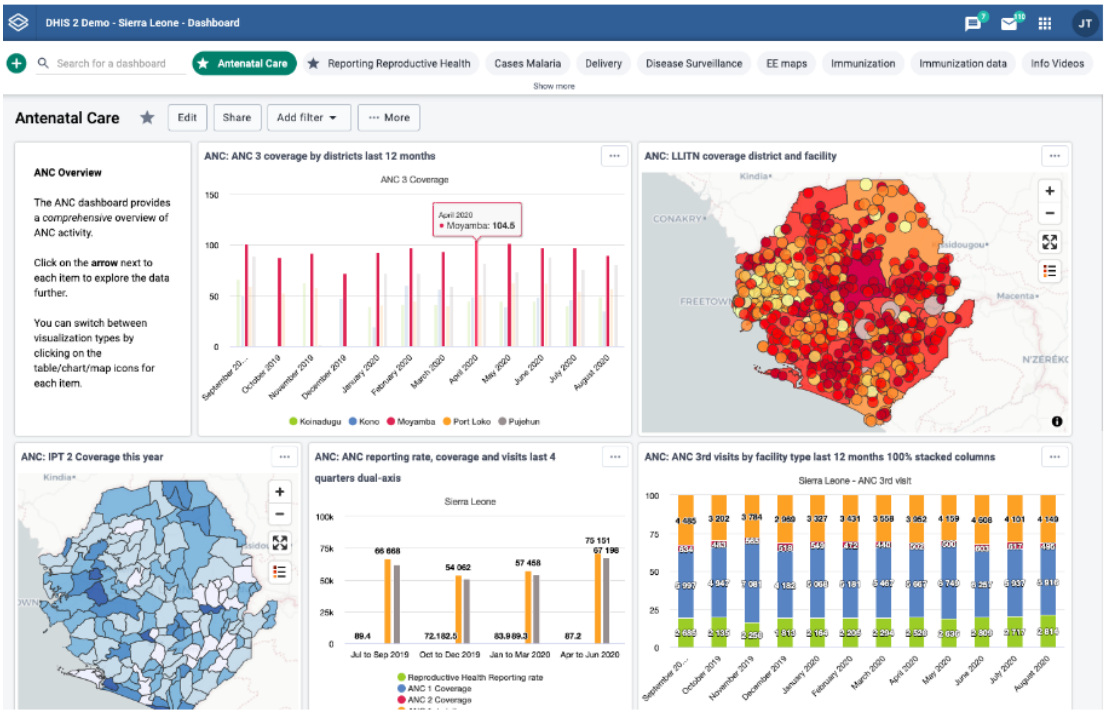

Figure 1: DHIS2 dashboard showing data analytics

The figure above is the Dashboard of the DHIS2 application. We can see that data analysis can be done through various visualization techniques like graphs, charts, plots, and maps.

2. **Individual Data Records through Tracker**

   DHIS2 enables us to collect, manage and analyze transactional, case-based data records. It can be used to store information about individuals and track these persons over time using a flexible set of identifiers. For example, we can use DHIS2 to collect and share essential clinical health data records across multiple facilities.

3. **Mobile Data Entry through Android**

   Through the DHIS2 Android App, we can expand the reach of DHIS2. The app lets us register cases, events, and personal information, track individuals, conduct surveys, and capture aggregate data. Data collection can also be done from locations where there's no access to the Internet.

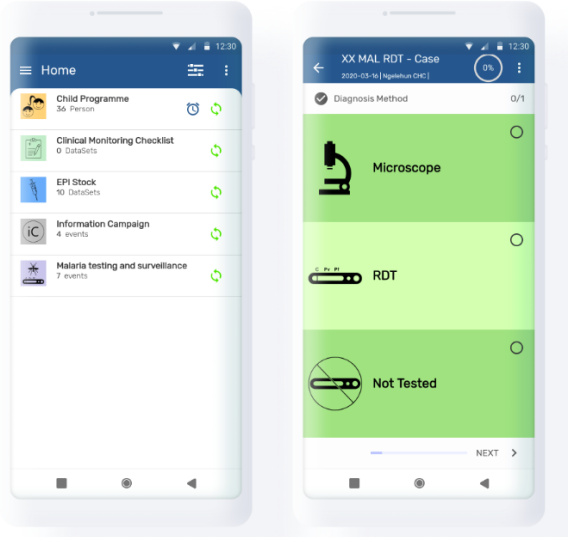

Figure 2: DHIS2 mobile application

4. **Local/Cloud Hosting**

   As DHIS2 is a FOSS, it can be installed on our servers for free. A managed DHIS2 instance (*DHIS2 Cloud Hosting*, n.d.) takes care of the backup, security, monitoring, and high-speed connectivity aspects of the deployment and allows us to focus on the information system itself.

5. **Open Source**

   Released under the liberal BSD-3-Clause license, DHIS2 is free and open-source software. It is developed in Java and runs on any platform with a Java Runtime Environment, JRE8 installed. Further perks of DHIS2 being an open-source software are discussed later in the study.

6. **Internationalized**

   DHIS2 is not only limited to the English language but the UI comes fully translated into 8 languages. Moreover, it lets us translate the database content into as many languages as we like. Switching between languages for each user is very easy.

7. **Highly scalable**

   Using only a single, standard web server, DHIS lets us have thousands of concurrent users and hundreds of millions of data records. A large number of countries are using DHIS2 as a national health information system. With DHIS2, data can be analyzed and explored to get answers within a tenth of a second.

8. **Integration and Interoperability**

   DHIS2 features its own format for meta-data and data exchange called DXF 2 as well as the ADX standard which help in system integration and interoperability. Most parts of the system can be accessed through the extensive REST-based Web API, making interoperability with third-party clients like Android apps, Web portals, and other information systems easy.

9. **Documentation and community support**

   Comprehensive guidance for implementing, configuring, maintaining, and using DHIS2 systems, as well as developing custom applications within the DHIS2 platform is provided by DHIS2. This helps users follow along with the documentation to get their work done. In addition to this, a global network of DHIS2 experts is taking part in the development and deployment of the DHIS2 by posting and answering technical questions.

1.3. Open Source Support

DHIS2 is a free and open-source health management data platform that boasts data warehousing, visualization features, and the possibility for data users and policymakers to generate analysis from live data in real time. The developers of DHIS2 are located all around the world and its development is coordinated by HISP UiO.

DHIS2 is released as free and open-source software under the BSD 3-clause license which means that the application can be used free of charge, the code is accessible to everyone and users may even modify and redistribute the software as per the requirements (Landley, n.d.). However, there is one restriction, i.e., the copyright notice must be maintained in the source code, and neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

The entire codebase of DHIS2 can be found on GitHub. The code for the DHIS2 core server, documentation, UI, releases, and app platform are all provided by the DHIS2 GitHub organization. This is a major plus point in the open source field as anyone with a GitHub account can view the entire source code of DHIS2 and might as well contribute to the code by forking the repository, making necessary changes, and creating a Pull Request (PR). The DHIS2 core is written in Java and it contains the service layer and Web API. Currently, there are around 83 contributors to the DHIS2-core repository but any open-source enthusiast is free to contribute.

DHIS2 not only lets open-source enthusiasts contribute to its repository, but since it is open-source software, it can also be used in other personal/commercial projects by respecting the license and giving the copyrights to the owner. It represents a community of developers, implementers, users, and researchers who constitute the Health Information Systems Programme (HISP) research and development network. Multiple NGOs, national governments, and universities are using DHIS2 free of cost for their research and/or health data management.

DHIS2 supports both web and mobile solutions, including SMS, plain HTML, and Java options for feature phones as well as web-based solutions with offline support for smartphones. Along with that, the rich user interface of DHIS2 has been fully translated into eight languages: English, Chinese, Spanish, French, Russian, Portuguese, Vietnamese, and Tajik, so development is easier and developers can personally translate database content into any number of languages.

DHIS2, being a **Free and Open Source Software (FOSS)** has a potentially high range of benefits within the public health sector including

- Lower costs as it's not necessary to pay for prohibited license costs.
- Easier access to source code to enable integration and interoperability. Two or more systems communicating metadata and data are becoming increasingly important in the health sector. DHIS2 helps to achieve this interoperability between different software applications.
- There is a need for the user to have the freedom to make the changes as per the user requirement, given the information needs for the health sector are constantly changing and evolving. This is often limited in proprietary systems (systems belonging to an enterprise).
- FOSS applications like DHIS2 are supported by a global network of developers, and thus have access to cutting-edge research and development knowledge.

1.4. Target Stakeholders

DHIS2 is most commonly used as a Health Management Information System (HMIS) to aggregate statistical data collection, validation, analysis, management, and presentation. The key stakeholders may include but are not limited to the following:

1. Ministry of Health
2. District staff
3. UNICEF staff
4. Development and implementing partners
5. Health institutions and organizations

Apart from the stakeholders listed above, DHIS2 can be used by anyone for health data management. Any software project aiming to digitize the health data collection and storage process can make use of DHIS2 tools to help ease the process. For instance, hospitals in Nepal can make use of DHIS2 to store and manage patient health information so that the information is digitally available and the paperwork hassle is reduced significantly. A project like this would require a huge budget and support from the government, but if we could achieve this, then the burden of storing and carrying medical reports from decades can be solved. DHIS2, therefore, encourages FOSS lovers to make efficient use of its features to help expand the global health domain and make health data easily accessible to everyone.

1.5. Dependencies

DHIS2 has a modular and layered architecture with a strong and open Application Programming Interface (API) i.e., DHIS2 essentially serves as a data warehouse with more than 60 native applications that pull or push data stored in the warehouse to perform different functions such as data quality checks. It is important to understand its dependencies like the frontend, backend, database used, and the API it provides. These dependencies are discussed in the following section:

1. **Frontend: W3 standard for HTML and CSS**

   DHIS2 is written according to the W3C (World Wide Web Consortium) standard for HTML and CSS and runs on any standard-compliant web browser like Google Chrome, Firefox, Safari, and Edge. As Google Chrome provides excellent performance for Javascript-intensive applications like DHIS2., the DHIS team recommends Google Chrome for use.

2. **Backend: Java**

   DHIS2 is written in Java Programming Language. Java applications run on any platform where exists a Java runtime environment - in practice all popular platforms including Windows, Linux, macOS, etc. DHIS2 is a web application based on standard Java technology. This means that the application runs on any Java-enabled web server or servlet container, and can be accessed via a web browser over the Internet. DHIS2 can be deployed on a national online server or hosted in the cloud, as well as deployed in an intranet offline setting.

3. **DBMS: Postgres RDBMS**

   DHIS2 runs on the open-source PostgreSQL Relational Database Management System (RDBMS). PostgreSQL is a powerful, open-source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

4. **API: REST Web API**

   DHIS2 features a rich RESTful Web API, allowing for apps to be built using purely Web technologies such as Javascript, CSS, and HTML5. A REST API is an application programming interface that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services (*Topics Understanding APIs What Is a REST API?*, 2020).

1.6. Architecture

DHIS2 has a three-layer architecture. The presentation layer is web-based and the system can be used online as well as stand-alone. The overall architecture is shown below:

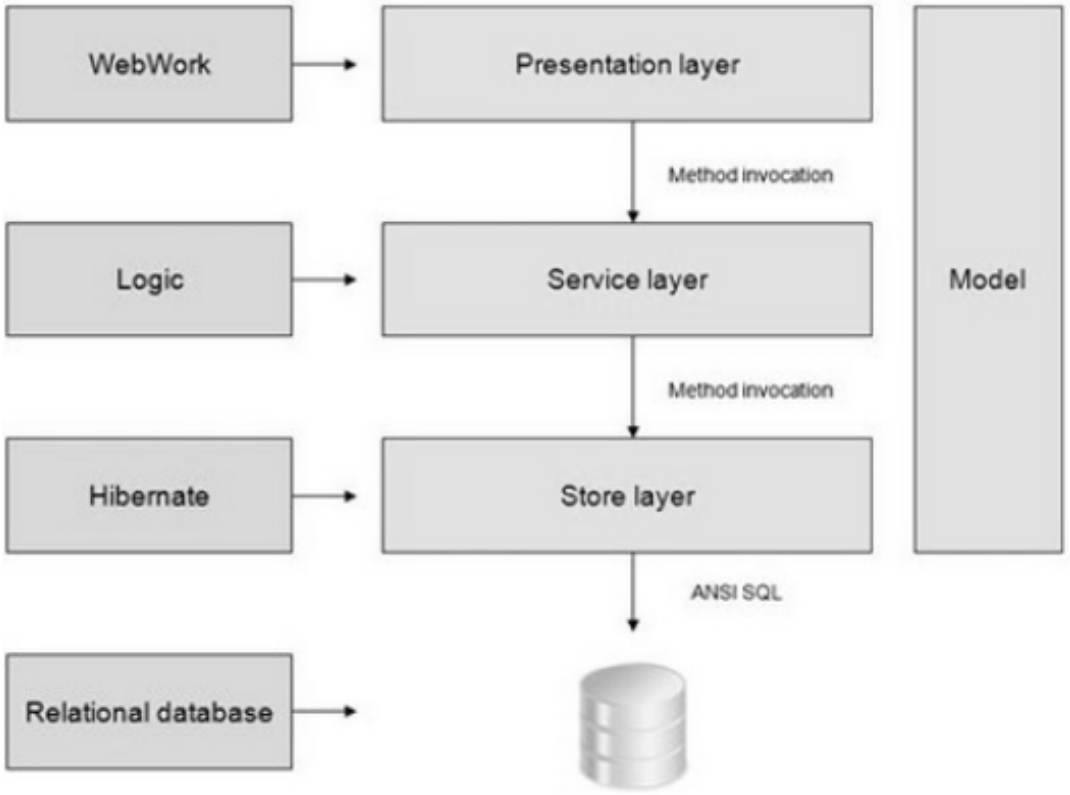

Figure 3: DHIS2 Technical Architecture

The technical architecture of DHIS2 is shown in the figure. It consists of the following three layers:

1. Presentation layer
2. Service layer
3. Store layer

It makes use of ANSI SQL as a relational database. The DHIS2 is a routine data-based health information system that allows for data capture, aggregation, analysis, and reporting of data. Using this three-layered architecture, all the activities listed above are performed.

1.7. Information Cycle

The information cycle of DHIS2 pictorially depicts the different components, stages, and processes through which the data is collected, checked for quality, processed, analyzed, and used. The information cycle is presented in the picture below:

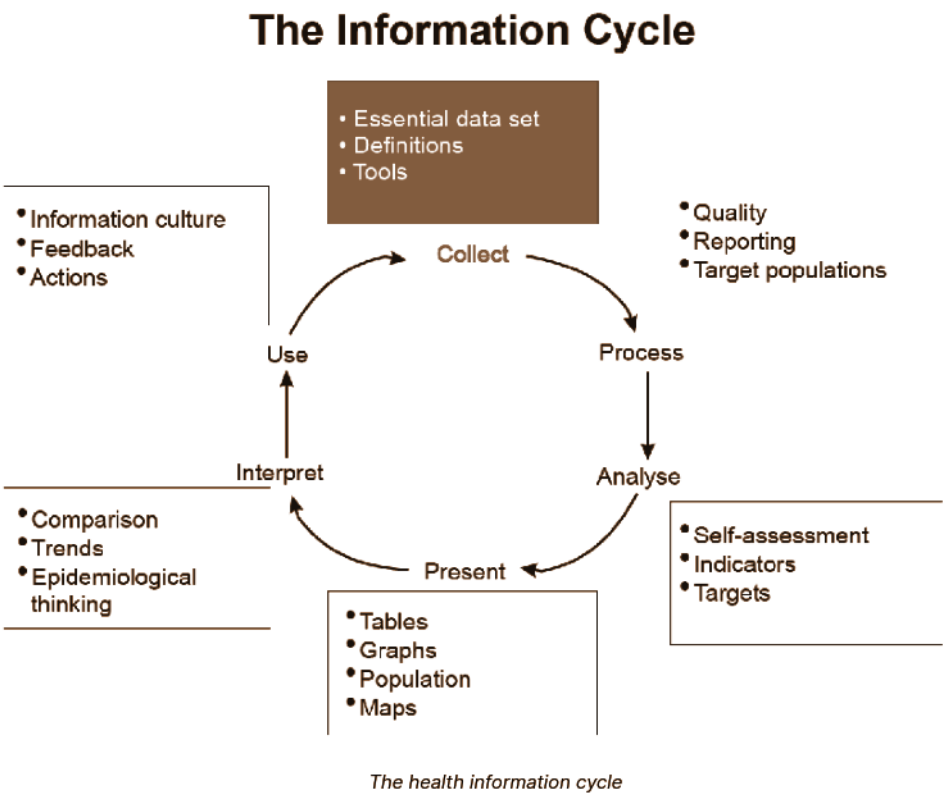

Figure 4: DHIS2 information cycle

The following are the features of the information cycle:

1. Data collection.
2. Quality checks.
3. Data access at multiple levels.
4. Reporting.
5. Analysis through graphs and maps.
6. Comparison across time (for example, last month) and space (for example, across districts).
7. View trends by displaying data in time series.

The first step of DHIS2 is data collection or data entry. It serves as a data collection, recording, and compilation tool, and all data can be entered into it. The next step is used to increase data quality. First, at the data entry point, a check can be made to see if data falls within acceptable

threshold values for any particular data element. With the help of such checking, typing errors can be minimized at the time of data entry. Further, various validation rules can be defined by the user, and DHIS2 can run the data through the validation rules to identify violations. Such checks help to ensure that data entered into the system is of good quality from the start. After the data has been entered and verified, DHIS2 can help make different kinds of reports like routine reports that can be predefined, so that all those reports that need to be routinely generated can be done with a click of a button. Moreover, DHIS2 can also help in the generation of analytical reports through comparisons. Similarly, graphs, maps, reports, and health profiles can also be produced by DHIS2 which are routinely produced, analyzed, and acted upon by health managers.

Chapter 2: OpenMRS

OpenMRS stands for Open Medical Record System. It is a collaborative open-source project to develop software to support the delivery of health care in developing countries. Founded on the principle of openness and sharing ideas, software, and strategies for deployment and use, the system is designed to be usable in very resource-poor environments and can be modified with the addition of new data items, forms, and reports without the need for programming. It is both – software and a community. As a software, it serves as an Electronic Medical Record System (EMRS) originally designed for developing countries and through its open-source community, it has grown into a medical informatics platform used on every continent.

OpenMRS is a Java-based, web-based electronic medical record system that started from a simple data model wrapped into an API. A web application was then built to use the API. The OpenMRS API works like a black box hiding the complexities of the data model beneath it and ensuring that applications and modules using the API work with a similar set of business rules for managing the electronic record system data.

OpenMRS is a common platform maintained by the OpenMRS community. It is the foundation for implementations that touch and help millions of patients throughout the world. It gives implementers the freedom to focus on creating innovative solutions that respond to changing local needs.

2.1. Objective

OpenMRS is mainly used by developing nations for electronic record keeping. The objectives of OpenMRS are discussed below:

- To improve healthcare delivery in resource-constrained environments by coordinating a global community that creates and sustains a robust scalable, user-driven, and open-source medical record platform.
- To provide a platform that countries and implementers use to create a customized EMR system in response to needs on the ground.
- To help mitigate the impact of COVID-19 on high-risk populations in countries with strained resources and a higher burden of disease.

2.2. Features

OpenMRS is a free, open-source health informatics software platform that allows users to self-customize an electronic medical record system without needing to know a programming language. The modular platform is built around a scalable conceptual database structure, with the advantage of being independent of any particular type of medical data or data collection form.

Some of the features of OpenMRS are discussed below:

1. It provides a patient repository that helps to maintain patient information including patient demographics, orders, clinical observations, and data.
2. It includes a concept dictionary at the core of the application which acts as a storage medium for all general questions, procedures, tests and diagnoses, and drugs-related information. It defines the name, code, and appropriate attributes for any observations or data collected.
3. It provides add-on modules for patient registration, id generation, and allocation services.
4. It provides features such as workflow management, cohort management, and document management.
5. It provides a feature to electronically import and export data so manual data entry hassling is mitigated.
6. It supports open standards for medical data exchange including HL7, FHIR, LOINC, and IXF.
7. It has multi-language support.

2.3. Hardware/Software Requirements

Based on the intended size of our implementation, the hardware size will vary. For example, for small implementations, OpenMRS can be run on just about any desktop or laptop. But for 100s of patients, the requirement is a 1 GHz processor or better, 256 MB memory or more, and 40GB hard drive or larger (Downey, 2013).

The additional requirements include

- A web browser (preferably Mozilla Firefox)
- Java Runtime Environment, JRE 1.6
- Tomcat 6.0.29 (An open-source web server and servlet container)
- MySQL latest stable release

2.4. Open Source Support

OpenMRS is also a FOSS i.e., it is completely free, open-source, community-driven software with no hidden costs. The OpenMRS community includes developers, implementers, and users from multiple countries who collaborate through mailing lists, IRC, and annual conferences. Each year, OpenMRS hosts its annual Implementers' Conference which is a global meet-up of developers and implementers hosted by different countries where OpenMRS is being used on a national scale.

OpenMRS is based on the principles of openness and sharing ideas, software, and strategies for deployment and use. Being a FOSS software, it has all the benefits that open-source software provides such as:

- Lower costs as it's not necessary to pay for prohibited license costs.
- Easier access to source code to enable integration and interoperability.
- There is a need for the user to have the freedom to make the changes as per the user requirement, given the information needs for the health sector are constantly changing and evolving.
- FOSS applications like OpenMRS are supported by a global network of developers, and thus have access to cutting-edge research and development knowledge.

2.5. Target Stakeholders

OpenMRS is a patient care system used in many developing countries. It houses more than 12.5 million patients' medical records, in over 5,000 health facilities across more than 65 countries in the world (*OpenMRS*, n.d.).

The target stakeholders are the countries that already experience constrained resources and a higher burden of disease. OpenMRS is well positioned to support these countries to respond to a pandemic by expanding their capacity through better scientific-based information. It is now used in clinics in Argentina, Botswana, Cambodia, Congo, Ethiopia, Gabon, Ghana, Haiti, Honduras, India, Indonesia, Kenya, Lesotho, Malawi, Malaysia, Mali, Mozambique, Nepal, Nicaragua, Nigeria, Pakistan, Peru, Philippines, Rwanda, Senegal, South Africa, Sri Lanka, Tanzania, The Gambia, Uganda, United States, Zanzibar, Zimbabwe, and many other places (Mamlin et al., 2021).

OpenMRS is used for treating the millions of HIV/AIDS and Tuberculosis patients in the developing world. So, such countries are also the target stakeholders. Besides, health institutions, the Ministry of Health, UNICEF & district staff are the other stakeholders.

2.6. Design

OpenMRS is based on a "concept dictionary" that describes all the data items that can be stored in the system such as clinical findings, laboratory test results or socio-economic data. With this approach, there's no need to modify the database structure to add new diseases and it facilitates sharing of data dictionaries between projects and sites. Because of its modular construction, we can program new functions without modifying the core code. Although OpenMRS is web-based. It can be deployed on a single laptop or a large server and it runs on Windows, Linux, or Mac OS X. Some other features of OpenMRS are discussed below:

1. It is programmed in Java programming language.
2. It is built on the MySQL database but uses Hibernate (which allows it to be ported to other databases.)
3. It provides access to between-release code through "Continuous Deployment".
4. It can be integrated with SMS messaging.
5. It has an atlas module, which gives information on all OpenMRS facilities using a visual map.

2.7. Architecture

OpenMRS is based on a modular architecture that consists of a core application and optional modules which provide additional functionality to the core workflows. The main architectural components of OpenMRS are shown below:

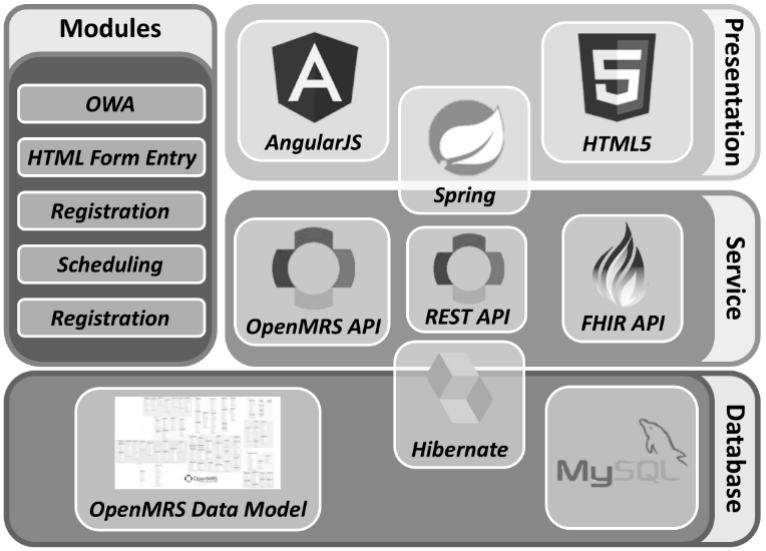

Figure: OpenMRS Architecture

There are different levels in the code architecture in the OpenMRS framework and modules. The source code is divided into three main segments:

1. The User Interface (Presentation layer)
2. The Service Layer
3. The Data Access Layer

With the help of this layering, various system responsibilities are isolated from one another, to improve both system development and maintenance. These layers are discussed below:

1. **User Interface layer**

   The User Interface layer for the legacy application is built upon Spring MVC, Direct Web Remoting (DWR), JSP, and JavaScript. DWR is used for AJAX functionality and it provides the mapping between our Java objects and methods to JavaScript objects and methods respectively (*Architecture · GitBook*, n.d.). To simplify the interactions with JavaScript and the browser, JQuery is used. Spring MVC is used to provide the Model-View-Controller design pattern.

2. **The Data Access layer**

   It is an abstraction layer from the actual data model and its changes. . It uses Hibernate as the Object Relational mapping tool, and Liquibase to manage relational database changes in a database-independent way. The relationships between our domain objects and database tables are mapped using a mixture of Hibernate annotations and XML mapping files. The data access layer is exposed to the service layer through interfaces, thereby shielding it from implementation details such as which object-relational mapping tool is being used.

3. **The Service layer**

   It is responsible for managing the business logic of the application. It is built around the Spring framework which helps perform a number of tasks like managing transactions in between service layer classes, providing dependency between components, authentication, logging, etc.

Chapter 3: openIMIS

‘openIMIS’ stands for Open Source Insurance Management Information System. It is an open-source platform that helps manage health financing processes. It is jointly funded by the Swiss Agency for Development and Cooperation (SDC) and the German Federal Ministry for Economic Cooperation and Development (*OpenIMIS*, n.d.). It provides schemes and insurance plans for reimbursement of health care providers like hospitals, pharmacies, and physicians.

It is basically a web-based software that allows for a seamless exchange of information between beneficiaries, health service providers, and payers of a health financing scheme. The software is modular and highly customizable and it helps manage: beneficiaries, benefits packages, providers, service claims generation, review and processing, client feedback, and monitoring and data analysis. It is a versatile open-source software that supports the administration of health financing and social protection schemes.

openIMIS contributes to achieving the Sustainable Development Goals (SDGs): especially SDG 1 i.e., *"End poverty in all its forms everywhere"*, and more specifically 1.3 i.e., *"Implement nationally appropriate social protection systems and measures for all, including floors, and by 2030 achieve substantial coverage of the poor and the vulnerable*" and SDG 3, target 3.8 i.e., *"Achieve universal health coverage, including financial risk protection, access to quality essential health-care services and access to safe, effective, quality and affordable essential medicines and vaccines for all"* (openIMIS *Home*, n.d.).

3.1. Vision and Mission

- **Vision**

  Effectively managed health financing and social protection mechanisms that contribute to Universal Health Coverage and Universal Social Protection.

- **Mission**

  Continuous improvement of the openIMIS software and its implementations, through the open source community, to efficiently manage health financing and social protection schemes.

With this vision and mission, openIMIS aims to improve health financing and social protection mechanisms globally through the open-source community.

3.2. Objective

openIMIS, being an open-source platform helps manage health financing processes. The objectives of openIMIS are discussed below:

- To improve health financing operations by offering a seamless exchange of data related to patients, health service providers, and payers for health care.
- To contribute to achieving sustainable goals by ending poverty of all forms everywhere and implementing nationally appropriate social protection systems and measures for all.
- To manage and improve health insurance schemes.

3.3. Features

openIMIS is a web-based software that allows for a seamless exchange of information between beneficiaries, providers, and payers of health financing and social protection schemes. Some of its features are discussed below:

1. **Modular design**

   openIMIS is highly modular and customizable. It offers functionality in beneficiary management (registration), benefits administration (including payments), and data analysis. Users can pick and choose the modules they require and can also modify the modules as per the requirement from social insurance to cash transfer schemes.

2. **Standards and interoperability**

   The openIMIS Initiative ensure interoperability in a larger systems context. openIMIS follows the international FHIR standard for information exchange, allowing it to work seamlessly with other systems which mean that openIMIS can readily exchange data with everything, for example, the data analytics platforms such as DHIS2 to electronic medical record systems like OpenMRS and other existing beneficiary management systems, such as those maintained by social security agencies. This makes openIMIS highly interoperable and compatible with other systems.

3. **Data security and privacy**

   openIMIS is related to financing so, security isn't compromised at all. It takes care to ensure that the software is free of security-related issues and that all required measures to ensure the safety of data are in place. However, system owners like the scheme operators are responsible for ensuring that the required security measures are in place on the virtual or physical server (where data from the openIMIS implementation are

stored) on which openIMIS is installed and hosted. They should regularly monitor whether required security measures are in place or not.

3.4. Open Source Support

openIMIS, like the DHIS2 and openMRS systems, is also open-source and is freely available without any cost. It is open-source software that helps scheme operators struggling with funding for such ICT systems by offering a free alternative for the digitalization and efficient management of health financing workflows. It is made up of a community of developers, implementers, users, and partners. Since it's a FOSS too, all the benefits of FOSS that we discussed in the previous sections apply to openIMIS to the same extent. The benefits include:

- Lower costs as it's not necessary to pay for prohibited license costs.
- Easier access to source code to enable integration and interoperability.
- There is a need for the user to have the freedom to make the changes as per the user requirement, given the information needs for the health sector are constantly changing and evolving.
- FOSS applications like openIMIS are supported by a global network of developers, and thus have access to cutting-edge research and development knowledge.

3.5. Target Stakeholders

openIMIS is open-source software that helps scheme operators struggling with funding for such ICT systems by offering a free alternative for the digitalization and efficient management of health financing workflows. The target stakeholders of this system are

- Health service providers and payers for health care.
- Health insurance organizations.
- Ministry of Health.
- Health institutions and organizations.
- Social security providers.

3.6. openIMIS Implementation in Nepal

The operator, Health Insurance Board (HIB) deployed openIMIS in Nepal in April 2016 [16]. The functionalities of openIMIS like enrolment, claims management, beneficiary management, claim review, and reports are used. The deployment has already covered 65% of districts in Nepal with the implementation partners GIZ Support to the Health Support Programme (S2HSP) and Swiss Tropical and Public Health Institute (Swiss TPH). It is supported by the German Development Cooperation.

The Social Health Insurance Scheme in Nepal is family-based, with up to 5 members contributing NRs. 3500 for a benefit package worth NRs. 100,000 per family, per year. The openIMIS deployment in Nepal uses resources from the interoperability framework of the Ministry of Health and Population – the health facility registry.

The primary focus of interoperability with openIMIS in Nepal has been around the electronic submission of claims via Electronic Medical Record Systems (EMRs). There are a few service providers that use EMRs in their hospitals and work is currently underway to design standard approaches to allow them to submit claims electronically. An integration of openIMIS and Bahmni has been implemented which is used in 4 health facilities in Nepal and is used by Bayalpata Hospital to submit their claims automatically to openIMIS.

As per the data provided by openIMIS Nepal, the facts and figures of the deployment are shown below:

Total enrolled	1,599,933
Active members	1,338,331
Health service providers	268
Total users (enrolment assistants, officers, claims entry staff, claims review staff)	4256

Within such a short span of time, more than 1.5 million people are already making use of the service. This shows that the deployment has really been helping the people with their insurance in Nepal. It has also helped the needy family with minimal income get health benefits from health institutions free of cost up to 100,000.

Chapter 4: FHIR HL7

FHIR stands for Fast Healthcare Interoperability Resources. It is a next-generation interoperability standard created by the standards development organization Health Level 7 (HL7). FHIR is designed to enable health data (including clinical and administrative data) to be quickly and efficiently exchanged. It is a standard describing data formats and elements, known as resources, and an API for exchanging Electronic Health Records (EHR). FHIR provides an alternative to document-based approaches by directly exposing discrete data elements as services. For example, using FHIR, basic elements of healthcare systems like patients, admissions, diagnostic reports, and medications can each be retrieved and manipulated via their own resource URLs.

HL7 is a set of standards, formats, and definitions for exchanging and developing electronic health records. It includes several standards like FHIR, CDA (Clinical Document Architecture), and CCD (Continuity of Care Document). FHIR aims to simplify implementation without sacrificing information integrity. It leverages existing logical and theoretical models to provide a consistent, easy-to-implement, and rigorous mechanism for exchanging data between healthcare applications. FHIR has built-in mechanisms for traceability to the HL7 RIM and other important content models.

4.1. Objective

The philosophy behind FHIR is *to build a base set of resources that, either by themselves or when combined satisfy the majority of common use cases.* The main objectives of FHIR HL7 are discussed below:

- To define a common standard of how healthcare information can be exchanged between different computer systems regardless of how it is stored in those systems.
- To create a standard that would lead to high adoption across disparate developer communities.
- To create a base set of resources that can meet the diversified needs of common healthcare data and interoperability use cases.
- To make health information easily accessible to the users regardless of the device they're requesting it from.

4.2. Core Capabilities

FHIR uses REST API for data exchange. REST defines categories of data, or 'resources', to exchange data. The philosophy of FHIR is to create a set of resources that, individually or in combination, satisfy the most common use cases. For example, the *Patient Resource* includes demographic data related to a patient – such as their name, address, and phone number. FHIR resources improve granular data retrieval so that a request returns just the relevant data rather than a full record or document that itself must then be searched.

FHIR uses modern security standards, including authentication and encryption. It can support labeling sensitive information so that only those who have the need and the right can see the information. Using FHIR, healthcare data is represented by many sets of vocabularies, terminologies, and codes which change and grow over time. Therefore, it is important for the data exchanged to be equally understood by the sender and receiver (semantic interoperability). The use of this data is managed by FHIR by including references to code definitions used for data verification and by allowing restrictions on the codes that can be used.

FHIR offers many improvements over the existing standards. They are discussed below.

- It has a strong focus on implementation: fast and easy to implement as multiple developers have had simple interfaces working in a single day.
- It has multiple implementation libraries, with many examples available to kick-start development.
- FHIR specification is free for use with no restrictions.
- It offers interoperability out-of-the-box: base resources can be used as is, but can also be adapted as needed.
- It has an evolutionary development path from HL7 Version 2 and CDA – standards can co-exist and leverage each other.
- It has a strong foundation in Web standards: XML, JSON, HTTP, OAuth, etc.
- Support for RESTful architectures, seamless exchange of information using messages or documents, and service-based architectures.
- It has concise and easily understood specifications.
- It has a human-readable serialization format for ease of use by developers

4.3. FHIR Resources

FHIR resources form the building blocks of healthcare. Resources are exchangeable content in interoperability solutions. Resources create commonly used groupings like lists and documents. In FHIR, healthcare data is broken down into categories such as patients, practitioners, laboratory results, insurance claims, etc. Each of these categories is represented by a FHIR Resource which defines the component data elements, data relationships that together make up an exchangeable patient record, and data constraints.

A Resource in its raw form does not require that most data elements be assigned a value but when it is customized for real-world use through the Profiling process, certain elements are then required so that the Resource can be functional. For example, a Patient Resource may be Profiled to require that a patient's name, telephone number, and address be supported to enable patient matching. Resources share the following set of characteristics:

- A common way to define and represent them, building them from data types that define common reusable patterns of elements
- A common metadata
- A human-readable part

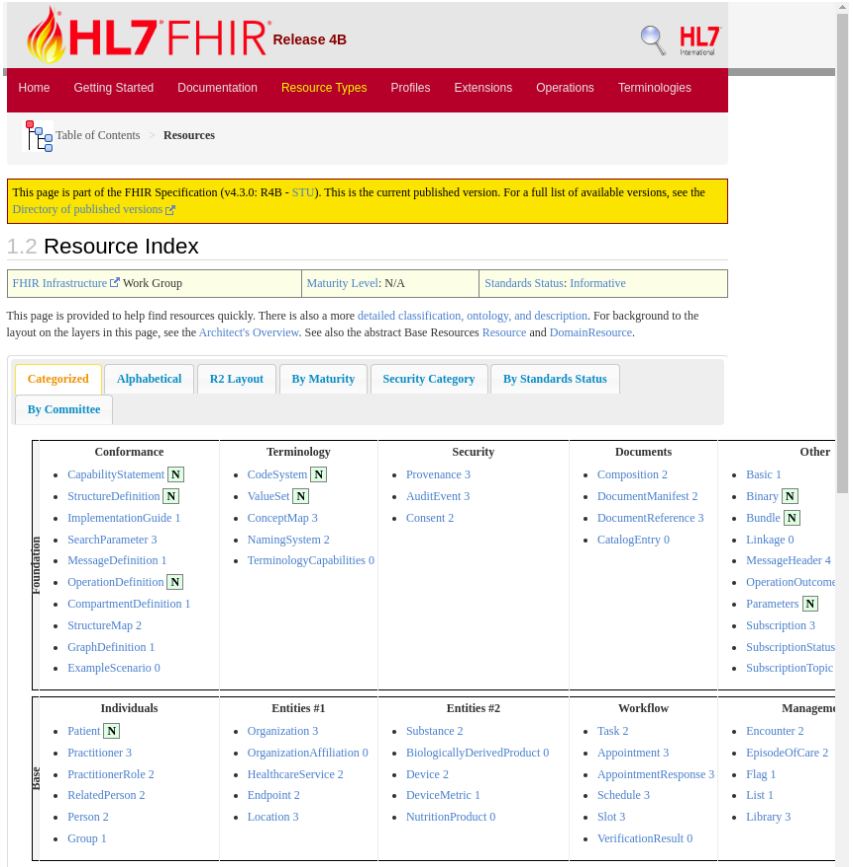

Figure 5: FHIR HL7 resource index

4.4. Levels in FHIR

The FHIR implementation is addressed by 5 levels. These levels are discussed in detail below:

1. **Level 1: Foundation Module**

   The Foundation Module is responsible for the overall infrastructure of the FHIR specification. Every implementer works with the content in the foundation module whichever way they use FHIR. The Foundation Module maintains most of the basic documentation for the FHIR specification.

   Some of its features are

   - All the other modules depend on the foundation module.
   - The Exchange module builds on the foundation model by defining the recognized methods for the exchange of resources.
   - The Terminology module provides the formal basis for using Concepts defined in Code Systems in the definition.
   - The Conformance module provides the basis for extending the foundation for national and local use.
   - The Security & Privacy provides the linking framework to external standards for security and privacy.
   - The Implementation Support module builds on the foundation to provide testing and reference implementations.

Figure 6: FHIR Level 1

2. **Level 2: FHIR Exchange Module**

   This module contains different modules to support the implementation. The different modules are

   i. **Implementor support**

      This section provides information that will be useful for FHIR implementers, including information about available libraries, tools, and other similar resources, as well as where to seek help.

ii. Security and privacy

The *Security and Privacy Module* describes how to protect the FHIR server through access control and authorization, how to document what permissions a user has granted (consent), and how to keep records about what events have been performed (audit logging and provenance).

iii. Conformance

The *Conformance Module* represents metadata about the datatypes, resources, and API features of the FHIR specification and can be used to create derived specifications.

iv. Terminology

The *Terminology Module* provides an overview and guide to the FHIR resources, operations, coded data types, and externally-defined standard and FHIR-defined terminologies that are used for representing and communicating coded, structured data in the FHIR core specification and profiles. Collectively, these capabilities are used to provide the terminology service functionality required for supporting the use of coded data in FHIR resources throughout the specification as described in the other modules.

v. Exchange

FHIR specifies the content of the data exchanged between healthcare applications, and how the exchange is implemented and managed. FHIR defines the following methods for exchanging data between systems:

- RESTful API
- Messaging
- Documents
- Services
- Database/Persistent storage

Most implementers focus on RESTful API. This is a client/server API designed to follow the principles of RESTful design. for Create, Read, Update and Delete operations, along with Search and Execute (Operations) support.

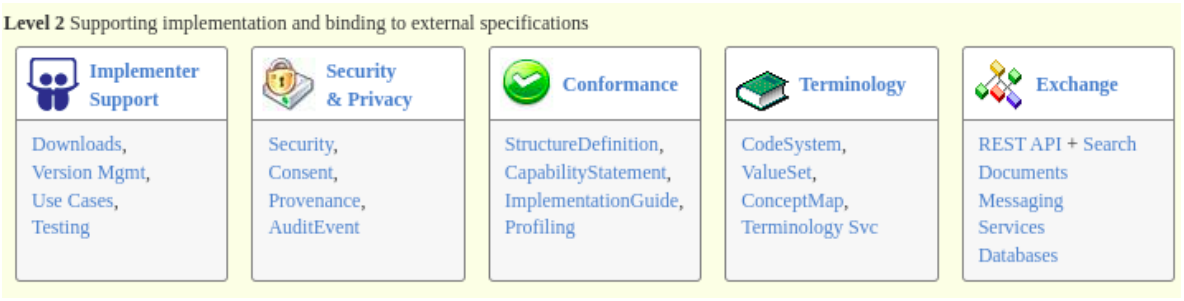

Figure 7: FHIR Level 2

3. Level 3: Administration Module

The *Administrative module* covers the base data that is then linked to other modules for clinical content, finance, billing, workflow, etc. It is built on the FHIR technology platform modules. It contains the standards for storing information about the following:

- Patient
- Practitioner
- Care team
- Device
- Organization
- Location
- Healthcare service

Figure 8: FHIR Level 3

4. Level 4: Record keeping and data exchange for healthcare process

This level provides standards for record-keeping and data exchange for clinical, diagnostics, medications, workflow and financial modules.

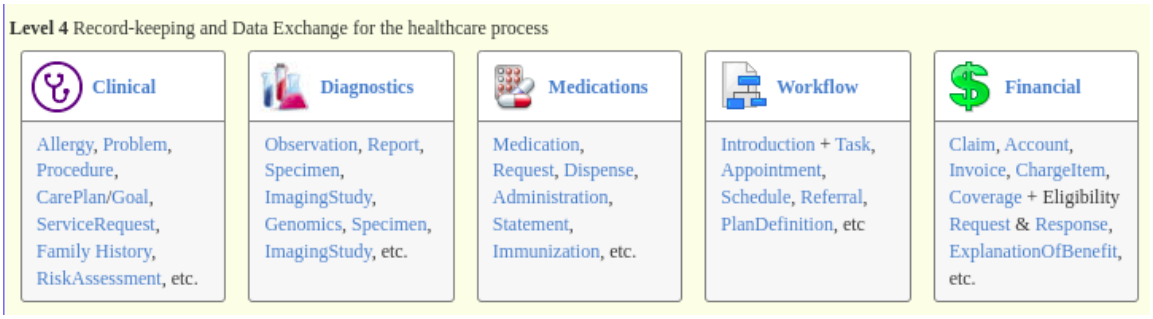

Figure 9: FHIR Level 4

5. Level 5: Providing the ability to reason about the healthcare process

This level stores the clinical reasoning and medication definition that provides the ability to reason about the healthcare process.

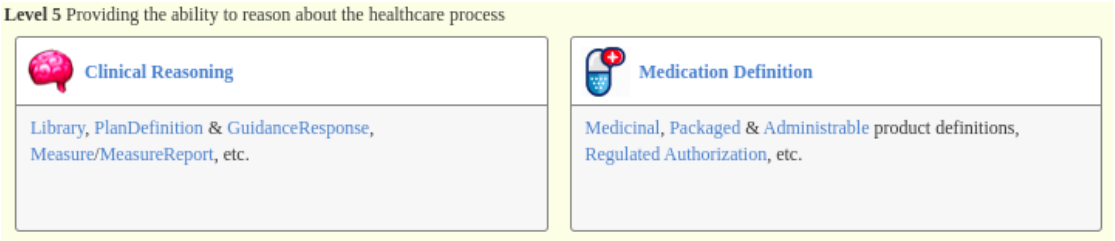

Figure 10: FHIR Level 5

4.5. Benefits of FHIR in the healthcare sector

Due to the most up-to-date standard for the exchange of data, FHIR allows the modification of the implementers leading to extensibility without violating the core standards. The benefits of FHIR in the healthcare sector are discussed below.

1. **Holistic patient experience**

   Patients have complete control over their health information and how they are going to share it with the healthcare provider due to the extreme simplicity in the implementation and integration of this protocol.

2. **Improvement in clinical treatment**

   The efficiency of the clinical system will increase drastically when healthcare providers have easy access to the EMR and patient-related information. The backbone of good treatment is patient data – previous medical records and treatment play a major role during research and diagnosis. With FHIR's implementation, data sharing standards will allow a safe and easy exchange of data which leads to efficiency and fast data sharing.

3. **Automated data structure facility**

   Automated data structuring plays a major role when clinical support is asked for providers and professionals in the healthcare industry. FHIR facilitates better care management and holistic plus a trustworthy patient experience which leads to cost savings on both ends.

4. **Enhancement in data management**

   Real-time data records are available which leads to improved data management and accuracy. Since, in real-time, data is derived from multiple sources across the globe

from various platforms, adds up to a large volume. Management of this whole data will require fast access and exchange across systems which is the core benefit of FHIR.

5. **Easy third-party integration**

   The third-party integration is very easy using FHIR. It doesn't require specific software, rather it works well with any third-party applications and software.

Discussion

The sole purpose of this case study was to get insights into the open-source tools used for healthcare systems – DHIS2, openMRS, openIMIS, and FHIRHL7. This study focused on these but there are more open-source tools available free to use than these. With this case study, I've found out how significant these tools are and each of these tools is equally important and contributes equally to the health sector if implemented properly. Many developing countries are benefitting from these tools. They are not only helping digitize medical records keeping but also helping in COVID, and HIV/AIDS mitigation in many countries. For example, openIMIS currently supports the management of processes for health financing schemes in four African countries: Cameroon, Chad, the Democratic Republic of Congo, and Tanzania. In Nepal, openIMIS manages the National Social Health Insurance scheme.

Similarly, resource-constrained countries are making use of the openMRS software as an electronic medical record system. The FHIR standard enables data interoperability as it is based on lightweight open-source RESTful services. The FHIR HL7 standards have changed the way health information is stored and transferred. They allow countries to create a digital "health web" by using the HTTP protocol as an information exchange platform.

This study proves that the ability to make efficient use of software that is freely available to us can change the entire healthcare system of any country. Developing and resource-constrained nations can benefit from such software and foster their development in the health sector. Health data is very important and to be able to digitally represent, store and transfer the health data would benefit any nation both economically and health-wise. Therefore, open-source tools like these should be explored and implemented. Nepal, for instance, has implemented the openIMIS system to help manage the health financing process and within a very short period, millions of people are already using the service and both the country and the citizens are benefitting from the service. In this way, if the traditional method of health record keeping in Nepal could be digitized using the tools we discussed, the health sector in Nepal would flourish which would ultimately help the country speed up the development.

Conclusion

There are many open-source tools available for healthcare systems – DHIS2, openMRS, openIMIS, FHIR HL7, etc., and making efficient use of these tools would help the health sector flourish. These tools are not only freely available to us, but come with extensive documentation, and guides to help us implement them as per our requirements. The study on this healthcare software has helped broaden my knowledge of the medical healthcare software that is freely available to us and the benefits of implementing it. This study has helped provide insights into the open-source healthcare tools that are being used by countries all over the world, including our country Nepal itself, and the impact these tools have made in those countries.

To sum up, open-source healthcare tools are like a boon to the health sector. The ability to implement them efficiently would help solve many problems – for health institutions, medical personnel, patients, and the country itself. DHIS2 helps aggregate statistical data collection, validation, analysis, management, and presentation of patient-based statistical data. Similarly, openMRS helps improve healthcare delivery in resource-constrained environments by coordinating a global community that creates and sustains a robust scalable, user-driven, and open-source medical record platform. openIMIS, on the other hand, helps improve health financing operations by offering a seamless exchange of data related to patients, health service providers, and payers for health care. Lastly. FHIR HL7 helps define a common standard of how healthcare information can be exchanged between different computer systems regardless of how it is stored in those systems.

References

About DHIS2. (n.d.). DHIS2. Retrieved November 1, 2022, from https://dhis2.org/about/

Architecture · GitBook. (n.d.). GitBook. Retrieved November 1, 2022, from http://devmanual.openmrs.org/en/Technology/architecture.html

DHIS2 Cloud Hosting. (n.d.). DHIS2. Retrieved November 2, 2022, from https://dhis2.org/hosting/

Downey, M. (2013, October 30). *System Requirements - Documentation*. OpenMRS Wiki. Retrieved November 2, 2022, from https://wiki.openmrs.org/display/docs/System+Requirements

Home. (n.d.). YouTube. Retrieved November 2, 2022, from https://openimis.atlassian.net/wiki/spaces/OP/pages/3179380737/Opening+remarks

Home. (n.d.). YouTube. Retrieved November 3, 2022, from https://hl7.org/fhir/

Landley, R. (n.d.). *BSD licenses*. Wikipedia. Retrieved November 21, 2022, from https://en.wikipedia.org/wiki/BSD_licenses

Mamlin, B., Issa, A., & Downey, M. (2021, May 6). *Introduction to OpenMRS - Documentation*. OpenMRS Wiki. Retrieved November 3, 2022, from https://wiki.openmrs.org/display/docs/Introduction+to+OpenMRS

Nepal. (n.d.). openIMIS. Retrieved November 3, 2022, from https://openimis.org/nepal

openIMIS. (n.d.). Wikipedia. Retrieved November 4, 2022, from https://en.wikipedia.org/wiki/OpenIMIS

OpenMRS. (n.d.). United Nations Development Programme. Retrieved November 4, 2022, from https://www.undp.org/policy-centre/singapore/openmrs

Topics Understanding APIs What is a REST API? (2020, May 8). Red Hat. Retrieved November 21, 2022, from https://www.redhat.com/en/topics/api/what-is-a-rest-api

What is DHIS2? (2022, September 13). DHIS2 Documentation. Retrieved November 7, 2022, from https://docs.dhis2.org/en/use/what-is-dhis2.html