FIND DxPulse App

Technical Design Documentation

FIND
Diagnosis for all

HISP
SOUTH AFRICA

Version Approval Control

Role	Author	Reviewer and approver
Name	Farai Mutero	Kabelo Molepo
Designation	Manager: Innovation	Project Manager
Organisation	HISP-SA	HISP-SA
Signature		

Document No# : DxPulse Technical Documentation	
Current Version: Version 2	
Effective Date: 2025/04/01	Review Date: 2025/04/01

Name of Person Making Change	Role of Person Making Change	Date of Change	Version Number	Notes
Farai Mutero	Manager: Innovation	15/12/2024	Draft 0.1	First draft of document
Mosa Lefu	Software Engineer	16/12/2024	Draft 0.1	Contributed to key sections of the document
Farai Mutero	Manager: Innovation	17/12/2024	v 1.0	Finalized and submitted first version of document
Farai Mutero	Manager: Innovation	01/04/2025	v 2.0	Added the following sections: 1. Add API endpoints for all the favourites for all pages 2. Architecture Diagram (Show DHIS and DxPulse interactions) 3. DHIS App installation guide 4. CI/CD and code management 5. App Folder Structure 6. Read me file updates?
Farai Mutero	Manager: Innovation	21/05/2025	v2.1	Updated The term DNO App to DxPulse

1. Table of Contents

2. Abbreviations

API	Application Programming Interface
CORS	Cross-Origin Resource Sharing
CPU	Central Processing Unit
DHIS2	District Health Information System 2
DxPulse	Diagnostics Network Optimization
FIND	Foundation for Innovative New Diagnostics
GIS	Geographic Information System
HTTP	HyperText Transfer Protocol
IMSD	Instance Management Specification Document
ISO	International Organization for Standardization
Lab	Laboratory
OrgUnit	Organisational Unit
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSL	Secure Sockets Layer
TLS	Transport Layer Security
URL	Uniform Resource Locator

3. Introduction

The Diagnostic Network Optimization (DxPulse) App has been designed as a web app built on the back of DHIS2's powerful and mature web app management platform and Web API. The idea behind this design decision was intended to identify an app delivery mechanism that will reduce app hosting requirements for countries seeking to implement DxPulse in their contexts as well as leverage existing infrastructure investments targeting platforms like DHIS2, which is widely used in many countries around the world (HISP Centre, University of Oslo (UiO), 2024).

4. Purpose

The purpose of this document serves as high-level guideline for how the DxPulse app was designed, its core capabilities and how the capabilities that the user interacts with are delivered based on the application's design. It is designed to be a companion to the code base, the link to which, is shared in

"Other" section of this document. This document is not intended to exhaustively document all the code that makes up the DxPulse App, however it is designed to explain key capabilities and features that may be useful for an implementer or a developer moving forward.

5. Version Information

Any changes to the instance implementation and management information should be recorded in this section and it is the responsibility of the Instance Data Manager to ensure that this document is always up to date and that the changes are approved by the relevant parties.

Name of Person Making Change	Role of Person Making Change	Date of Change	Version Number	Notes
Farai Mutero	Innovation Manager	17/12/2024	Draft 0.1	First draft of document
Mosa Lefu	Software Engineer	18/12/2024	V1.0	Review and corrections where needed.
Comfort Mankga	Software Engineering Manager	03/05/2025	V1.0	Review

6. App Folder Structure

The FIND DxPulse App project is organized into a well-structured folder hierarchy to ensure modularity, maintainability, and scalability. At the root level, there are several key directories and files that serve specific purposes.

The app folder contains the core logic of the application. Within this folder, there are subdirectories such as components, which houses reusable UI components like buttons, modals, and charts.
The services subdirectory contains service files responsible for handling API calls and data processing. Additionally, the utils folder includes utility functions such as helpers and formatters. The app folder also includes specific scripts like welcome.js, which manages the logic for the welcome page, and main.js, which serves as the central configuration file for routes, server settings, and API endpoints.

The templates folder is dedicated to storing HTML templates for different pages of the application. For example, dataupload.html is the template for the data upload page, welcome.html is used for the welcome page, and welcome_page.html serves as the template for the home page.

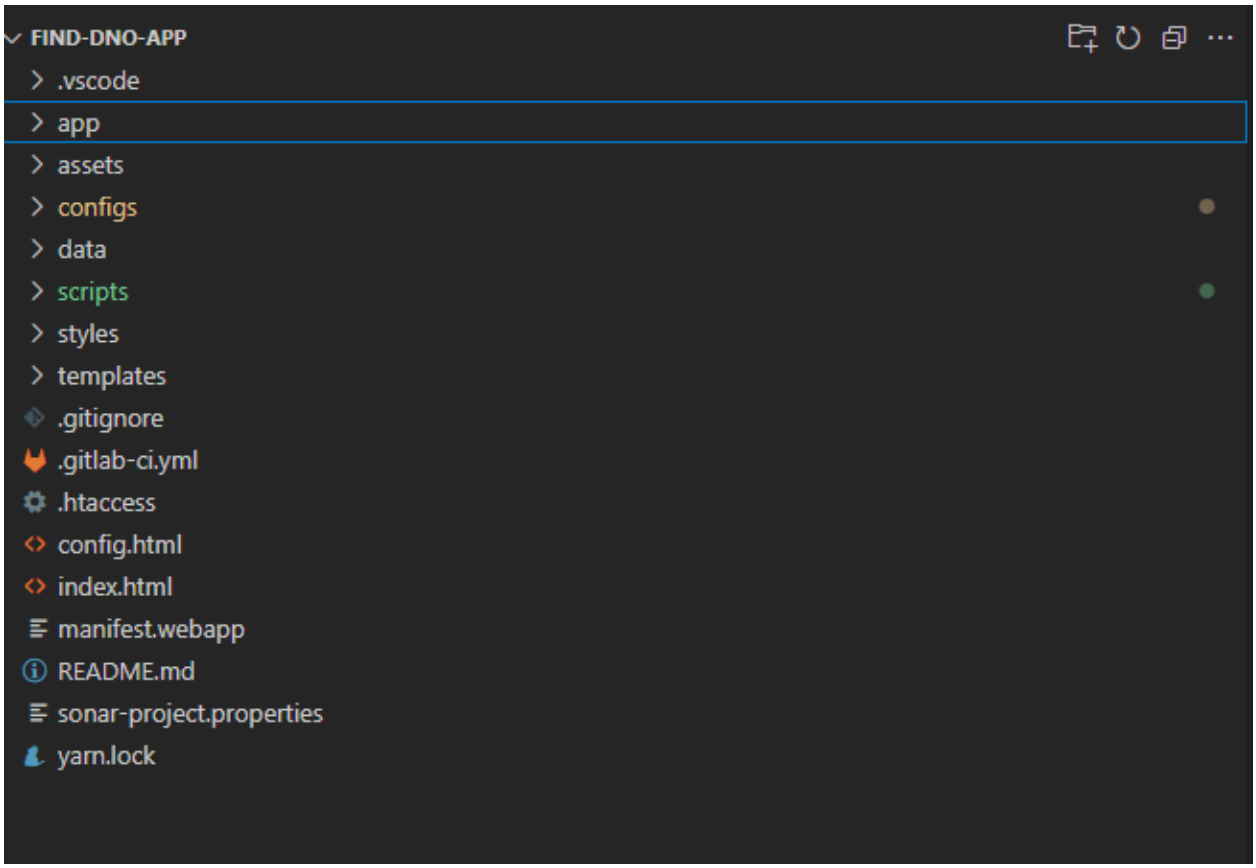


Fig 1.0 – DxPulse App Folder Structure

The assets folder contains all static resources used in the application. It is further divided into subdirectories such as css for stylesheets and images for image assets.

At the root level, there are several important files. The README.md file provides documentation about the project, including its purpose, features, and usage instructions. The package.json file is a Node.js configuration file that lists the project's dependencies and scripts. The .gitignore file specifies which files and folders should be ignored by version control.

This folder structure is designed to separate concerns, making it easier for developers to navigate, understand, and extend the application. Each folder and file have a clear purpose, contributing to the overall organization and maintainability of the project.

6.1 App folder

The JavaScript files located in the app folder are essential to the functionality and user experience of the FIND DxPulse app. These files play a critical role in displaying the main dashboard, various types of maps, and network performance indicators. The dashboard.js manages the logic for displaying the main dashboard, collecting and processing data from various sources, and presenting it in a user-friendly format. The various maps related JavaScript files are dedicated to rendering various types of maps within the application, generating interactive maps that display geographical data related to the laboratory network.

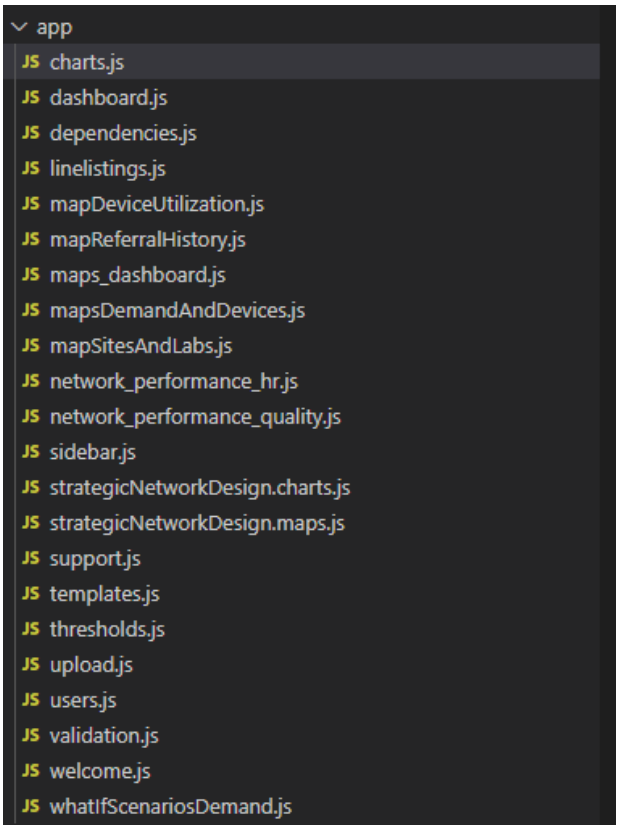


Fig 1.1 – **app** folder contents

network_performance_hr.js and network_performance_quality.js focuses on network performance indicators, retrieving and processing data like accredited labs, biosafety levels distribution, and overall laboratory accredication, and presenting this information in charts and graphs.

welcome.js handles the logic for the welcome page, providing users with an entry point into the application and setting up shortcuts to auxiliary resources such as the User Guide, Helpdesk & Contact Info, and Data Security Note.

By organizing these JavaScript files into specific functionalities, the FIND DxPulse App project maintains a clear and efficient structure. Each file has a distinct role, contributing to the overall goal of optimizing the laboratory diagnostic network. This modular approach facilitates easier maintenance, scalability, and future enhancements, ensuring that the app continues to meet the needs of its users.

6.2 Assets folder

The assets folder in this project is dedicated to storing all static resources required by the application. It is organized into subdirectories to ensure a clean and maintainable structure, making it easier to manage and locate specific assets. Below is a detailed description of its contents:

1. **fontawesome:** The directory contains resources needed to integrate and use FontAwesome icons in a project. It includes a **CSS folder** with stylesheets like fontawesome.css or fontawesome.min.css that define the classes for the icons. A **webfonts folder** contains font files in formats such as .woff, .woff2, .ttf, .eot, and .svg, ensuring compatibility across different browsers.

2. **images**: This subdirectory stores all image assets used in the application, such as logos, icons, and other graphical elements. These images are typically used to enhance the user interface and improve the overall user experience.

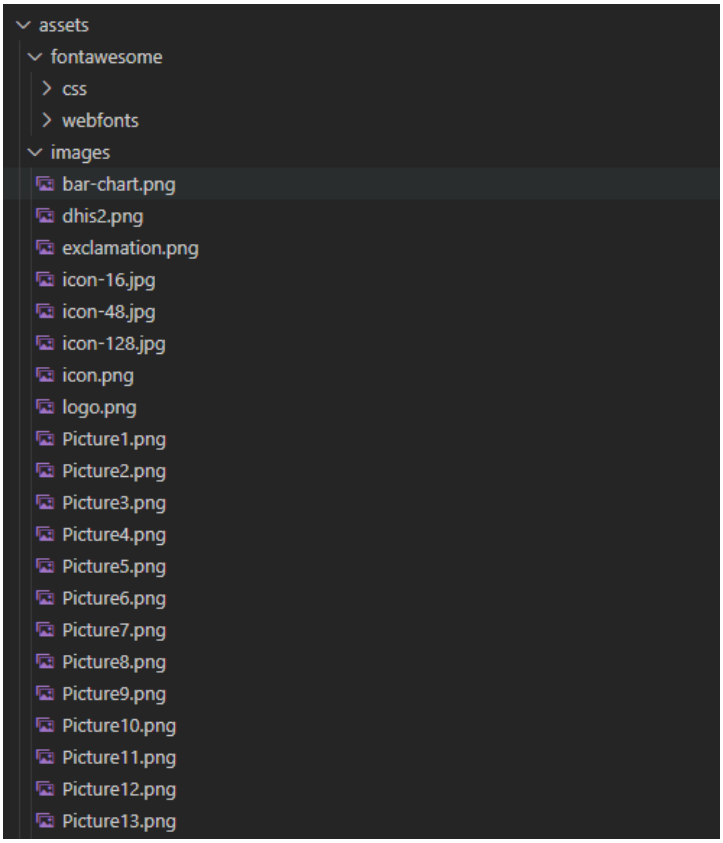


Fig 1.2 – **`assets`** folder contents

6.3 Configs folder

The open file, main.js, defines the routing configuration for the application. It maps specific URL paths to corresponding templates, scripts, and other dependencies required to render each route. Each route is represented as a key-value pair within the routes object, where the key is the URL path, and the value is an object containing details about the route. These details include the HTML template to be loaded, the JavaScript file to execute, and any additional components or dependencies required for the route. For example, the root route ("/") loads the dashboard.html template, executes the dashboard.js script, and includes a sidebar component with its own template. It also specifies a dependency on the treeView.js script and an empty array for global filters.

The file also includes routes for more specific functionalities, such as "/mapsdashboard," "/what-if-scenarios/demand," and "/strategicNetworkDesign/capacity." Each of these routes is configured similarly, with unique templates and scripts tailored to their respective purposes. For instance, the "/strategicNetworkDesign/capacity" route uses the strategicNetworkDesign.html template and the strategicNetworkDesign.maps.js script, along with dependencies like strategicNetworkDesign.charts.js and treeView.js.

This modular structure allows the application to dynamically load and manage different views and functionalities based on the user's navigation, ensuring a clean and maintainable codebase. The

configuration also highlights the use of reusable components, such as the sidebar, which is included in multiple routes to maintain consistency across the application's interface.

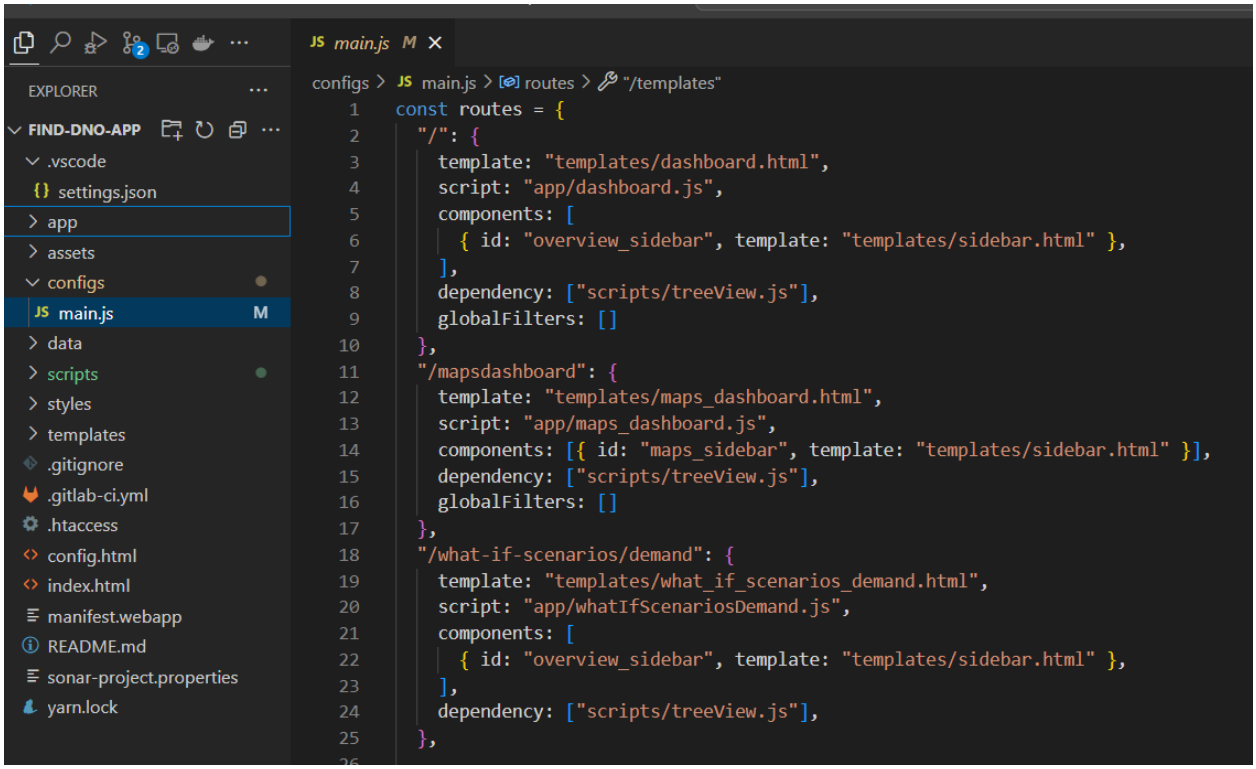


Fig 1.3 – Snippet of **`main.js`** file in the **`configs`** folder

6.4 Data folder

The data folder for the DxPulse App serves as a centralized location for storing static or dynamic data files that the application relies on. For example, one of the data files the app depends on the is the analytics.json file.

This file is structured JSON file containing metadata and data headers used for analytics purposes. The headers section defines the structure of the data, including attributes such as "dx" (Data), "ou" (Organisation unit), and "value" (Value). Each header specifies properties like the column name, value type (e.g., TEXT or NUMBER), data type (e.g., java.lang.String or java.lang.Double), and whether the field is hidden or contains metadata.

Other data files in the directory, HIS_KE_ou2.json, contains structured JSON dataset representing organizational units, specifically counties in Kenya, along with their associated metadata and geographic information.

Each organizational unit is described as an object within the organisationUnits array, including attributes such as the county's name, a short name, a unique identifier (id), and the hierarchical path within the organizational structure (path).

Additionally, the file includes geographic data under the geometry attribute, which specifies the type of geometry (e.g., "Polygon") and the coordinates outlining the boundaries of each county. This data is used for mapping and visualization purposes, enabling applications to display the spatial distribution of

organizational units. The file also includes timestamps (lastUpdated) to track when each unit's data was last modified, ensuring the information remains current and reliable.

6.5 Scripts folder

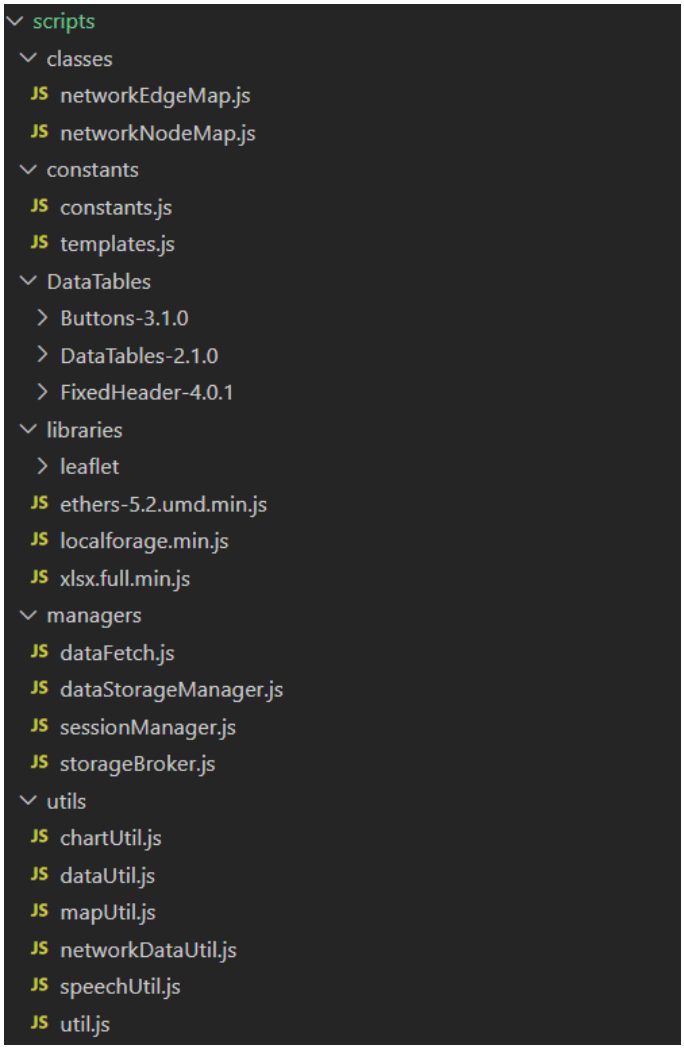


Fig 1.4 – Contents of the **`scripts`** folder

The scripts folder contains JavaScript or other script files that handle the core functionality and interactivity of the application. It includes utility scripts with reusable functions for tasks like data formatting, API requests, or validation.

Additionally, it includes page-specific scripts that manage the behaviour and logic of individual pages or components. Event handler scripts are also included, enabling user interactions such as button clicks or form submissions. The folder also includes third-party libraries or custom plugins to extend functionality, as well as build or deployment scripts for automating tasks like bundling or minification. Some of these libraries include:

Library	Description	Features	Use Case
DataTables	Versatile jQuery plugin for enhancing HTML tables	Pagination, sorting, filtering, customizable styling, Ajax integration	Creating dynamic, interactive tables on web applications
Leaflet	Lightweight, open-source JavaScript library for interactive maps	Flexibility for customizing map's appearance and functionality	Web mapping applications
LocalForage	JavaScript library for improving offline experience of web applications	Asynchronous storage, wraps IndexedDB and WebSQL	Storing and retrieving data
SheetJS	JavaScript library for working with spreadsheet data	Read, write, and manipulate Excel files and other spreadsheet formats	Web browsers or server environments
Ethers	JavaScript library for interacting with the Ethereum blockchain	Creating dApps, managing wallets, connecting to Ethereum nodes	Decentralized applications

6.6 Styles folder

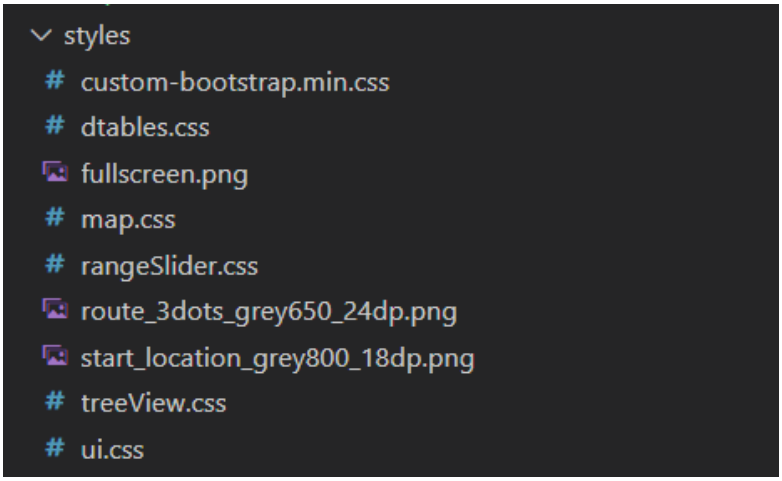


Fig 1.5 – Contents of the **`styles`** folder

The `styles` folder contains all the CSS files that define the visual appearance and layout of the application. It includes global stylesheets that set the overall design, such as typography, colors, and spacing, as well as component-specific stylesheets for individual UI elements like buttons, modals, and treeView.

6.7 Templates folder

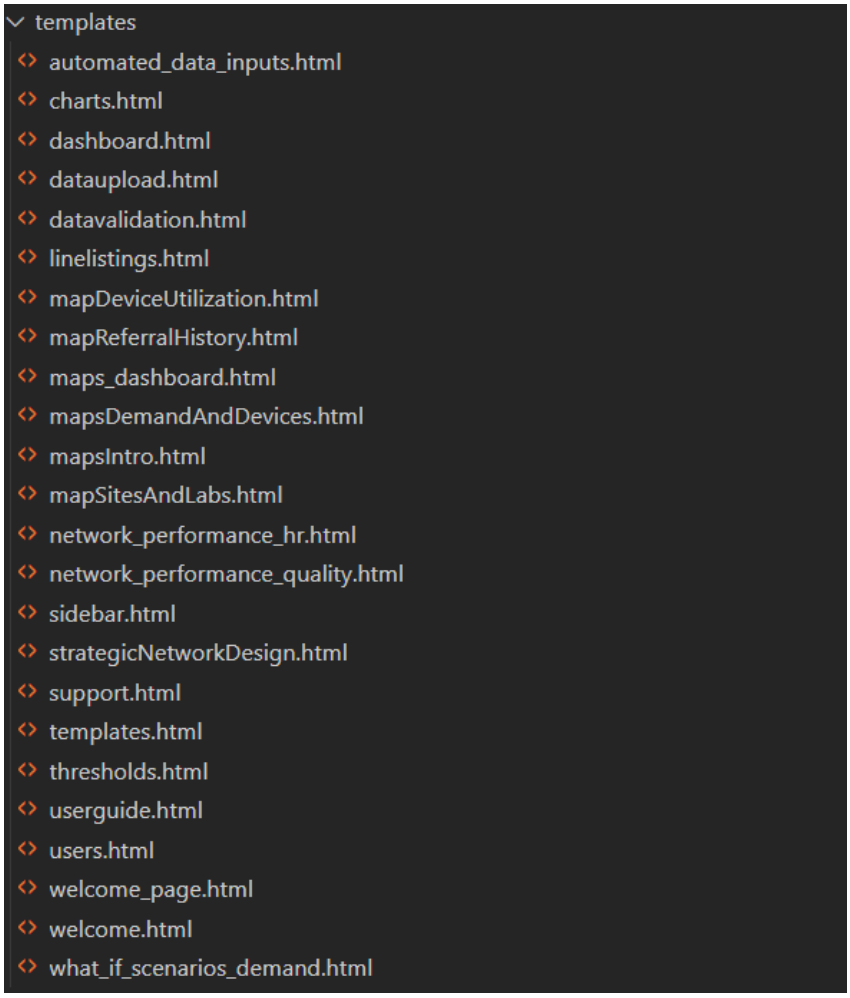


Fig 1.6 – Contents of the **`templates`** folder

The templates folder contains the HTML files template files that define the structure and layout of the application's user interface. These templates serve as the foundation for rendering different pages or components of the application, including placeholders or dynamic elements that are populated with data at runtime.

The folder may include templates for specific pages, such as dashboard.html for the General Summary dashboard page, forms, or reports, as well as reusable partial templates for common elements. By organizing these files in a dedicated folder, the project ensures a clear separation between the visual structure and the underlying logic, making it easier to maintain and update the user interface. This folder is essential for projects that use templating engines or frameworks to dynamically generate content.

7. App Architecture

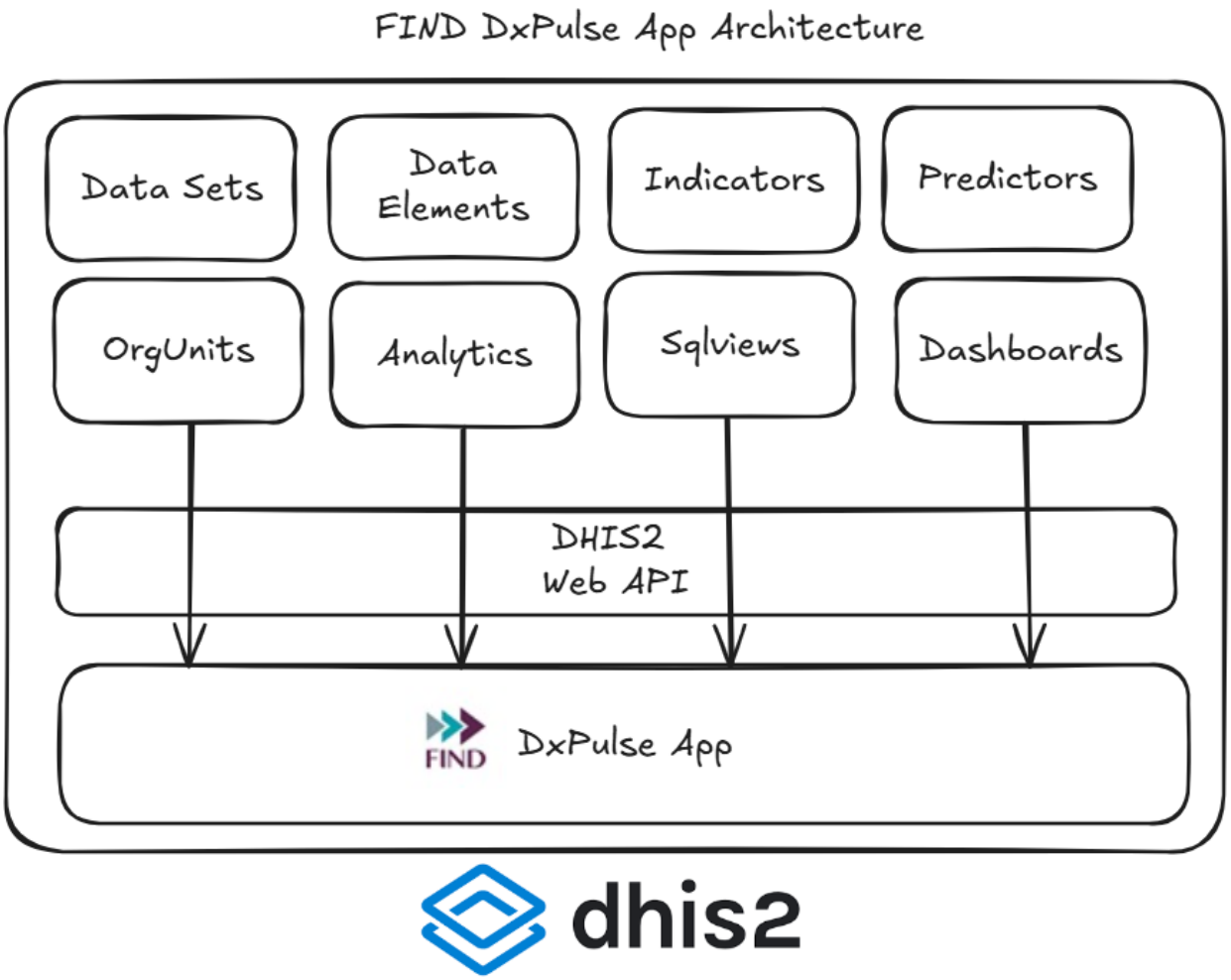


Fig 1.7 – DxPulse App Architecture

As per the architecture diagram above, FIND DxPulse App is a sophisticated web application designed to enhance the performance and efficiency of laboratory diagnostic networks. It is specifically built to be integrated and installed within the DHIS2 platform, leveraging its robust infrastructure and comprehensive data management capabilities. Installation is straightforward, allowing users to seamlessly incorporate the app into their existing DHIS2 environment. This integration ensures that the app benefits from the extensive functionalities offered by DHIS2, including data collection, management, and analysis.

The DxPulse App relies heavily on the DHIS2 Web API to access essential metadata and functionalities. This dependency allows the app to efficiently retrieve and process key information, ensuring accurate and up-to-date data is available to users. The specific dependencies include organizational units for mapping various laboratories, analytics sqlviews for complex data analyses, and DHIS2 dashboards to present key performance indicators. These integrations enable better decision-making, improved performance, and ultimately, a more optimized laboratory network.

8. Welcome Page

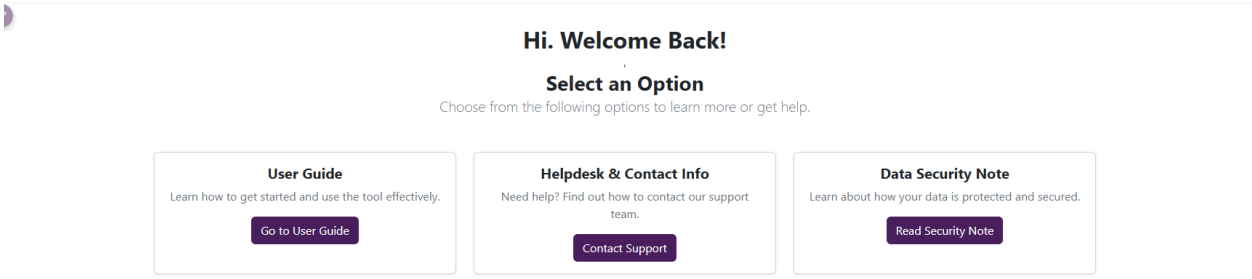


Fig 1.8 – Welcome Page

Purpose

The Welcome Page is designed to serve as an entry point into the DxPulse App, as well as to provide some shortcuts to some useful auxiliary resources. The shortcuts available on this page are as follows:

- **User Guide:** Provides a link to the Training App that contains user training resources
- **Helpdesk & Contact Info:** Leads the user to a "Contact Support" page where they can complete a form, to request support
- **Data Security Note:** This is a link to the Data Security Guidelines related to the usage of the app.

Technical Design

Templates/welcome.html

All functionality visible on the front-end has been setup as HTML markup in this page. This includes links to the FIND Training App as well as the text that appears in the modal for the Data Security Note.

9. General Summary

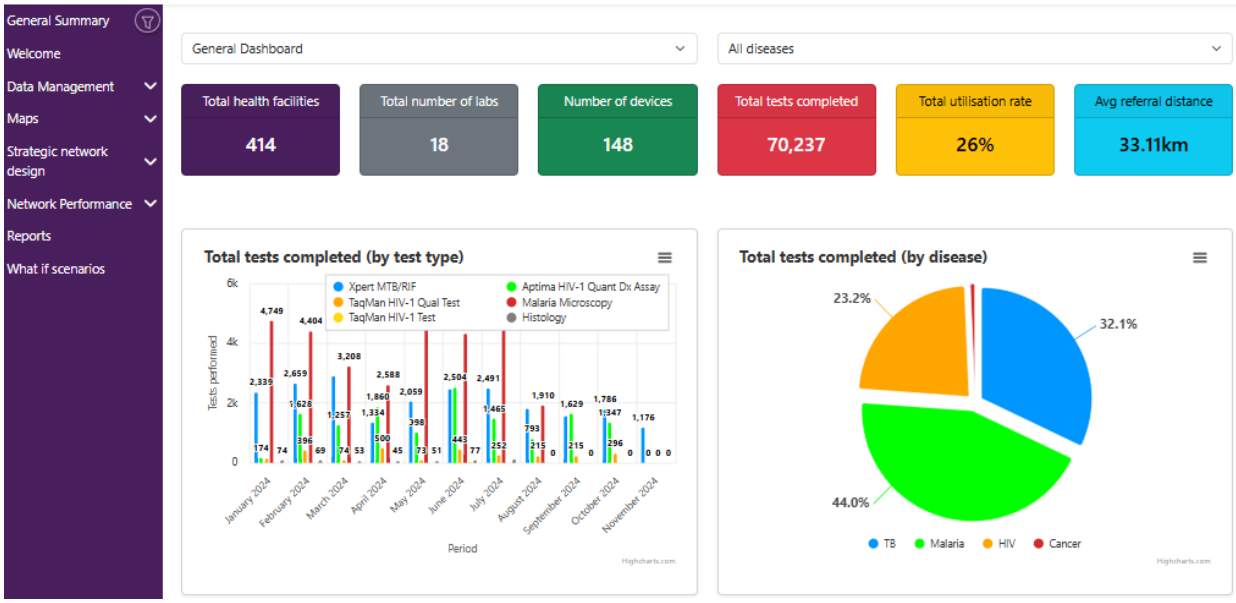


Fig 1.9 – General Summary web page

Purpose

The General Summary is the main dashboard of the DxPulse app. It displays high level critical data points related to the performance of the diagnostic network. At the top of dashboard there is a filter for dashboard type as well as disease type.

Fig 2.0 – General Summary Indicator cards

At the top of the dashboard page is a series of indicator “cards” (see Fig.1.2 above) that display key indicators that give an indication of the state of the diagnostic network. These indicators are s follows:

1. **Total health facilities:** Displays the number of health facilities that make the diagnostic network. This includes facilities where testing is done, and facilities that submit sample referrals
2. **Total number of labs:** This card only the number of labs in the network where testing is done
3. Number of devices: This card displays the number of diagnostic devices by disease in the diagnostic network. It is responsive to changes in the disease filter
4. **Total tests completed:** This card displays the total number tests completed for the reporting period in the underlying data set.
5. **Total utilization rate:** This card displays a total utilization rate for all the devices in the diagnostic network calculated based on testing demand, tests completed and availability factor of the testing devices.

6. **Average referral distance**: The average referral distance from referral sites to testing laboratories.

Technical Design

- templates/dashboard.html
- app/dashboard.js

Dependencies

- *templates/sidebar.html*
- *configs/main.js*
- *scripts/treeView.js*

DHIS Favourites Dependencies

Top Cards

Top cards refer to the key KPI values that are displayed at the top General Summary Page, just below the Dashboard and Disease Type drop-down.

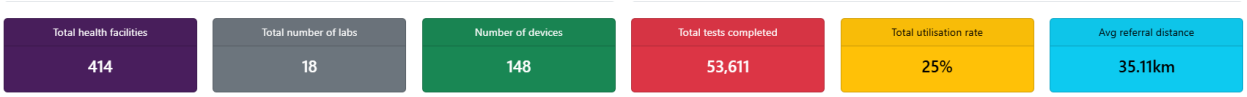


Fig 2.1 – General Summary Indicator cards

The DHIS2 dashboard favourites that provide the data that is displayed on each of the cards are as follows:

DHIS2 Favourite Name	Description
Total health facilities	Provides a value of the total health facilities in the diagnostic network. The API call is filterable by: • Period • Disease Type • Facility (OrgUnit)
Total number of labs	Provides a value of the total number of health facilities that are labs, in the diagnostic network. The API call is filterable by: • Period • Disease Type • Facility (OrgUnit)
Number of devices	Provides a value of devices in the diagnostic network. The API call is filterable by: • Period • Disease Type • Facility (OrgUnit)

Total tests completed	Provides a value of the total number of tests completed. The value displayed is dependent on the selected disease. The API call is filterable by: • Period • Disease Type • Facility (OrgUnit)
Total utilization rate	Provides a value of the total utilization rate. This is a dynamic value that is dependent on the selected disease. The API call is filterable by: • Period • Disease Type • Facility (OrgUnit)
Average referral distance	Provides a value of the average referral distance between diseases in the diagnostic network. The API call is filterable by: • Period • Disease Type • Facility (OrgUnit)

Charts – General Dashboard

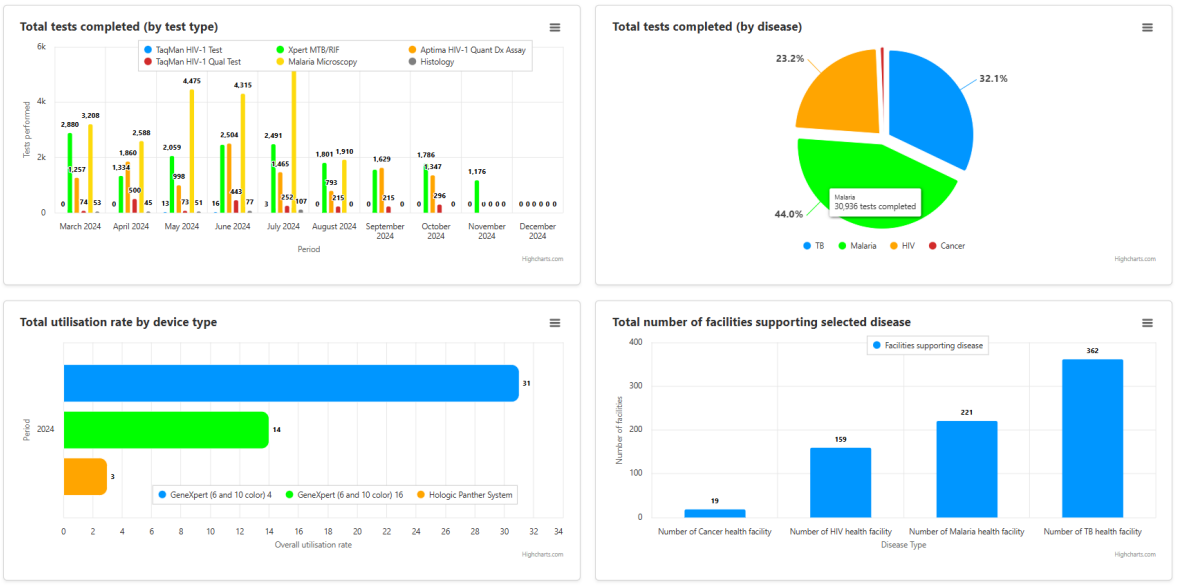


Fig 2.2 – General Dashboard on General Summary page

The General Dashboard charts provide a wide range of key insights related to completed tests, utilization rates and facilities. The DHIS2 dashboard favourites that provide the data that is displayed on each of the cards are as follows:

DHIS2 Favourite Name	Description
Total tests completed (by test type)	Provides data for total number of tests completed. The data is distributed by test type. The API call is filterable by: • Period • Disease Type • Test Type • Device Make • Facility (OrgUnit)
Total tests completed (by disease)	Provides data for total number of tests completed. The data is distributed by disease type. The API call is filterable by: • Period • Disease Type • Test Type • Device Make • Facility (OrgUnit)
Total utilisation rate by device type	Provides data for total utilisation rate. The data is distributed by disease type. The API call is filterable by: • Period • Disease Type • Test Type • Device Model • Facility (OrgUnit)
Total number of facilities supporting selected disease	Provides data for the total number of facilities. The data is distributed by disease type. The API call is filterable by: • Period • Disease Type • Test Type • Device Make • Facility (OrgUnit)

Charts – Historical Demand Dashboard

Fig 2.3 – Historical Demand Dashboard on General Summary page

The Historical Demand dashboard charts provide a wide range of key insights related to samples sent distributed by county, samples sent distributed by test type and utilization rate-based insights, namely utilization rate by facility and utilization rate by device type. The DHIS2 dashboard favourites that provide the data that is displayed on each of the cards are as follows:

DHIS2 Favourite Name	Description
Samples Sent (By County)	Provides data for the total number of samples sent in the diagnostic network. The data is distributed by county. The API call is filterable by: • Period • Disease Type • Test Type • Device Make • Facility (OrgUnit)
Samples Sent (By Test Type)	Provides data for the total number of samples sent in the diagnostic network. The data is distributed by test type. The API call is filterable by: • Period • Disease Type • Test Type • Device Make • Facility (OrgUnit)
Utilization Rate (By Facility)	Provides data for total utilization rate of devices. The data is distributed by facility type. The API call is filterable by: • Period • Disease Type

	• Test Type • Device Make • Facility (OrgUnit)
Utilization Rate (By Device Type)	Provides data for total utilization rate of devices. The data is distributed by device type. The API call is filterable by: • Period • Disease Type • Test Type • Device Make • Facility (OrgUnit)

10. Data Management

10.1 Templates

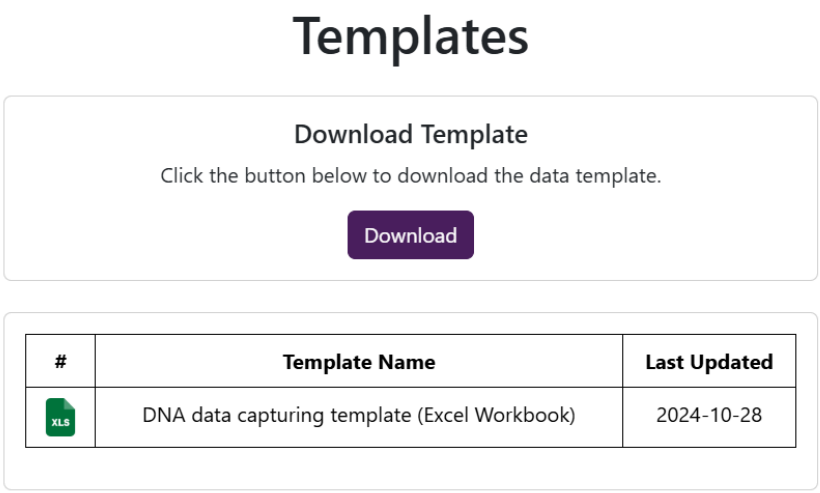


Fig 2.4 – Templates section

Purpose

Templates serve as a way of guiding the user regarding the imports of the data into the system. It provides a dummy file that indicates how data must be structured into the Excel file to match the data import method expected. It provides the user with the download button to download the template file.

Technical Design

- *templates/templates.html*
- *app/templates.js*
- *index.html*

The index.html provides a reference to the templates.html page. The HTML page serves as the front-end UI for the visualization and all the functionality of the page and calls are done on the .js file.

10.2 Data Validation

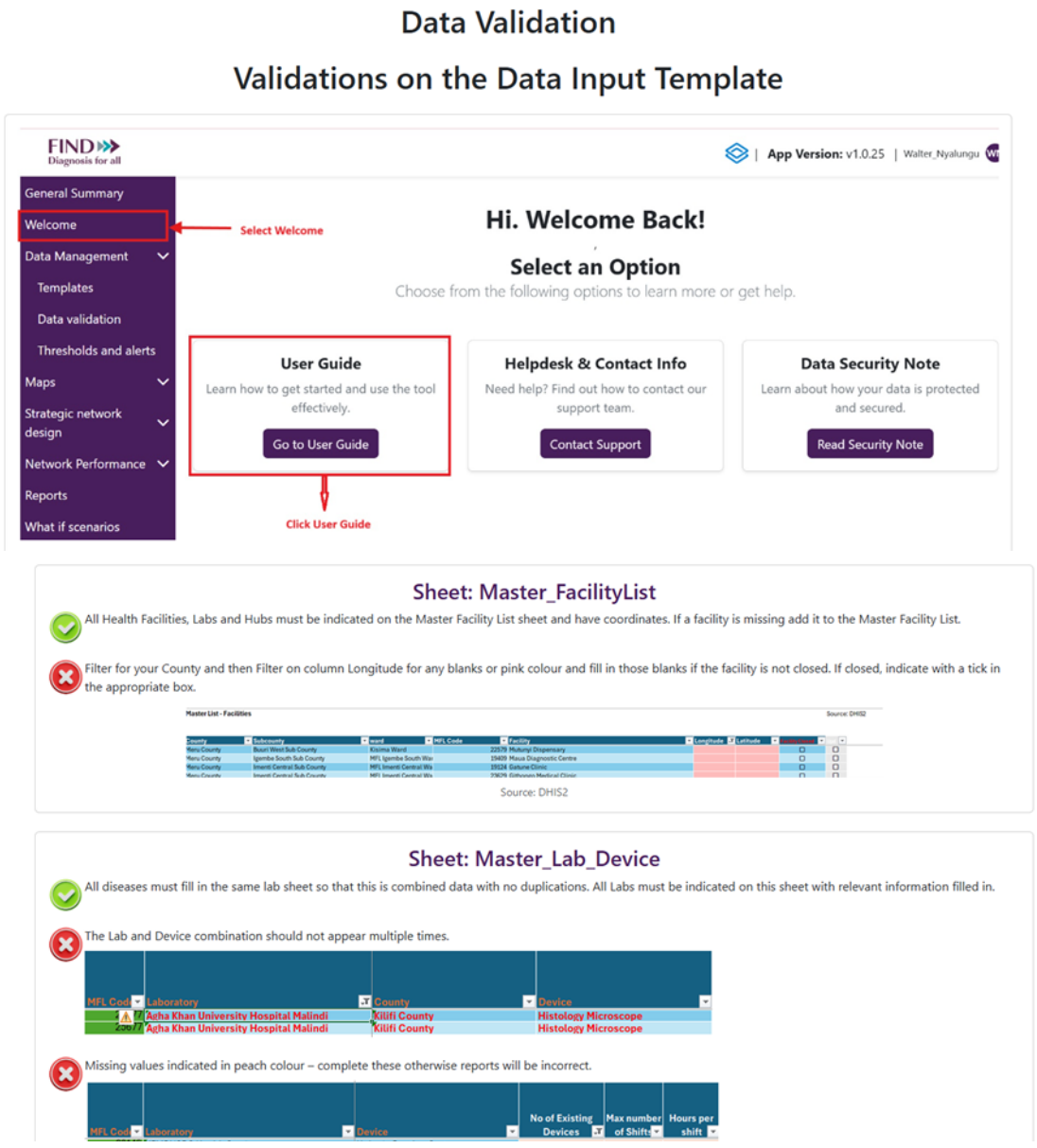


Fig 2.5 – The Data Validation page

Purpose

It provides the user with guidelines on how to input the data into the file in the correct way to avoid errors. The page provides a clear indication of different sheets and how to capture data for columns in the sheet. It then also provides a breakdown table data that provides data for orgUnits that have data but no coordinates.

Technical Design

- *templates/datavalidation.html*
- *app/validation.js*
- *index.html*

Thresholds and alerts

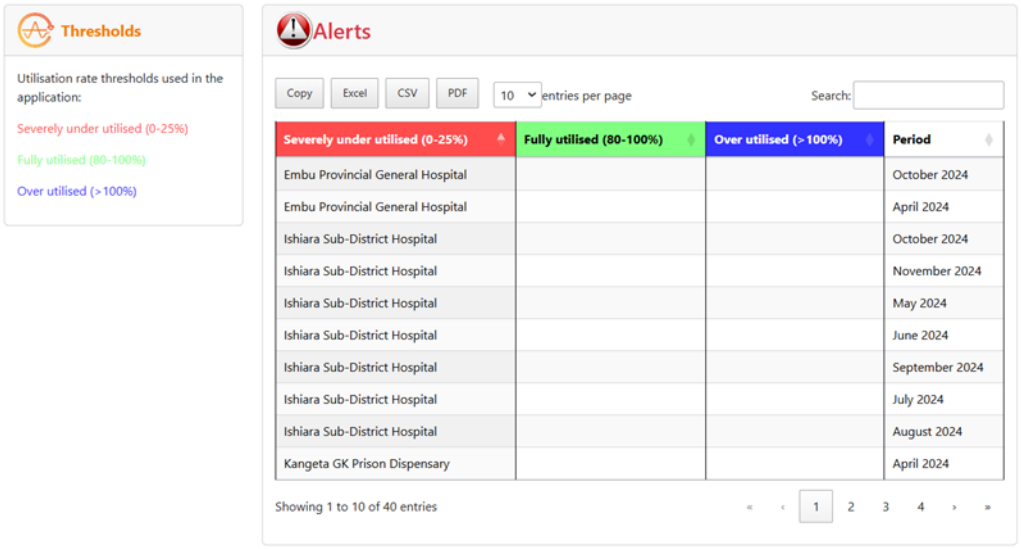


Fig 2.6 – Threshold and alerts page

Purpose

It provides the user with information about facilities that have devices, checking whether those devices are severely underutilized, fully utilized, and over-utilized and the period where this report on utilization was made.

Technical Design

- *templates/thresholds.html*
- *app/thresholds.js*
- *index.html*

11. Maps

11.1 Maps Intro

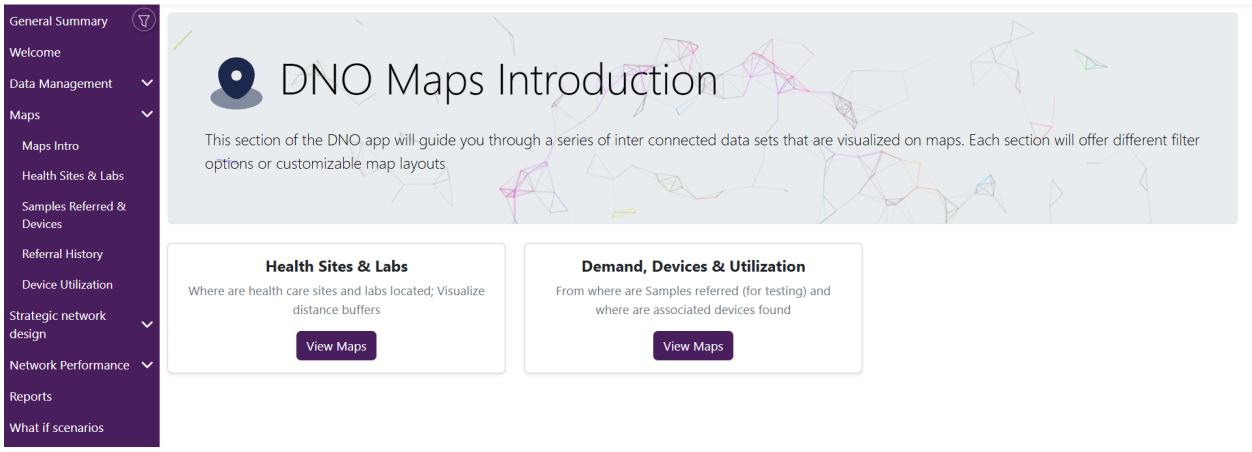


Fig 2.7 – The Maps Intro page

Purpose

The Maps Intro section has been designed to be a landing page for the various custom maps that are available in the app. Shortcuts that appear on that page are:

- **Health Sites & Labs**
- **Demand, Devices & Utilization**

Technical Design

- *templates/mapsIntro.html*
- *app/mapsIntro.js*

11.2 Health Sites & Labs

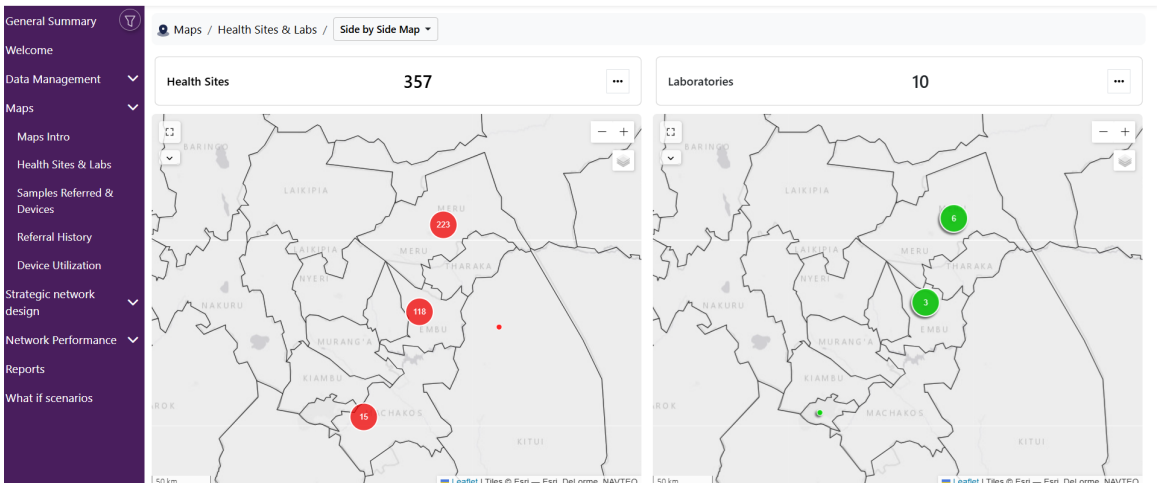


Fig 2.8 – Health Sites & Labs map

Purpose

This is a specialized map designed to plot on a map, all the lab sites within the network (laboratories, clinics and hospitals) as well as laboratories on the right side of split view. The aim of this map display is to show the spread of laboratories within the geographic area where they serve the available health sites (based on the sample referral pathways)

Technical Design

- *templates/mapSitesAndLabs.html*
- *app/mapSitesAndLabs.js*

Dependencies

- *templates/sidebar.html*
- *scripts/treeView.js*

11.3 Samples Referred & Devices

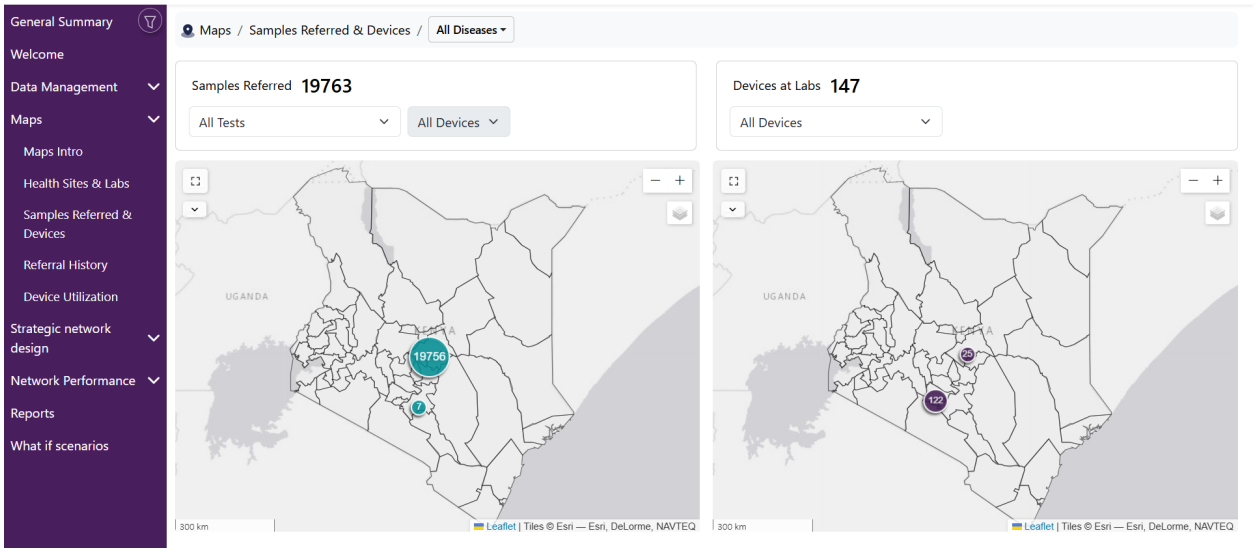


Fig 2.9 – Samples Referred & Devices map

Purpose

The Samples Referred & Devices section is designed to display (demand) which is a distribution of samples referred on the left-side, as well as devices at labs on the right-hand side (capacity). Through this display the user can, at a high level, get a sense of the size of the testing demand vs available capacity in the network.

Technical Design

- *templates/mapsDemandAndDevices.html*
- *app/mapsDemandAndDevices.js*

Dependencies

- *templates/sidebar.html*
- *scripts/treeView.js*

11.4 Referral History

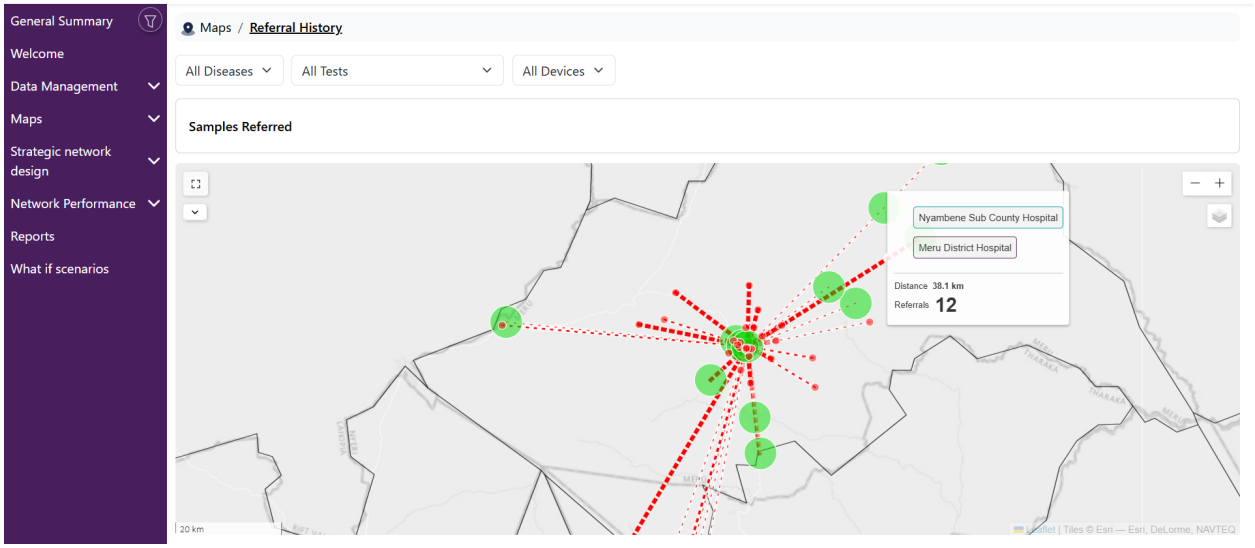


Fig 2.8 – Referral History map

Purpose

The Referral History map is a specialized map that the is at the centre of diagnostic network analysis. The map is designed to display referral pathways based on historical referral data, clearly demonstrating the relationships between facilities that submitted samples for testing, and facilities that received the samples that received and processed the samples. For each referral pathway, the (straight-line) distance between the two facilities is displayed as well as the number of samples referred (for the reporting period in the underlying data set)

Technical Design

- *templates/mapReferralHistory.html*
- *app/mapReferralHistory.js*

Dependencies

- *templates/sidebar.html*
- *scripts/treeView.js*

11.5 Device Utilization

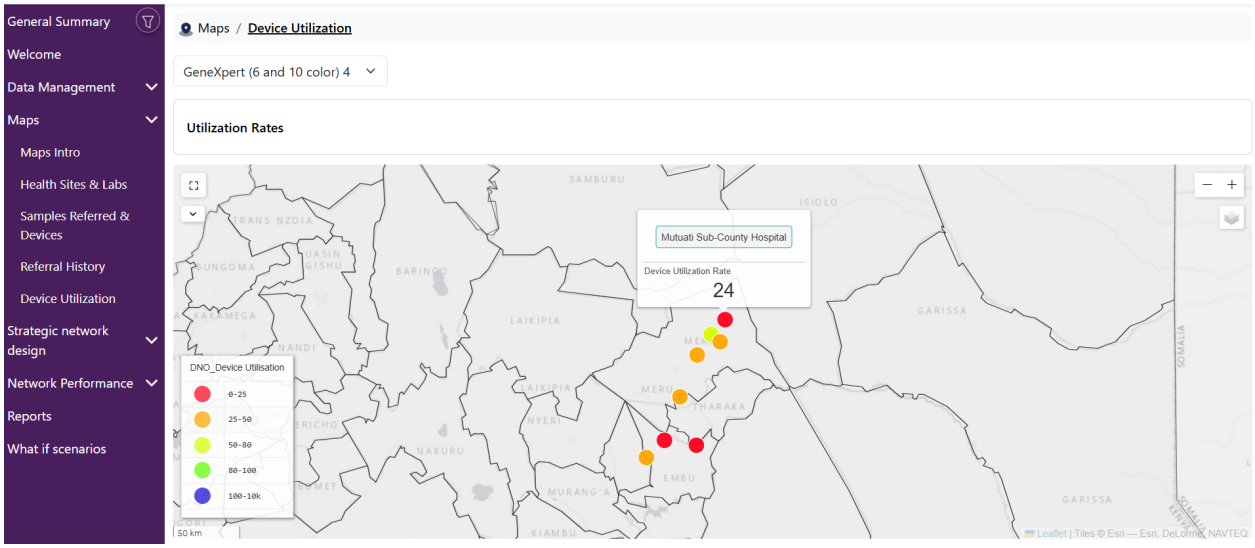


Fig 2.9 – Device Utilization map

Purpose9

The Device Utilization map, in one map displays the impact of testing demand on available devices in the diagnostic network. Using a color-coded sliding scale to measure device utilization from 0% to 100% and beyond, one can get a sense of the state of health of the diagnostic network.

Through this map display one can identify hotspots where there is overutilization of available devices as well as pockets of low utilization, which can prompt the redirection of testing demand (referrals) to such facilities where there is general underutilization of testing devices.

Technical Design

- *templates/mapDeviceUtilization.html*
- *app/mapDeviceUtilization.js*

Dependencies

- *templates/sidebar.html*
- *scripts/treeView.js*

12. Strategic Network Design

12.1 Capacity

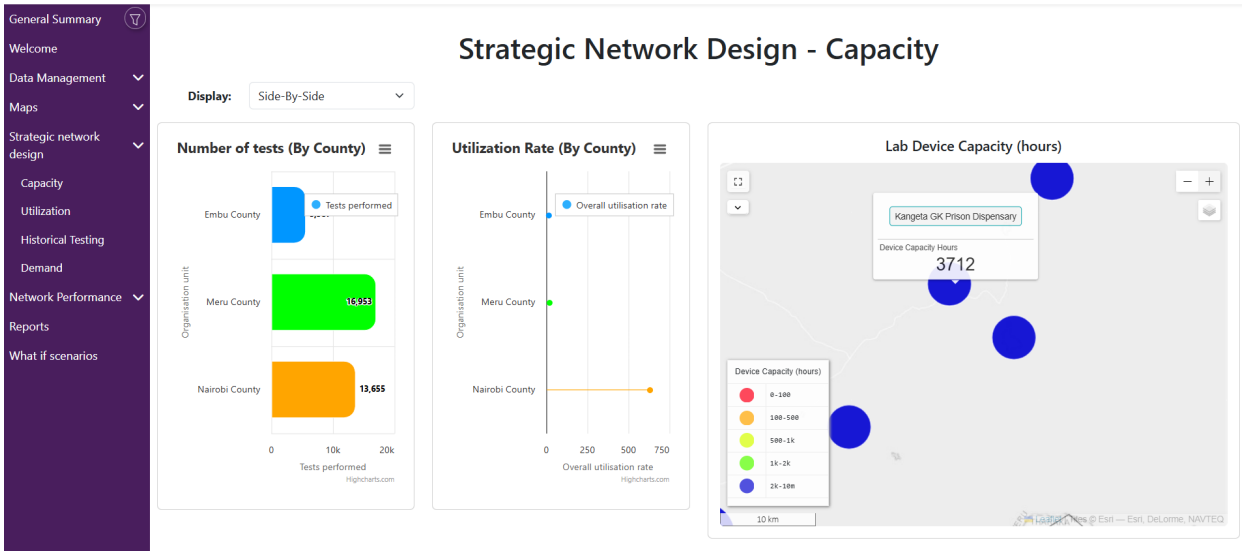


Fig 3.0 – Strategic Network – Capacity section

Purpose

The Strategic Network – Capacity section is designed to provide deep insights into the amount of available capacity in the diagnostic network, which is a key concern usually identified by hospital managers as well as sub district / sub county laboratory managers. Displayed charts and maps in this section include:

1. **(Chart) Number of tests (By health area e.g. county):** This chart is meant to give an indicator of the level of testing demand within the health area.
2. **(Chart) Utilization rate (By health area e.g. county):** This chart provides a measure of average utilization rate of testing devices across the health area related to the testing demand (number of tests done)
3. **(Map) Lab Device Capacity (hours):** This map displays a calculated indicator: "lab device capacity (hours)" which is calculated based on demand, utilization rate as well the general utilization potential of each testing device

Collectively, the charts and maps discussed above can help health facility managers to determine how to best utilize the diagnostic network to ensure that laboratory testing samples are routed to facilitate of high-capacity availability

Technical Design

- *templates/strategicNetworkDesign.html*
- *app/strategicNetworkDesign.maps.js*

Dependencies

- *app/strategicNetworkDesign.charts.js*
- *templates/sidebar.html*
- *scripts/treeView.js*

12.2 Utilization

Purpose

The Utilization section of the Strategic Network Design capability of the DxPulse App is designed to provide rich insights related to the general utilization of the testing devices in the network. Device utilization is approached from 2 vantage points which are facility level device utilization as well as device level utilization

s

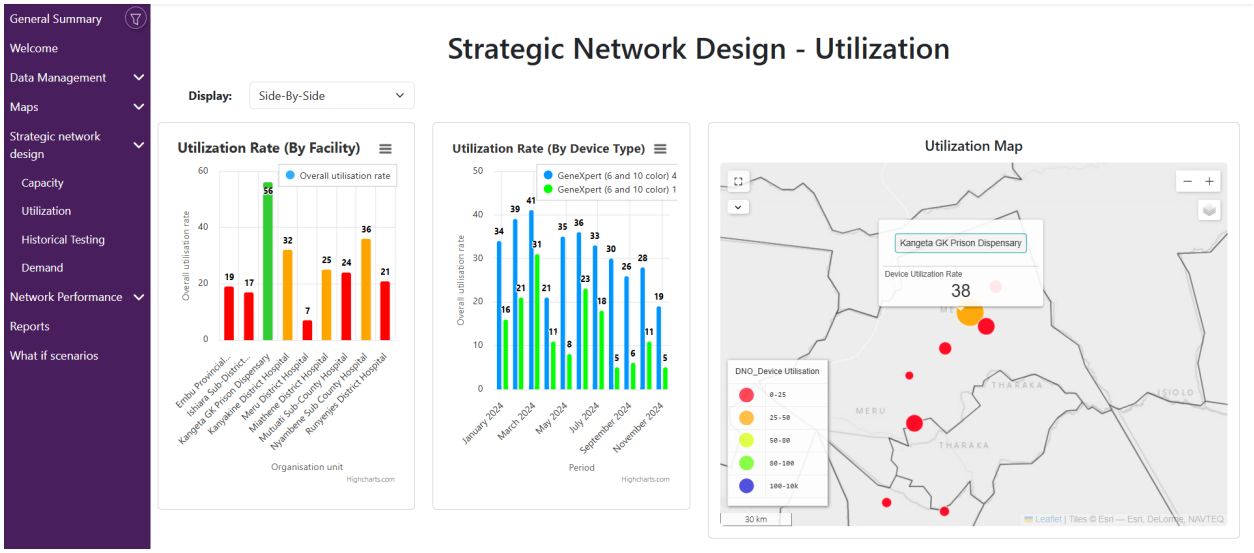


Fig 3.1 – Strategic Network – Utilization section

Technical Design

- *templates/strategicNetworkDesign.html*
- *app/strategicNetworkDesign.maps.js*

Dependencies

- *app/strategicNetworkDesign.charts.js*
- *templates/sidebar.html*
- *scripts/treeView.js*

12.3 Historical Testing

Purpose

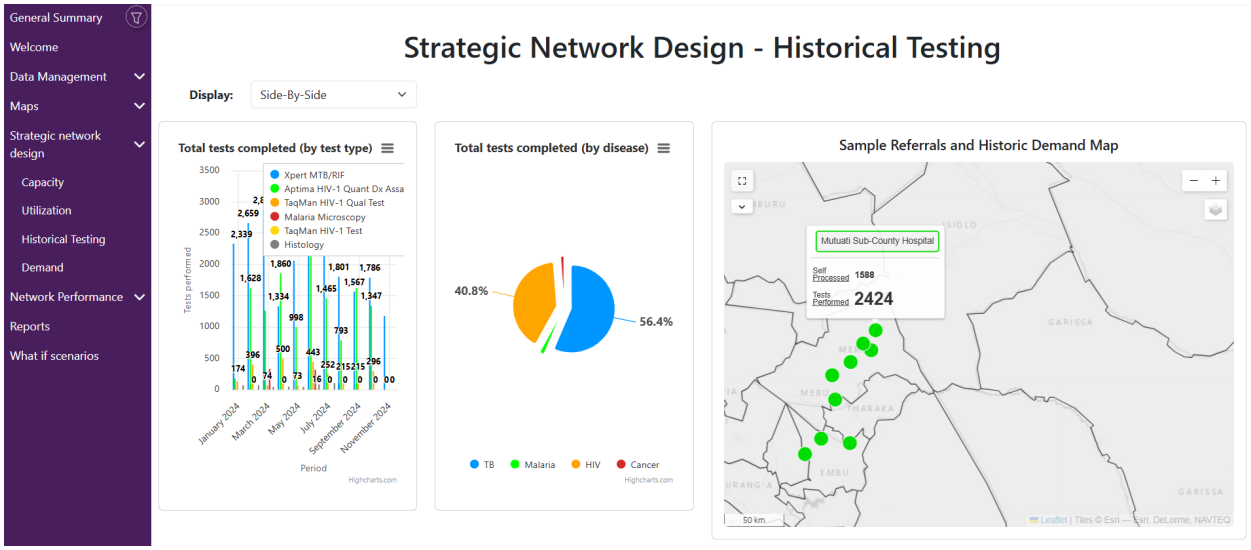


Fig 3.2: Strategic Network – Historical Testing section

The Historical Testing section of Strategic Network Design is designed to provide an evidence-based analysis of the current testing demand for the period under review. Insights provided by historical testing analysis help decision makers to ascertain which disease programs are driving demand, which helps to shape the direction in which resources are channelled. The following charts and maps are displayed in this section:

- **(Chart) Total tests completed (by test type)** – This is a column chart that displays the distribution of tests done by test type over the last 12 months (default period)
- **(Chart) Total tests completed (by disease)** – This is a pie chart showing a summary of all completed tests by disease type over the reporting period (default – last 12 months)
- **(Map) Sample referrals and historic demand** – This is a map display of total number tests done (at laboratory facilities). The plotting of this data on a map helps one visualize the physical location where testing was done across the related counties

Technical Design

- *templates/strategicNetworkDesign.html*
- *app/strategicNetworkDesign.maps.js*

Dependencies

- *app/strategicNetworkDesign.charts.js*
- *templates/sidebar.html*
- *scripts/treeView.js*

12.4 Demand

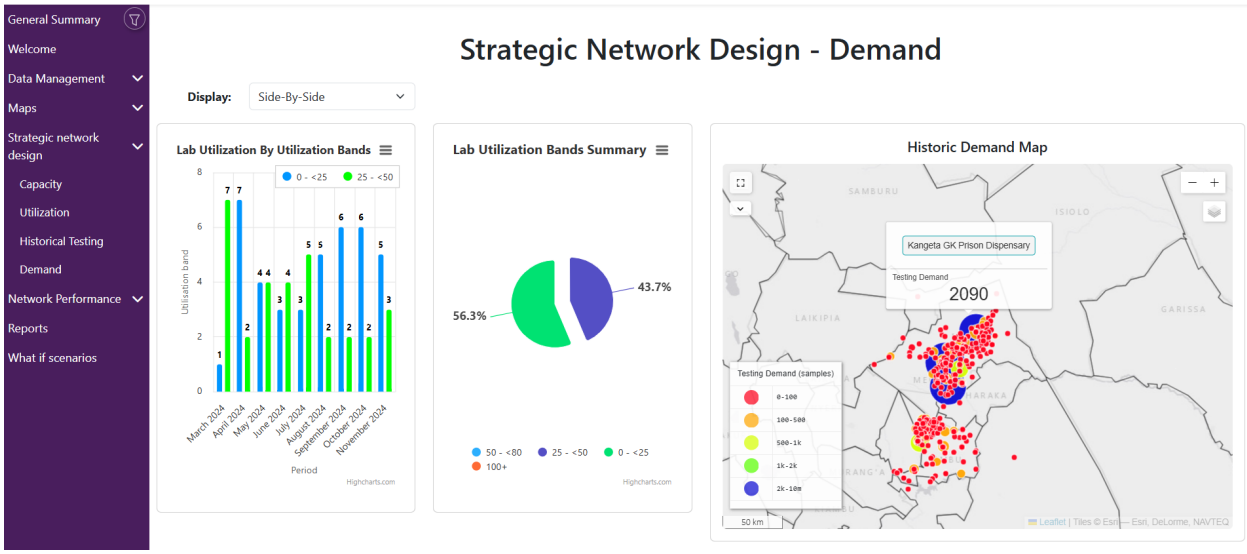


Fig 3.3: Strategic Network – Demand section

The Demand section of Strategic Network Design provides a dashboard that is useful for health facility and laboratory managers to review the utilization rates of laboratories within their network. Insights from this dashboard include utilization rates over the reporting period. Utilization rates as a proportion as well as utilization rates per testing facility. The charts displayed in this section are:

- **(Chart) Lab Utilization by Utilization Band:** This is a clustered column chart displays the utilization rates of testing facilities over a period (12 months by default). This chart helps to analyze utilization tends which may be affected by outbreaks or seasonality
- **(Chart) Lab Utilization Bands Summary:** This is a pie-chart that provides a snapshot of view of composition of testing facilities with regards to the set utilization bands. This chart helps one to see the percentage of highly utilized facilities compared to underutilized ones
- **(Chart) Historical Demand Map:** This map provides bubbled color display or utilization rates of testing facilities according to the provided legend. This map helps one to identify the physical location of underutilized testing facilities as well as overutilized testing facilities to help guide decisions on how to optimally route testing samples

Technical Design

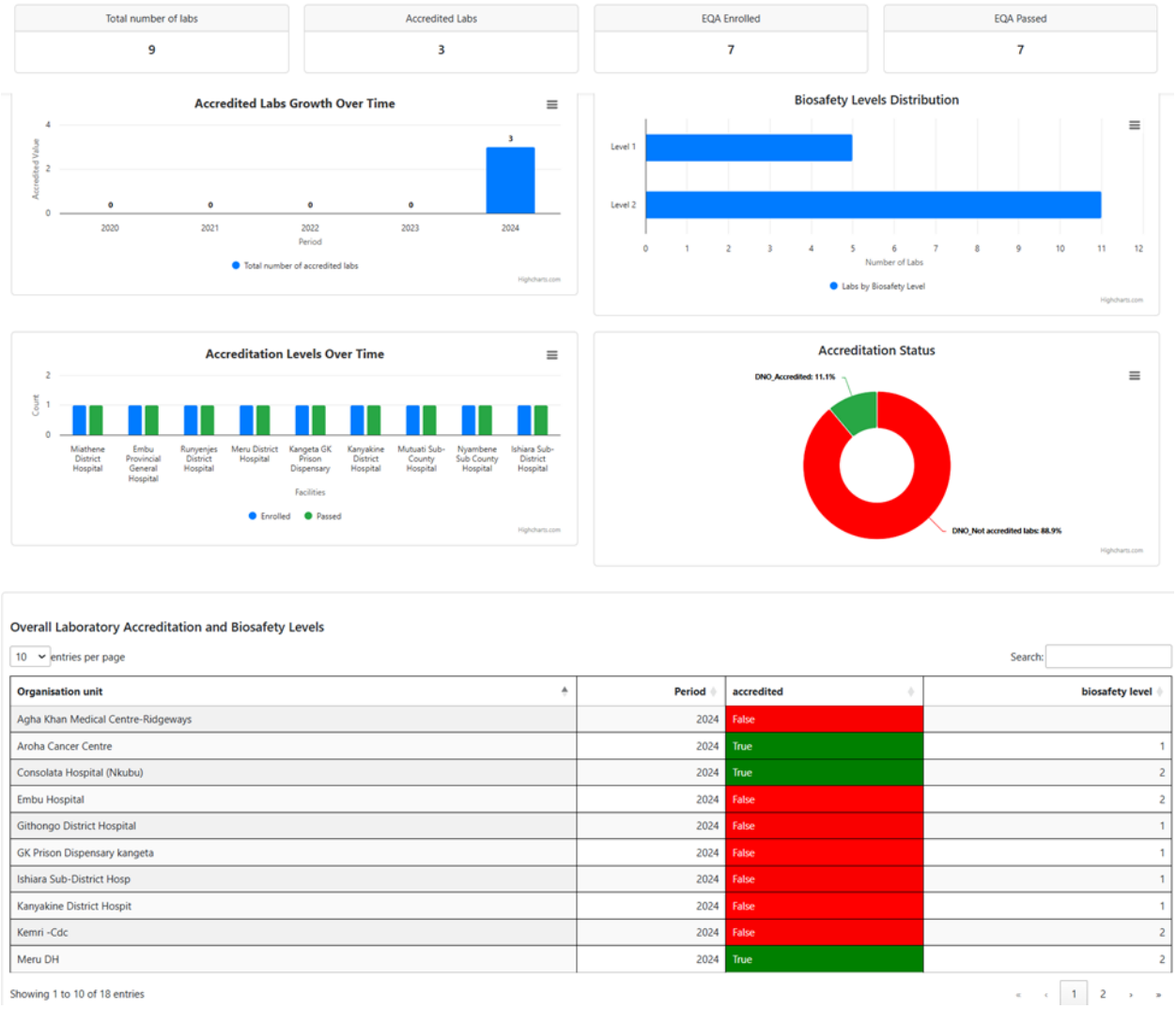
- *templates/strategicNetworkDesign.html*
- *app/strategicNetworkDesign.maps.js*

Dependencies

- *app/strategicNetworkDesign.charts.js*
- *templates/sidebar.html*
- *scripts/treeView.js*

13. Network Performance

13.1 Quality



Overall Laboratory Accreditation and Biosafety Levels

10 entries per page — Search:

Organisation unit	Period	accredited	biosafety level
Agha Khan Medical Centre-Ridgeways	2024	False	
Aroha Cancer Centre	2024	True	1
Consolata Hospital (Nkubu)	2024	True	2
Embu Hospital	2024	False	2
Githongo District Hospital	2024	False	1
GK Prison Dispensary kangeta	2024	False	1
Ishiara Sub-District Hosp	2024	False	1
Kanyakine District Hospit	2024	False	1
Kemri -Cdc	2024	False	2
Meru DH	2024	True	2

Showing 1 to 10 of 18 entries — 1 2

Fig 3.3: Network Performance – Quality section

Purpose

The Quality page serves as a visualization page highlighting how credible the labs are if the labs meet specific standards for Quality focusing on the Accreditation Levels Over Time, Accreditation status, and Overall Laboratory Accreditation and Biosafety Levels. These Visualisations are designed to track, compare, and assess various aspects of accreditation processes, standards, and outcomes for DxPulse.

- **Card Visualization:** Provides a summary of data for the total number of labs, Accredited labs, EQA Enrolled and EQA passed
- **Accredited Labs Growth Over Time:** Highlights or visualizes the lab growth over time highlighting the past 5 years
- **Biosafety Levels Distribution:** Provides an overview to show how various biological research laboratories are distributed across the different biosafety levels which are determined based on the risk posed by the biological agents they handle.
- **Accreditation Levels Over Time:** The visualization has two accreditation statuses, passed and enrolled. Which highlights which districts have passed and enrolled and the count thereof. Passed indicates the district has passed the accreditation review and enrolled indicates that the district is actively enrolled in the accreditation program.
- **Accreditation Status:** This visualization provides the status percentage for accreditation for labs highlighting accredited labs and not accredited labs' overall percentage
- **overall Laboratory Accreditation and Biosafety Levels:** The table visualization looks at the Biosafety level of the district for the current period and provides information on their accreditation status

Technical Design

- · *index.html*
- · *templates/ network_performance_quality.html*
- · *app/ network_performance_quality.js*

The index.html provides a reference to the network_performance_quality.html page. The HTML page serves as the front-end UI for the visualization and all the functionality of the page and calls are done on the .js file.

13.2 Human resources

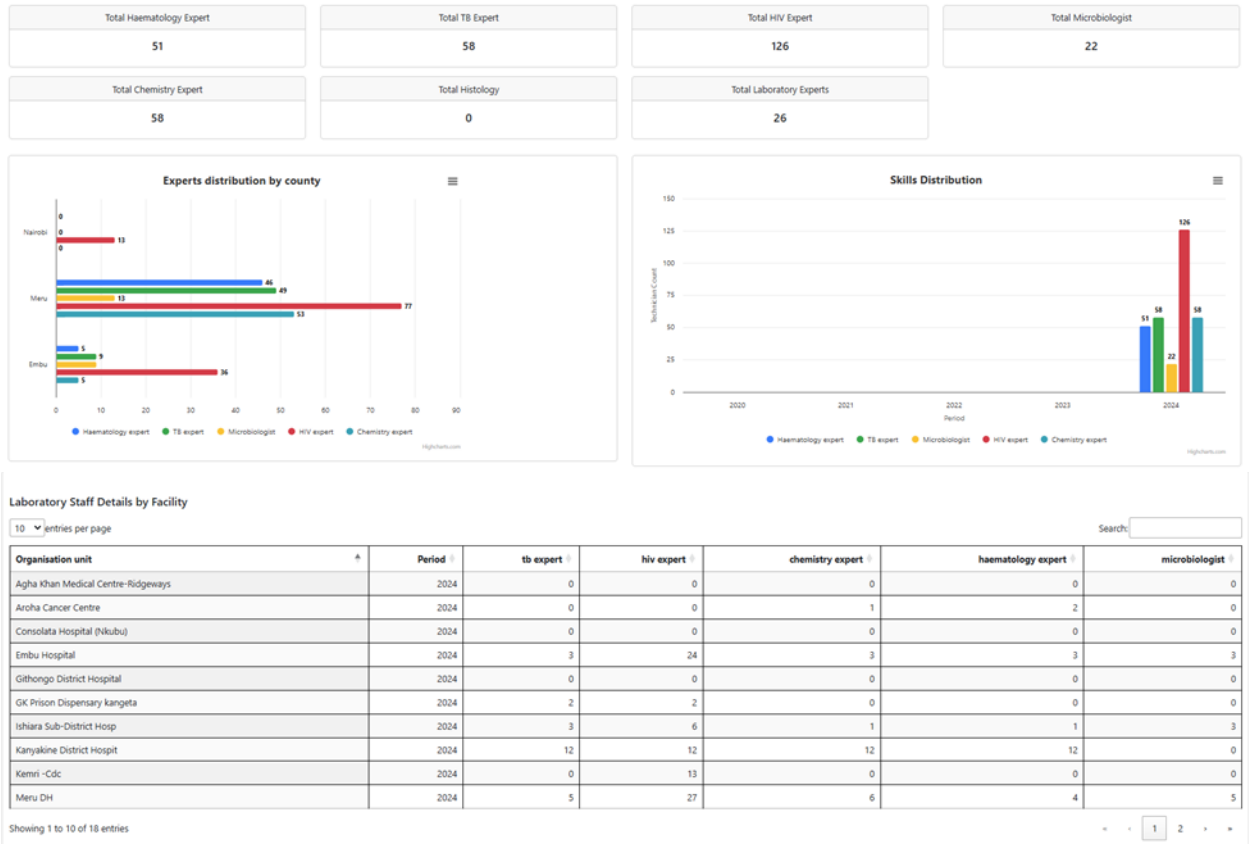


Fig 3.4: Network Performance – Human resources section

Purpose

Human Resources visualization refers to using visual tools and techniques to represent lab experts' distribution, skills, roles, and other critical aspects. Displays the number of experts by their roles looking at TB, HIV, Microbiologist, Chemistry, Histology, Laboratory, and Haematology Experts.

- **Card Visualization:** Provides several experts across different positions looking at TB, HIV, Microbiologist, Chemistry, Histology, Laboratory, and Hematology Experts
- **Experts distribution by county:** Analyses how professionals/experts are spread across different facilities
- **Skills Distribution:** Examines the variation in skills and understanding how these skills are spread out. Understanding the number of technician experts throughout 5 years
- **Laboratory staff Details by Facility:** Focus on the specific types of professionals employed at various laboratories across different facilities. Highlighting a number of those professionals across different fields in a particular facility

Technical Design

- *index.html*
- *templates/ network_performance_hr.html*
- *app/ network_performance_hr.js*

The index.html provides a reference to the network_performance_hr.html page. The HTML page serves as the front-end UI for the visualization and all the functionality of the page and calls are done on the .js file.

14. Reports

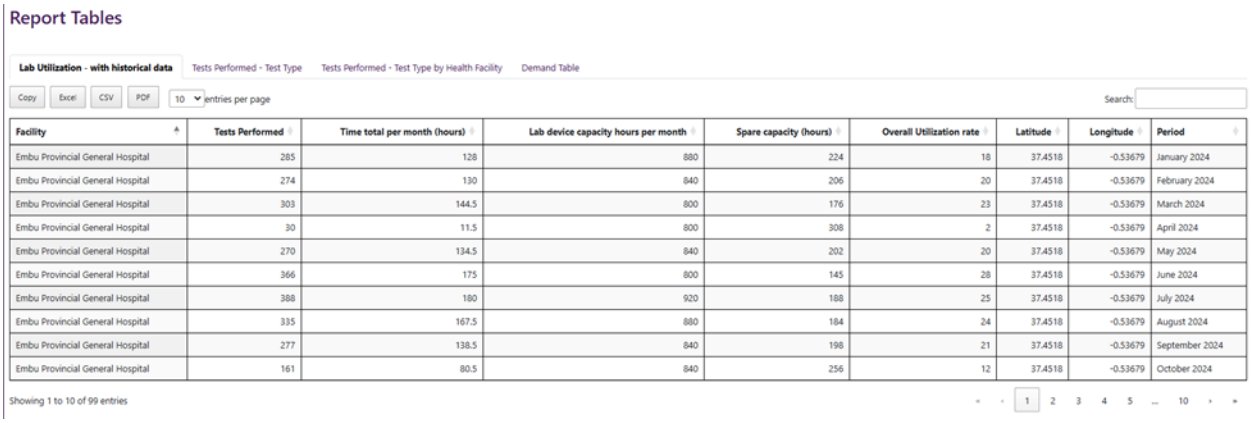


Fig 3.5: Reports section

Purpose

This report provides an overview of lab utilization, test types performed and forecasted demand for facilities. The data includes historical trends, test breakdowns, and future predictions to help administrators monitor performance and optimize lab operations.

- **Lab Utilization -with historical data:**
  - **Tests Performed:** The total number of laboratory tests performed in the given month.
  - **Time Total per Month (hours):** The total amount of time (in hours) the laboratory spent performing tests in that month.
  - **Lab Device Capacity (hours/month)**: The total available capacity (in hours) of lab equipment to perform tests per month.
  - **Spare Capacity (hours):** The unused lab capacity (in hours) available after performing the tests.
  - **Overall Utilization Rate:** The percentage of lab capacity used based on the tests performed compared
  - **Latitude:** The latitude coordinate of the hospital location.
  - **Longitude:** The longitude coordinate of the hospital location.
  - **Period:** The month and year of the recorded data.

- **Test Performed – Test Type:** This table provides a breakdown of test types performed across facilities for a specific duration in 2024. It includes the total number of tests for each test type by month.

- **Test Performed – Test Type by Health Facility:** This table provides the breakdown of tests performed at facilities for each month in 2024. It focuses on several DxPulse tests.

- **Demand table:** Provides details of the test device demand and test volumes at facilities over several months in 2024. It includes GENEXPERT and MICROSCOPEH devices, showing the number of tests performed and their respective demands.

Technical Design

- *templates/ linelistings.html*
- *app/ linelistings.js*

15. What If Scenarios

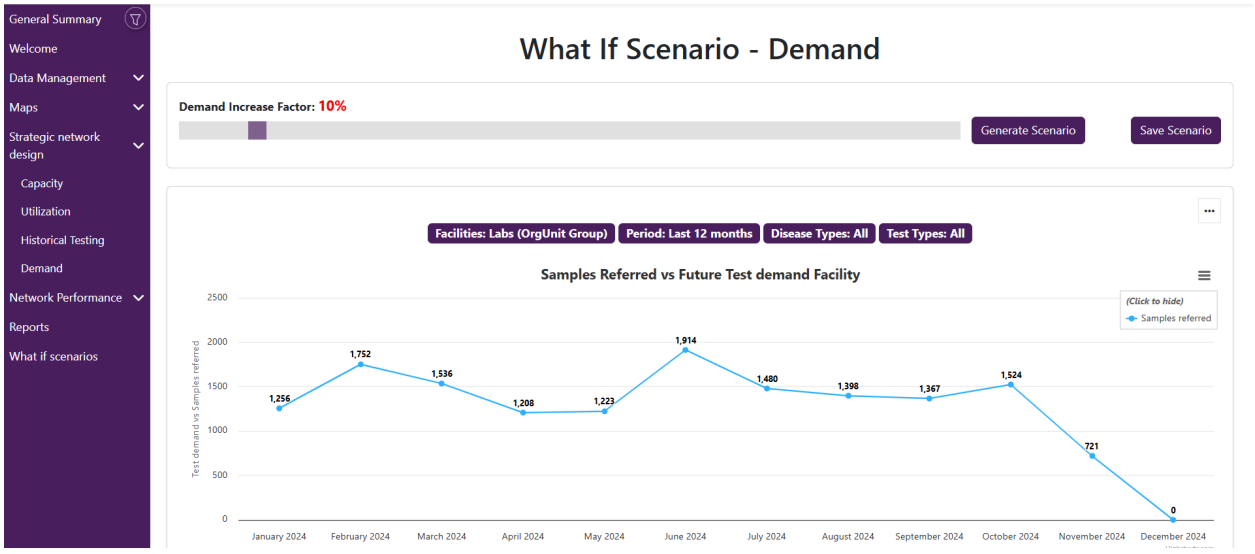


Fig 3.6: What If Scenarios - Demand section

Purpose

Health information system decision makers often need to come up with data driven projection of future scenarios that may potentially materialize if certain conditions are met. The “What if Scenarios” section is a specialized area in the DxPulse App that can help to provide such insights. The “Demand” sub-area in this tool is focused on forecasting how demand could possibly change if there was a net increase or decrease to testing. The tool requires the user to select a “Demand Increase

Factor" to facilitate the calculation by specially designed algorithm embedded in the tool. Once selected, the user simply needs to click the "Generate Scenario" button to allow the calculation to be done, after which the chart display will be updated.

Technical Design

- *templates/what_if_scenarios_demand.html*
- *app/whatIfScenariosDemand.js*

Dependencies

- *app/strategicNetworkDesign.charts.js*
- *templates/sidebar.html*
- *scripts/treeView.js*

16. CD/CD & Code Management

16.1 GitLab

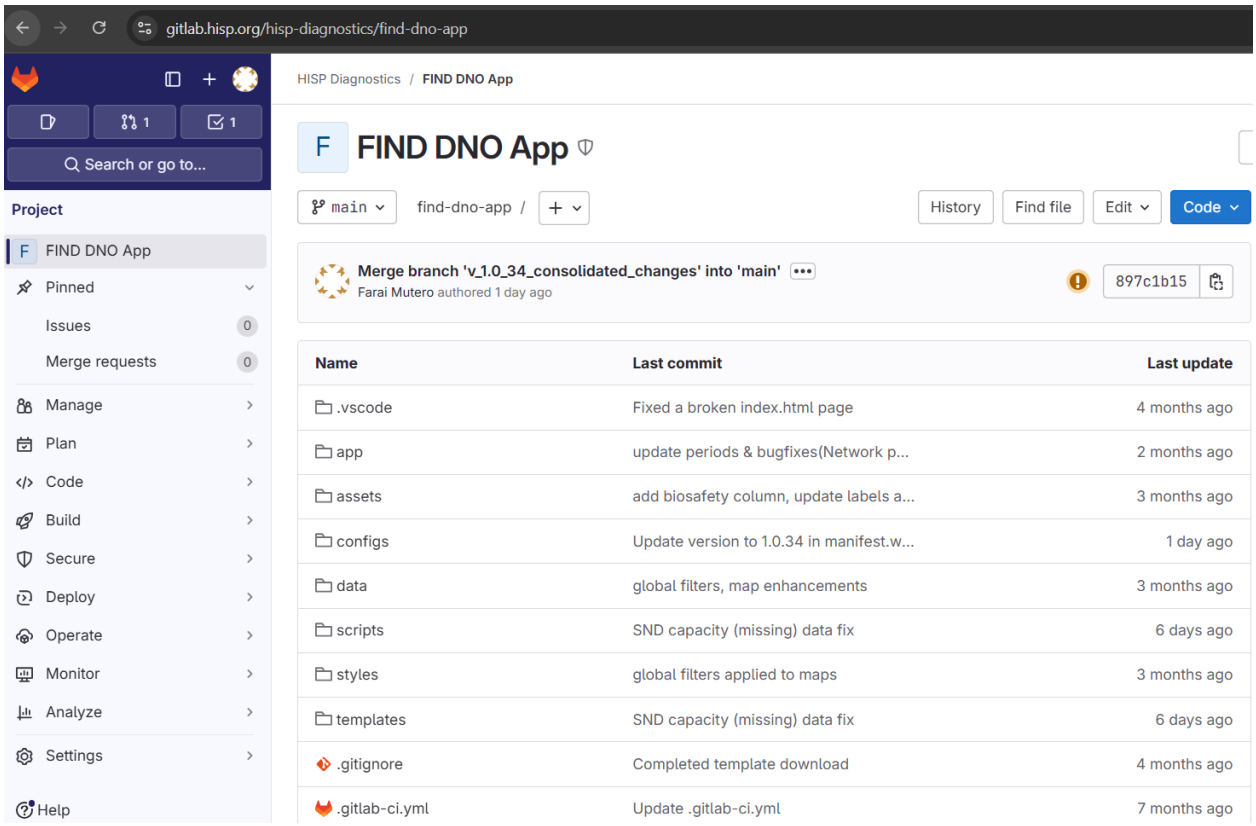


Fig 3.7: Snippet of FIND DxPulse App repo in Gitlab

HISP uses GitLab, which is a web-based DevOps platform that provides a complete suite of tools for software development, version control, and continuous integration/continuous deployment (CI/CD). It is built around Git, a distributed version control system, and offers features such as source code management, issue tracking, code reviews, and collaboration tools. GitLab also includes robust CI/CD pipelines, allowing developers to automate testing, building, and deployment processes.

GitLab can be self-hosted or used as a cloud-based service, making it flexible for teams of all sizes. It supports integration with various third-party tools and provides advanced features like container registry, Kubernetes integration, and security scanning. GitLab's all-in-one approach streamlines the software development lifecycle, enabling teams to collaborate efficiently and deliver high-quality software faster.

16.2 Code Management

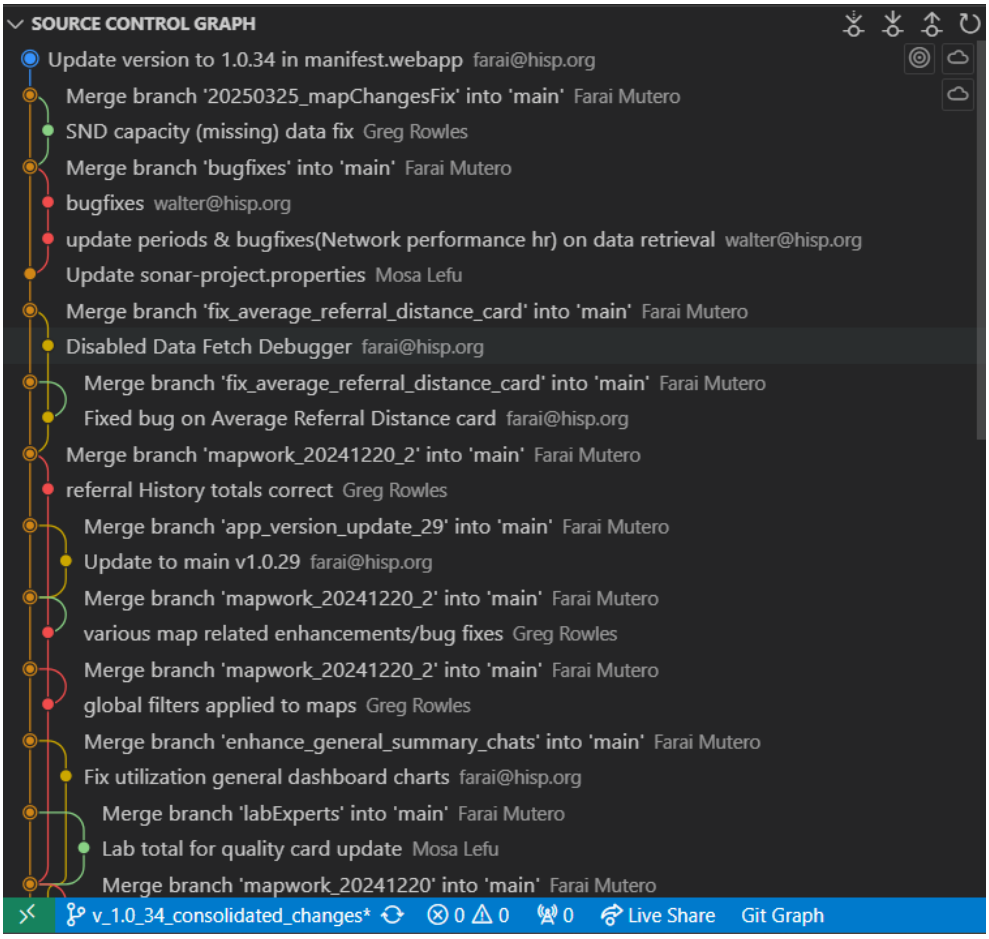


Fig 3.7: Snippet of Source Control Graph tool in VS Code

Best practices were followed with regards to code management during the development of the FIND DxPulse App. Some of the best practices approached used are as follows:

1. **Using a Branching Strategy**:
    - We adopted a clear branching model, during software development. Feature branches for were used for new work, and a main branch was used for integration for production-ready code.
2. **Committing Frequently and Logically**:
    - We encouraged our development team to make small, logical commits that focus on a single task or feature. This made it easier to track changes and debug issues. Developers were discouraged from committing large, unrelated changes in a single commit. An example of this best practice in action is displayed in the figure above.
3. **Write Clear Commit Messages**:
    - Developers were encouraged to use descriptive commit messages that explain the purpose of the change. As much as possible, developers were encouraged to follow a consistent format, such as starting with a verb (e.g., "Add", "Fix", "Update").
4. **Pull and Rebase Regularly**:
    - Developers were encouraged to frequently pull changes from the central repository to keep the branch up to date. The lead developers, occasionally use rebasing to maintain a clean commit history and resolve conflicts early.
5. **Testing Before Committing**:
    - Developers were encouraged to ensure their code is tested locally before committing.
6. **Using Code Reviews**:
    - Code reviews were conducted for all pull requests or merge requests. This ensured code quality, catching potential issues, and promoting knowledge sharing within the team.
7. **Avoiding Committing Directly to Main**:
    - The Gitlab repository was configured to reject direct commits into the main branch. Developers were encouraged to work on feature branches and merge changes into the main branch only after thorough testing and review.
8. **Documentation and Communication**:
    - Maintaining clear documentation for the repository, including guidelines for branching, committing, and merging encouraged. Communication with the team about ongoing work was emphasized to avoid duplication or conflicts.

17. DxPulse App Installation

The FIND DxPulse App is a fully installable web application within the DHIS2 environment. To integrate this app, users must adhere to the standard procedures for installing custom DHIS2 apps, ensuring they follow the guidelines provided in the DHIS2 documentation for smooth deployment and functionality. To install the DxPulse App apply the following steps:

17.1 Zip Folder Contents

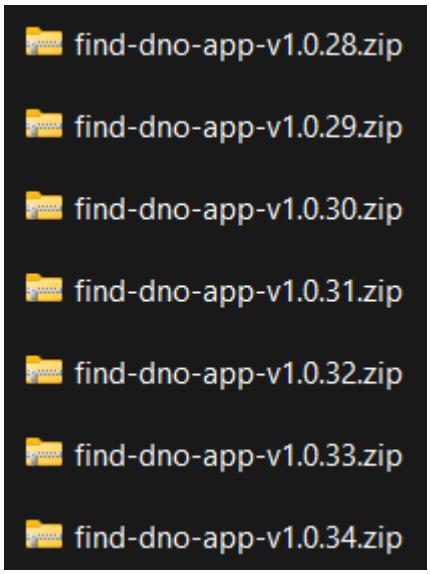


Fig 3.8: Example of deployment ready FIND DxPulse App zip files

Add all the contents the folder to a zip file, except for the following files:

- .vscode
- .htaccess
- .gitignore
- .gitlab-ci.yml
- yarn.lock
- sonar-project.properties

17.2 Login into DHIS2

Login into DHIS2 using the credentials provided. If you do not have the credentials, reach out to the HISP team for assistance.

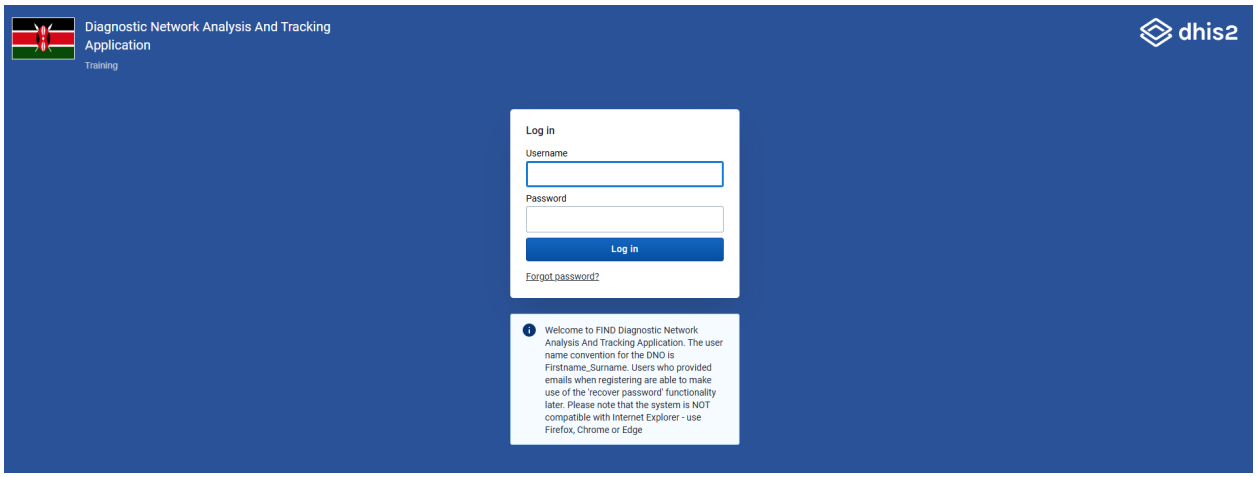


Fig 3.9: DHIS2 Login Page

17.3 Navigate to App Management

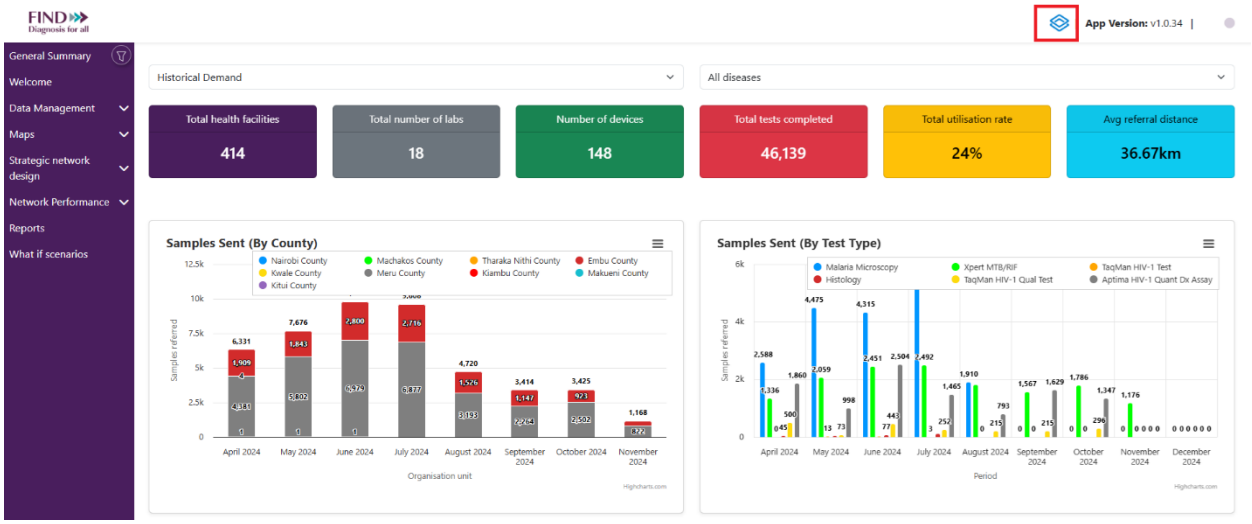


Fig 4.0: DxPulse App landing page

Upon logging in, the DxPulse App may be set as the landing app. In such a scenario, click on the short-cut DHIS2 logo button on the top-right (coloured in red) to navigate to the main DHIS2 dashboard page.

Upon loading the DHIS2 dashboard page, locate and click on the tile icon situated at the top-right corner to access the application menu drawer. Subsequently, select ‘App Management’ to proceed. If this option is not immediately visible, utilize the search bar by entering 'App' to locate it.

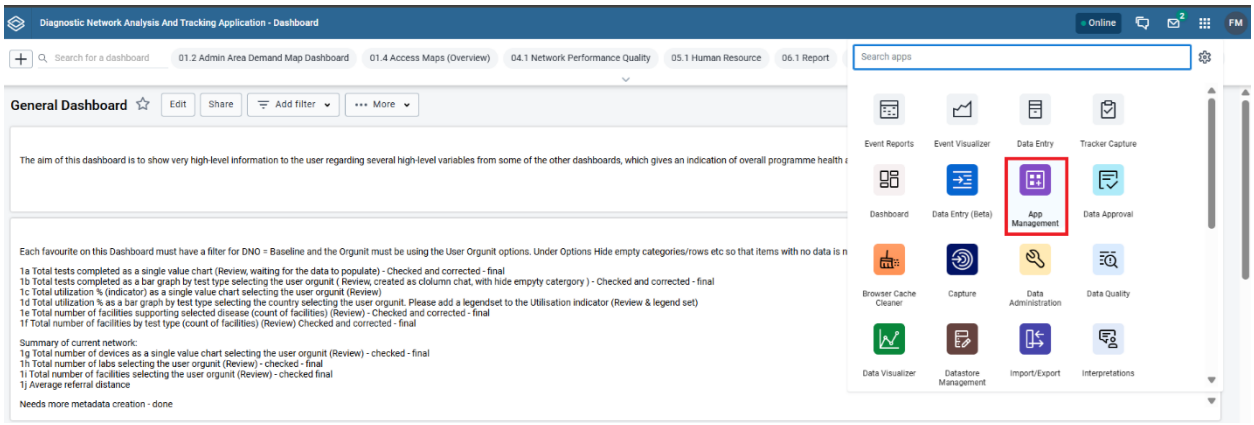


Fig 4.1: DHIS2 Home Page - Dashboard

In the 'App Management' section, observe the available choices: Core Apps, Custom Apps, App Hub, and Manual Install.

- **Core Apps**: These are essential applications that come pre-installed with DHIS2. They include key functionalities required for data management, analysis, and visualization within the system.
- **Custom Apps**: This option allows users to integrate applications tailored to their specific needs. These apps can be developed by internal teams or third-party developers to extend the standard capabilities of DHIS2.
- **App Hub**: The App Hub is a centralized repository where users can browse and install a variety of community-contributed applications. These apps are designed to enhance DHIS2's functionality and are typically shared by other organizations or developers.
- **Manual Install**: This option enables users to manually upload and install applications. It is particularly useful for installing custom-developed apps or updates that are not yet available in the App Hub.

Choose the Manual Install option to continue.

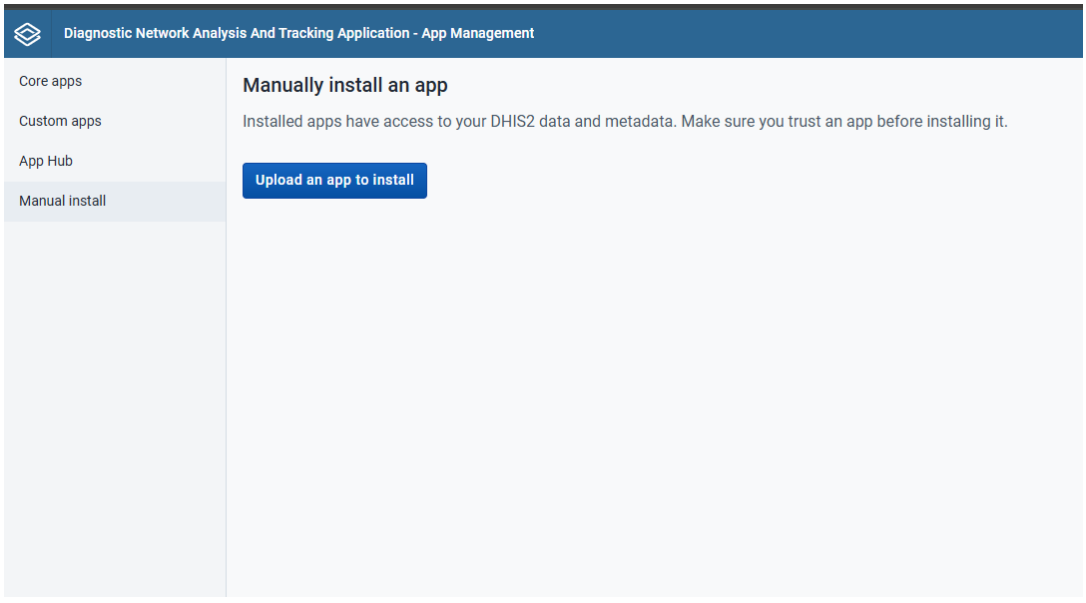


Fig 4.3: **App Management > Manually install an app** snippet

Navigate to the folder / directory where the DxPulse App zip file is located and click on the “Open” button to upload it to DHIS2

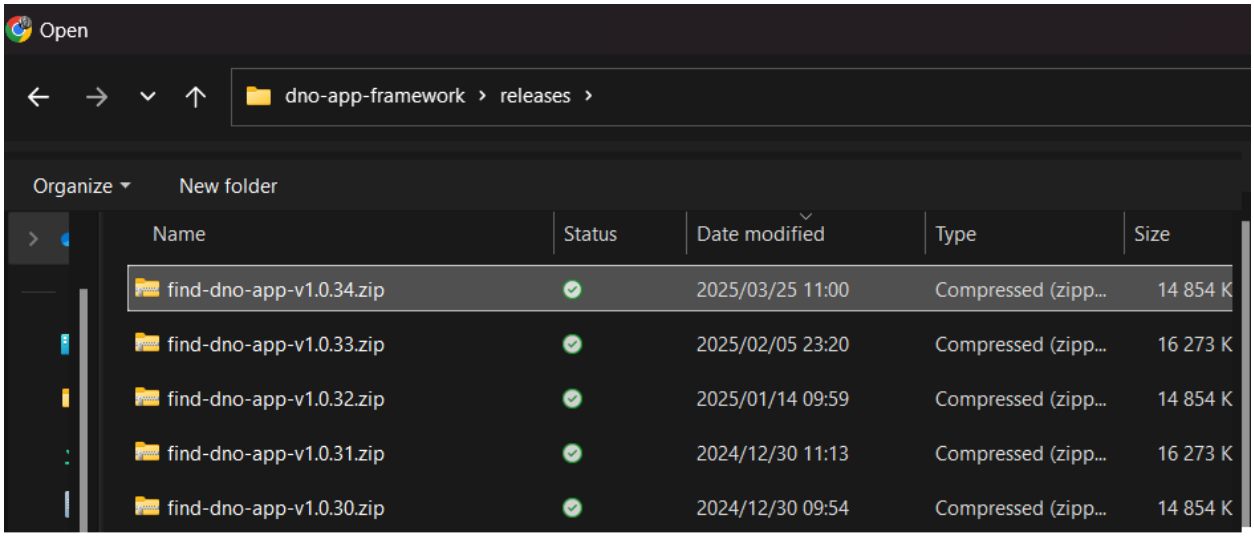


Fig 4.4: Navigating to the folder where the DxPulse App zip file is located

Upon successful installation of the application, a toast message will appear at the bottom of the screen confirming that the installation was completed successfully. If the installation fails, an error message will be displayed with troubleshooting steps.

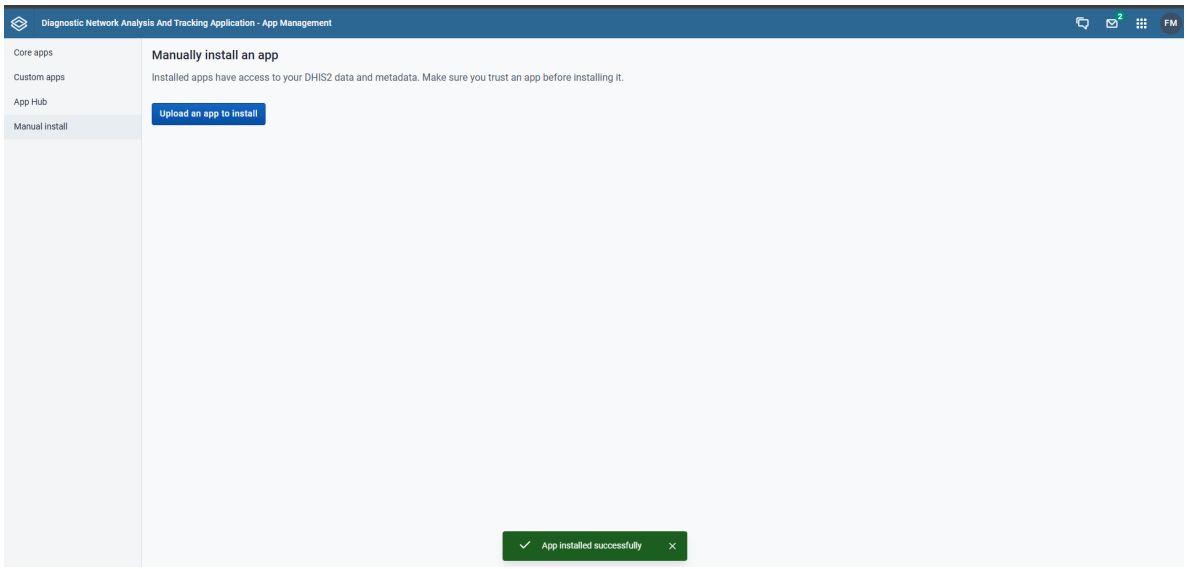


Fig 4.5: Confirming that DxPulse App install was successful

After the app upload is finished, please proceed to the **Custom Apps** section to verify the availability of the app.
Sometimes, the app upload process may encounter issues. Common reasons for failure include:

- **File Format Issues**: Ensure that the uploaded file is in a supported format.
- **Network Connectivity Problems**: Verify that your internet connection is stable throughout the upload process.

- **Authentication Errors**: Make sure you have the necessary permissions and are logged in with the correct credentials.
- **Size Limit Exceeded**: Check that the app file does not exceed the maximum allowed size.
- **Incomplete Metadata**: Confirm that all required metadata fields are correctly filled out before attempting the upload. Especially make sure that the **manifest.webapp** file for the app is included in the zip file

By considering these potential issues, you can troubleshoot and resolve most common problems encountered during the app upload process.

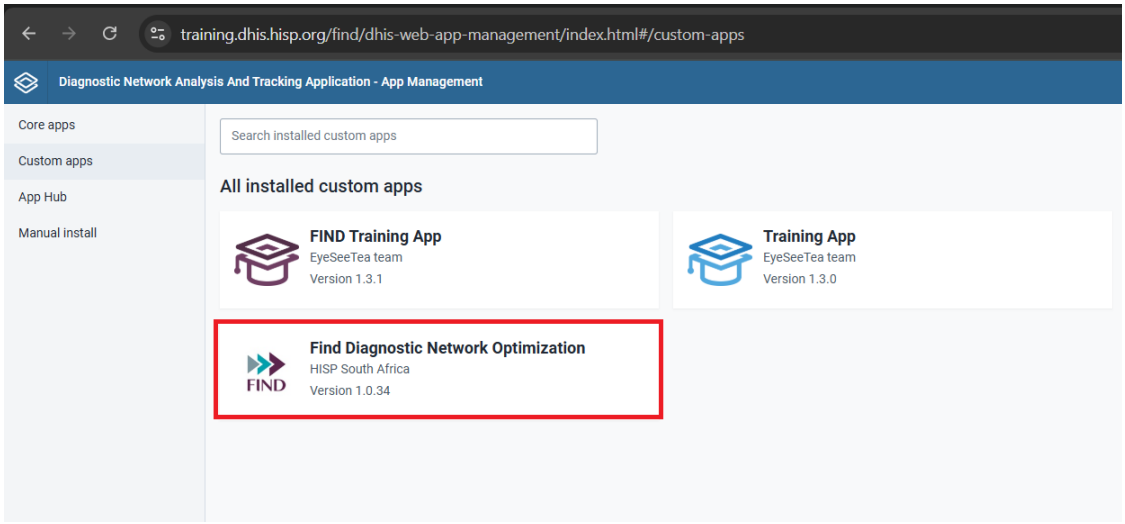


Fig 4.6: Confirming presence of uploaded DxPulse app on DHIS2

18. References

HISP Centre, University of Oslo (UiO). (2024, 12 18). *The Global HISP Network*. Retrieved from DHIS2: https://dhis2.org/hisp-network/?fbclid=IwY2xjawHPc_dleHRuA2FlbQIxMAABHYmw0mguL_frJD-3qlo3Mq8B7tRSBWJh-Ym-L4eSH1gXwejkSCVN4clTHA_aem_QgqyabtEmjNNiYtNKexoAg