Agile development of web applications in resource-constrained scenarios: the case of Medicus Mundi in Mozambique

Francesco Zonaro — francesco.zonaro@gmail.com

July 2024

This document provides a summary of a comprehensive thesis project, outlining the primary research objectives, methodologies employed and key conclusions drawn from the development. Its purpose is to offer a succinct yet thorough overview, highlighting the critical aspects of the project and enabling readers to gain a clear understanding of the work conducted.

1 Introduction

The REACH project (Research for Equitable Access and Continuity of Care in HIV and Tuberculosis) is a vital initiative aimed at enhancing healthcare access and quality for HIV and TB patients in the remote communities of the Inhambane province in Mozambique. This region faces alarmingly high rates of new HIV infections, including pediatric cases from vertical transmission, alongside a significant incidence of tuberculosis. The rural and widely dispersed population, coupled with a low density of healthcare centers, severely hampers access to medical services. These challenges have been further exacerbated by the SARS-CoV-2 pandemic and natural disasters such as cyclones Dineo (2017), Idai (2019) and Freddy (2023). Medicus Mundi Italia (MMI), in partnership with the Fondazione Museke, Universidade SAVE (UNISAVE), the Direcção Provincial de Saúde de Inhambane (DPSI) and under the supervision of the Università degli Studi di Brescia (UNIBS), is implementing the two-year REACH project with the primary goal to combat HIV and Tuberculosis in these hard-to-reach communities through innovative strategies and advanced diagnostic tools.

In alignment with the REACH project's policy of developing and utilizing advanced diagnostic tools, one of the critical aspects involves the tracking of clinical data obtained during patient visits. This approach not only facilitates the provision of high-quality healthcare on the spot but also enables continuous analysis and improvement of the project's efficacy based on the collected data. Currently, the collection and management of clinical data relies predominantly on paper forms and Excel spreadsheets, with healthcare providers, namely the Brigadas Moveis (BM) and TARV Moveis (TM) mobile units, documenting patient information manually during their visits. Additionally, another goal of this data collection is to create a database that can be analyzed to track disease trends over time. For instance, by monitoring the frequency of visits and the prevalence of certain symptoms, healthcare providers can detect outbreaks of HIV, TB or other infectious diseases at an early stage. Furthermore, tracking vaccine distribution helps in assessing the coverage and effectiveness of immunization campaigns. Finally, in the following years, properly structured data could be used to conduct additional in-depth research on the health dynamics of the Inhambane province, studying the correlation between environmental factors and disease prevalence or evaluating the long-term outcomes of different treatment protocols. Currently, the data collected on paper forms is subsequently transcribed into Excel spreadsheets and stored in shared folders for further analysis and reporting. The reliance on paper for data collection is a requirement mandated by the Mozambique government. However, to mitigate inefficiencies and the potential for data loss associated with the physical handling and transfer of paper documents, it is crucial to minimize their use. While Excel spreadsheets are commonly employed for data storage, there is a pressing need to streamline the data collection and management process. A potential solution is the gradual integration of digital tools, ensuring compliance with government regulations while reducing reliance on Excel sheets.

2 Requirements

This chapter outlines the essential specifications and functionalities required to enhance data collection and analysis for the REACH project. It provides a comprehensive description of features and technical constraints.

2.1 Analysis of Excel sheets

The new system aims to leverage the strengths of current methodologies while enhancing efficiency, accuracy and collaboration capabilities among all Medicus Mundi stakeholders. On this note, the involvement of domain experts in the development phase has been paramount. By engaging domain experts, we aim to validate the proposed functionalities at an early stage, enabling timely adjustments and refinements during the development phase and preempting potential issues arising during system deployment. One important premise of the project is that the development of the web application is progressing in tandem with the ongoing activities of the REACH project, which means there is a need for a solution that aligns with the existing work methodology of on-territory volunteers. Prior to the web app creation, the whole methodology relied solely on Excel spreadsheets, both for data recording and for performing the subsequent analysis. On this note, extensive data was provided by Medicus Mundi as the foundation for the project development, comprising multiple spreadsheets exemplifying the data collection and analysis process in the Morrumbene district. Upon analyzing the initial dataset, several key insights emerge:

1. **Territorial Division**: The province of Inhambane is divided into districts, including Morrumbene, for which we have the example spreadsheets. Each district is further subdivided into areas and each area contains multiple villages. It is important to note that the names and boundaries of villages, areas and districts may change over time based on government decisions. Therefore, the system must be designed to maintain a flexible and adaptable relationship between each territorial category to accommodate these potential changes throughout the project's duration.

2. **Temporal Division**: Analysis of the Excel spreadsheets reveals data collection occurs periodically, with roughly monthly intervals. Notably, the periodic division does not strictly align with the calendar months but instead commences and finishes on specific dates as mandated by local government regulations. Furthermore, the data is also aggregated into trimesters and semesters, providing different levels of temporal granularity for the subsequent analysis.

3. **Categorical Division**: The data is organized into modules under four main supercategories: CLINICA, PAV, PRN, and SMI. Each supercategory contains specific categories, and the system must be designed to flexibly accommodate new additions, such as an upcoming Tuberculosis supercategory that will be tracked starting in 2025. Indicators within these categories have properties that denote their active or inactive status, with inactive indicators retained for historical reference but excluded from data collection processes. Additionally, indicators are classified as either non-computed, representing raw data, or computed, representing aggregated values such as sums, differences or averages.

2.2 Main functional requirements

To ensure the web application meets the needs of its users, several functional requirements must be addressed. These requirements will guide the development and ensure the application is secure, efficient and user-friendly.

2.2.1 Data organization

Users with modification permissions can perform CRUD (Create, Read, Update, Delete) operations on administrative information such as provinces, districts, areas and villages. Each location can be managed within its specific category or its broader super-location hierarchy (e.g., villages within areas, areas within districts). The system maintains relationships between these entities, even if locations are moved. For example, if a village is transferred from one area to another, the previous historical relationship is preserved. This applies similarly to areas within districts. This capability ensures that any administrative change made by the Mozambique government is accurately reflected while maintaining the integrity of historical data. Additionally, users can perform CRUD operations on indicators. These indicators are systematically organized within categories, which are further nested under supercategories, ensuring precise categorization and efficient management. Additionally, indicators can be activated or deactivated as required. Deactivating an indicator prevents it from being displayed during module compilation, while still retaining all references and historical data in the database. Indicators may be classified as either single or computed. Computed indicators are associated with a specific formula and can be linked to their child indicators, allowing for more complex data calculations.

2.2.2 Measures entry and management

Efficient data entry and management are critical requirements for healthcare providers, who must be able to enter and update clinical data seamlessly. In the resource-constrained environment of Medicus Mundi, unique challenges necessitate a tailored solution. For healthcare providers, the ability to collect and manage clinical data directly on local devices is crucial due to intermittent internet connectivity and a preference for familiar tools, such as Excel. To address these needs, we have developed an integrated web application that leverages Excel files for offline data management. This system enables users to work with data locally on their PCs using Excel, allowing for continued data collection and management even without internet access. When connectivity is restored, users can import the locally gathered data into the application. This hybrid approach allows data collectors to gather information in a manner that best suits their workflow—whether on paper during field visits or on their local machines using familiar software. The ability to import the collected data into the system at a later stage ensures that healthcare providers can maintain an efficient and reliable data management process despite connectivity challenges.

Features

- Visualize the list of visits and their status as completed or pending.
- Detailed count of the number of indicators missing for each module to be completed.
- Import new visit data, either by uploading a folder or a single Excel file.
- Visualize the measures collected for each visit.
- Aid identification of computed indicators with a different background color.
- Merge indicators that share a common part of their name in the tabular view containing the collected data.
- Automatically aggregate and update computed indicators.

2.2.3 Excel Integration

In the integrated web application, users have specific capabilities to generate, download, and delete Excel files, which serve as templates for data collection. User roles and permissions are designed to ensure that only authorized personnel can modify the templates, while all users can access and download the necessary files. The system enforces that an Excel file cannot be generated for a specific administrative year if one has already been generated for that year. To generate a new file for the same year, the existing file must first be deleted from the system. However, this deletion is only permitted if there are no visits already linked to the existing Excel version, ensuring data integrity and consistency within the system.

2.2.4 Analysis and Reporting

The Analytics and Reports Page is designed to provide users with the ability to generate comprehensive reports and perform detailed data analysis. The primary requirements for this page include the capability for generating reports based on selected parameters, with support for export in PDF format. Users must be able to aggregate data by different geographical levels, including province, district, or area, with dynamic adjustments to the aggregation parameters, allowing for refined data views as needed. Furthermore, the system should enable users to select the administrative year for which the data should be reported, allowing for comparisons across different administrative years to facilitate trend analysis and historical comparisons. Additionally, the system should be designed with the flexibility to incorporate additional aggregation options in the future, based on evolving user needs and data structures.

3 Technology Stack

In this chapter, we discuss the technology stack employed and provide detailed insights into the actual implementation process, illustrating how technologies interrelate to create a robust, scalable and user-friendly application.

3.1 Laravel

Laravel is a PHP framework designed to make the development process easier and more efficient. The primary goal of Laravel is to enable developers to write clean, maintainable and efficient code, reducing the complexity of common tasks such as routing and authentication. Its key features include:

- **MVC Architecture**: Laravel uses the Model-View-Controller (MVC) pattern, which is a design pattern that divides an application into three interconnected components:
  - **Model**: The Model represents the data layer of the application. It is responsible

for retrieving data from the database, performing business logic and returning the processed data. In Laravel, the Eloquent ORM facilitates this interaction by providing a straightforward and efficient way to query the database.

  * **View**: The View is the presentation layer. It is responsible for displaying the data provided by the Model to the user. Laravel uses the Blade templating engine to build the views. Blade provides a simple yet powerful syntax for creating dynamic web pages, allowing developers to embed PHP code within their HTML.

  * **Controller**: The Controller acts as an intermediary between the Model and the View. It handles the user input, processes it (by interacting with the Model) and returns the appropriate response (by rendering a View). Controllers in Laravel are responsible for defining the application's behavior by connecting user requests to the corresponding business logic.

  The MVC pattern fosters a clear separation of concerns, enhancing modularity, maintainability and testability of the code. Models, Views and Controllers are kept in separate directories, following a consistent structure. This organization streamlines navigation through the codebase, simplifying the development process and keeping the code highly maintainable. This is especially useful in our case, where the maintenance will not be executed by the original developer of the website.

- **Artisan CLI**: Artisan is Laravel's command-line interface, offering a set of commands that assist in building and managing Laravel applications. It can automate repetitive tasks, such as database migrations and code generation. We will mention the usage of some commands throughout this chapter.

- **Eloquent ORM**: Eloquent provides an elegant, Active Record implementation for working with databases. It allows developers to interact with their database objects and relationships using an expressive and intuitive syntax.

Finally, the Laravel documentation is meticulously maintained, providing clear and comprehensive guidance on leveraging the framework's capabilities. All these elements together prompted us to choose Laravel as our development framework.

3.2 Eloquent ORM

Eloquent is an Object-Relational Mapper (ORM) included with Laravel: its purpose is to allow developers to interact with the database in an object-oriented manner. Its main features are:

- **Active Record Implementation**: Eloquent uses the Active Record pattern, which ties the database records directly to the objects, making CRUD operations straightforward.

- **Simplicity**: Eloquent simplifies database interactions by using models to represent database tables, allowing common database operations with minimal code. This abstraction reduces the complexity of SQL queries, making database management more intuitive.

- **Readability**: Code becomes more readable and maintainable due to the use of models and relationships.

Models in Eloquent are created using the `php artisan make:model` command, leveraging the Artisan CLI mentioned in Section 3.1. Eloquent models follow conventions such as naming the table after the plural form of the model name. These conventions can be customized by overriding default properties and methods in the model class, but in general Laravel's *convention over customization* approach has proven useful throughout the development of this work.

3.3 MySQL

MySQL is a relational database management system (RDBMS) that allows for the efficient storage, retrieval and management of data. An integral part of many web applications, ours included, MySQL provides a structured way to store data in tables, which can be queried using Structured Query Language (SQL). Its main features include:

  – **Open Source:** MySQL is open-source software, which means it is free to use and has a large community for support.

  – **High Performance:** Known for its speed and reliability, it is capable of handling high volumes of data and complex queries efficiently.

  – **Ease of Use:** With tools like phpMyAdmin, managing and visualizing databases is straightforward.

In the context of our web application, as already mentioned in Section 3.2, Laravel's Eloquent ORM abstracts many of the SQL operations, allowing us to interact with the database using an object-oriented approach, eliminating the need to write raw SQL.

3.3.1 Migration Files

Laravel migrations are a feature of the Laravel framework that allows developers to define and manage the database schema writing code. Migrations provide a way to version control the database schema, making it easy to share and apply changes across different environments. Each migration file contains instructions for creating, modifying or dropping database tables and columns. Creating migration files in Laravel is straightforward leveraging the Artisan CLI. For example, to create a new migration for the 'users' table, the following command was used:

```
php artisan make:migration create_users_table
```

Once the migration file is created, we define the schema changes by editing the file. Each migration file includes two methods: `up()` and `down()`. The `up()` method contains the instructions for applying the migration, while the `down()` method provides the instructions for reverting the migration. After defining the schema changes, migrations are executed using the command

```
php artisan migrate
```

which applies all pending migrations to the database, creating or modifying the tables as defined.

3.3.2 Seeding the Database

Database seeding is the process of populating the database with initial data. Laravel provides a convenient way to seed the database using seed classes, allowing us to prepopulate essential data from the start. To create a new seeder, the following Artisan command was used:

```
php artisan make:seeder DatabaseSeeder
```

In the seeder class, we defined the data to be inserted into the database. For example, the default user for our testing environment has been inserted as

```
public function run()
{
    User::create([
        'name' => 'Test User',
        'email' => 'testEmail@gmail.com',
        'password' => bcrypt("Test@password7"),
        'role' => 1
    ]);
}
```

In order to establish a robust and comprehensive initial dataset for the application, the `DatabaseSeeder` class is employed to run a series of seeders that populate various tables with essential data. The `AdministrativeSeeder` class is responsible for populating the database with administrative entities such as provinces, districts, areas and villages, and for creating the relationships between them. This seeder starts by creating a province named 'Inhambane' and then creates several districts within this province, namely Funhalouro, Massinga, Panda, Morrumbene and Homoine. For each district, it loads associated areas and villages from corresponding CSV files. These CSV files, provided by Medicus Mundi, contain rows where the first column is the area name, followed by village names in the subsequent columns. Each area is linked to its district, and each village is linked to its respective area. The `IndicatorSeeder` class is designed to populate the database with all the required indicators. This script is designed to import data from one of the Excel files provided by Medicus Mundi. It systematically extracts information related to supercategories, categories, and indicators commonly tracked. This data is then efficiently stored in the database for further use by personnel. The `ExcelSeeder` class utilizes the `ExcelService` to populate temporal schedules (Periods) and create Excel templates for all villages. The `ExcelService` starts by creating an Excel record for the specified year and then generates periods for each month, starting from January and ending in December. These periods are defined by start and end dates, not necessarily equal to commonly known calendar months. After this, the service creates a working sheet for each category. Each worksheet contains data for active and non-computed indicators. After being created, seeders can be executed using

```
php artisan db:seed
```

which runs all seed classes, populating the database with initial data.

3.4 RESTful API

3.4.1 Routing and Controllers

Routing in Laravel is a straightforward process that involves defining the URLs that the application should respond to and the corresponding controller actions. Laravel provides a routing system that is both flexible and expressive, allowing us to easily manage the routes of our application. Routes are typically defined in the 'routes/web.php' file for web routes or 'routes/api.php' for API routes. To create a controller, the following Artisan CLI command can be used:

```
php artisan make:controller ElementController
```

This command will create a new controller file in the 'app/Http/Controllers' directory, which can then be modified defining all the methods to handle actions on the model. RESTful API design is centered around a few core principles, including statelessness

and resource-based URIs. Statelessness ensures that each request from a client contains all the information needed to understand and process the request. Resource-based URIs provide a clear structure for accessing different resources, while standard HTTP methods (POST, GET, PUT, DELETE) correspond to CRUD (Create, Read, Update, Delete) operations.

3.4.2 Middleware

Middleware in Laravel provides a convenient mechanism for filtering HTTP requests entering the application. Middleware can be used for various purposes, such as logging, authentication and authorization. Laravel includes several built-in middleware, but custom middleware can be created using the Artisan CLI. The command

```
php artisan make:middleware MiddlewareName
```

creates a new middleware file in the 'app/Http/Middleware' directory, where the logic of the middleware is defined. For example, in our application, we usually check if the user is logged in before allowing them to access the majority of routes. Once the behavior of the middleware is defined, it can be assigned to routes or groups of routes in the 'routes/web.php' file, ensuring it will be executed for all routes within the group.

3.4.3 Internationalization

Internationalization is the practice of building applications that can support multiple languages and locales. Laravel provides a simple way to manage language strings and switch between languages based on user preferences or application settings. Language files in Laravel are stored in the 'resources/lang' directory, where each language supported by the application has its subdirectory. For example, to support English, Italian and Portuguese, we created the 'resources/lang/en', 'resources/lang/it' and 'resources/lang/pt' directories. Inside each directory, language files containing translated strings can be defined

```
// resources/lang/en/common.php
return [
    'WELCOME_MESSAGE' => 'Welcome',
];

// resources/lang/pt/common.php
return [
    'WELCOME_MESSAGE' => 'Bienvenido',
];
```

In our application, when a user switches their locale, their language preference is saved in two places: a session variable for the current session and a cookie for persistence across multiple sessions, ensuring their language preference is maintained for future visits.

3.5 Frontend Development

In this section, we delve into the frontend technology stack utilized in our web application, encompassing HTML, CSS, JavaScript, TailwindCSS, Chart.js and jQuery. We will explore how these technologies interact to create a dynamic, responsive and user-friendly interface.

3.5.1 HTML & Blade

HTML (Hypertext Markup Language) is the standard language for creating web pages. It provides the structure of a webpage by using a series of elements that define different types of content and their relationships. In our application, we integrate HTML with Blade, Laravel's templating engine, to dynamically generate web pages. Blade provides directives like `@extends`, `@section`, `@yield` and `@include` to manage layouts and content sections. These directives allowed us to create modular and reusable components, making it easier to maintain a consistent structure across multiple pages. Blade also facilitates data binding by allowing the easy insertion of dynamic content into views using curly braces. This feature enables the rendering of variables, functions and expressions directly within the template, providing a seamless way to display data passed from controllers to the views. Here is an example of how HTML and Blade work together:

```
{{-- resources/views/home.blade.php --}}

@extends('layouts.app')

@section('title', 'Home')

@section('content')
    <h1>{{translate('common.WELCOME')}}/h1>
    <p>This is the home page.</p>
@endsection
```

In this example, the `home.blade.php` view extends the `app.blade.php` layout. The `@section` directive helps define and display specific sections of the content dynamically, building on the initial layout.

3.5.2 CSS & TailwindCSS

CSS (Cascading Style Sheets) plays a crucial role in the presentation layer of web applications, providing a powerful mechanism to control the aesthetic and structural layout of web content. The use of CSS in our application transcends basic styling; it forms the

foundation for a responsive and user-friendly interface, ensuring it delivers a cohesive visual experience that aligns with our design principles. A responsive design ensures that our application functions similarly across various screen sizes enhancing user satisfaction by providing a consistent experience. Moreover, utilizing animations makes interactions more engaging and intuitive. For example, highlighting buttons or providing visual feedback on hover or click actions helps guide users through their tasks more efficiently. Displaying error or success messages in appropriate colors and positions is critical for user clarity. For instance, using red for error messages and green for success notifications can quickly convey the status of user actions. Organizing content through a grid layout maintains a clean and orderly interface. Grids help users orient themselves within the application by providing a predictable structure, making it easier to locate information and navigate between different sections. **TailwindCSS** is a utility-first CSS framework that streamlines the process of building custom designs by providing a comprehensive set of low-level utility classes. This approach eliminates the need to write CSS from scratch, significantly speeding up development and ensuring consistent styling throughout the application. It includes built-in responsive utilities that make it straightforward to adjust designs for various screen sizes. Developers can extend its default configuration or create custom utility classes to meet specific design requirements. Additionally, TailwindCSS includes a base stylesheet known as Preflight. Preflight normalizes styles across different browsers, providing a consistent foundation for the application. This helps maintain a uniform look and feel, regardless of the user's browser choice.

3.5.3 JavaScript & Chart.js

JavaScript is a versatile programming language used to create dynamic and interactive web applications. It allows us to manipulate the DOM (Document Object Model), handle events and perform asynchronous operations. In our application, we used jQuery, a fast, small and feature-rich JavaScript library that simplifies the previously mentioned tasks. Additionally, we leveraged the **Chart.js** library for charts and graphs. Chart.js is a flexible and straightforward JavaScript charting library that offers a variety of chart types, including bar charts and doughnut charts, allowing for diverse data visualization. Its main features include responsiveness, as charts created with Chart.js automatically adjust to different screen sizes and extensive customization options, enabling us to tailor the appearance and behavior of charts to meet our specific needs.

3.6 Layouts and Views

3.6.1 Sidebar

The design of the application incorporates a sidebar that remains consistently accessible across all views and sections of the application. This sidebar is a crucial navigational component, ensuring that users can seamlessly move between different parts of the application without needing to return to a main menu. The primary function of the sidebar is to provide users with quick and intuitive access to key milestone points within the application. These milestone points represent significant sections or features that are essential for the user's journey or workflow within the application. By placing these important links in the sidebar, the application minimizes the number of clicks and time required to reach frequently accessed areas. Moreover, the sidebar's design is optimized for usability and accessibility. As shown in figure 3.1 it is positioned in a way that does not interfere with the main content area, allowing users to navigate while maintaining their focus on their current tasks. Additionally, the sidebar works as a collapsible menu, allowing users to customize their view and access more detailed navigation options as needed. Figure 3.1 shows the sidebar on the left side.

3.6.2 Home

The home page serves as an overview of the project activities in the latest administrative year. It features statistics on the current number of villages managed, visits performed, active indicators and measures collected. These measures offer a snapshot of the REACH program's scale and activity. Additionally, the homepage features three informative graphs designed to provide deeper insights: two bar charts and one doughnut chart. The bar charts offer a detailed breakdown of the number of visits by district and by trimester, highlighting both the geographical and temporal distribution of visits. The doughnut chart provides a quick and clear overview of visit statuses, illustrating the proportion of completed visits versus those still pending, thereby helping users quickly assess their progress. Figure 3.1 shows an examplificative view of the homepage.

3.6.3 Provinces

The Provinces page provides the access point for managing Provinces, Districts, Areas and Villages. Users can navigate through these locations in a hierarchical path, for example, to reach Village D, they would follow the path Province A $\rightarrow$ District B $\rightarrow$ Area C where they can find Village D. Each level can be accessed step-by-step, ensuring a clear and intuitive navigation experience. At the same time, a

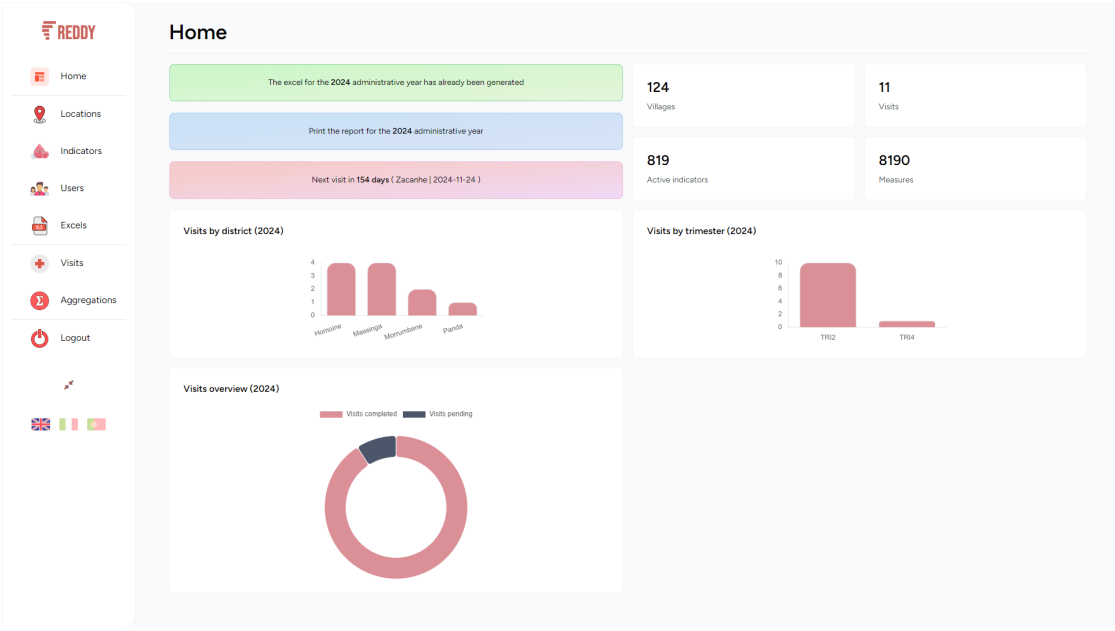

Figure 3.1: Website homepage and sidebar

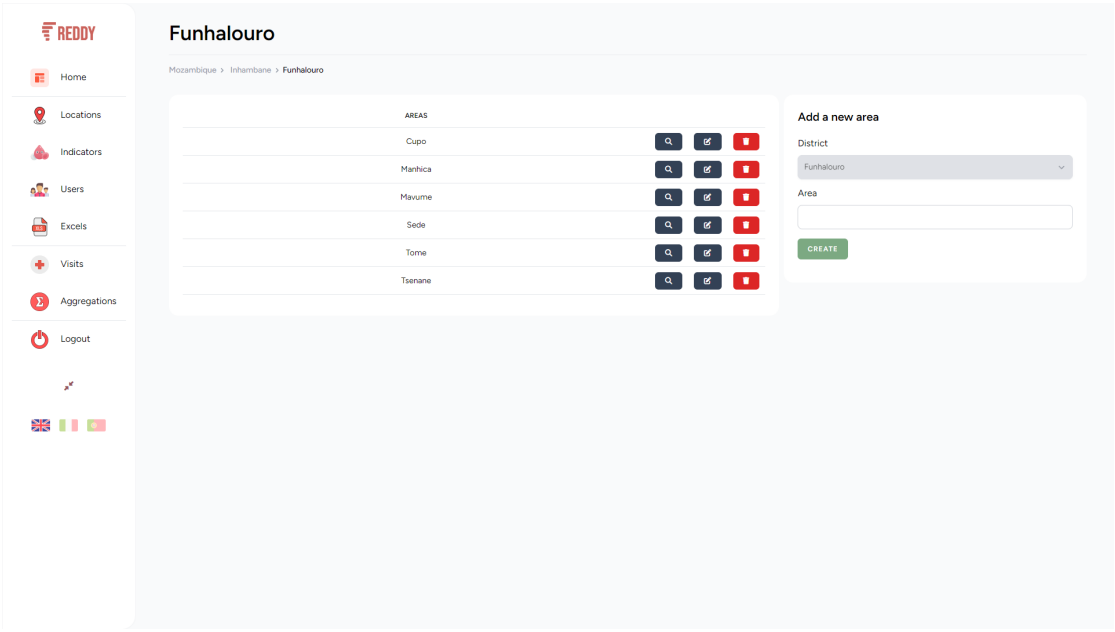

Figure 3.2: Areas management

breadcrumb component allows the user to easily go back to any of the previous hierarchical levels. Territorial entities can be deleted only if they do not contain any subordinate entities, maintaining the integrity of the hierarchy. Each page within the Locations section includes a form that allows administrative users to add new entities. Guest users and healthcare providers can only view existing entities and do not have permission to add new ones. At the same time, administrative users also can rename

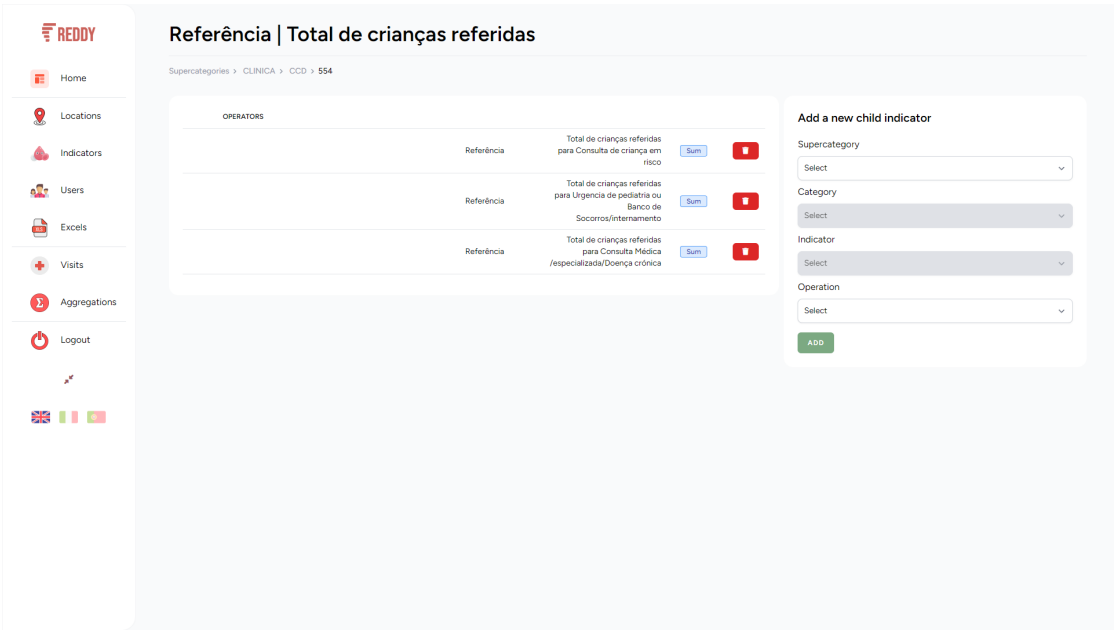

Figure 3.3: Details page for computed indicators

and move any territorial entity. As specified in our conceptual model, entities can also be relocated to different administrative territories without altering historical links. Figure 3.2 shows an example of the areas in the district of Funhalouro.

3.6.4 Supercategories

The Supercategories page shares a hierarchical structure similar to the Provinces page, organized from supercategories to categories, and finally to individual indicators within each category. Each page includes a form to create new entities and all information regarding supercategories, categories and indicators can be modified, including their relationships. On the Indicators page, there are additional features to enhance functionality. The form for adding new indicators allows the insertion of up to four parts per indicator. This is necessary because indicators can be complex and structured in various ways. For instance, there could be indicators with shared portion of their names: Disease A — Type B — Age — 0-5 months and Disease A — Type B — Age — 6-12 months, which identify different age groups. To improve the look and usability of the page, indicators with the same partial names are merged and grouped in the table. Figure **??** shows a typical indicator list. Additionally, each indicator has two boolean properties: one indicating whether it is active and the other indicating whether it is a computed indicator. Both properties can be toggled on the indicators management page. Inactive indicators remain in the indicator list but do not appear in the data aggregation charts and tables. The computed boolean flag shows whether the indicator requires a manually inserted value or if it is derived from other values. If an indicator is computed, clicking on the calculator icon redirects the user to a page where they can modify and insert the child indicators (operands) and specify how they are combined (formula).

3.6.5 Users

The Users page provides a comprehensive overview of all existing users, presenting key details such as their Name, Email and Role. This central hub for user management includes a form for adding new users. To create a new user, an admin needs to fill in the new user Name, Email and Password. Upon adding a new user, their default guest role can be adjusted if needed. This can be done through an intuitive editing popup that appears when the edit button next to a user entry is clicked. It is important to note that, by design, non-admin users are not permitted to delete or modify their credentials directly. This restriction is in place to maintain security and integrity. Any changes to a user's credentials, including deletion or modification, must be requested to and executed by an admin user.

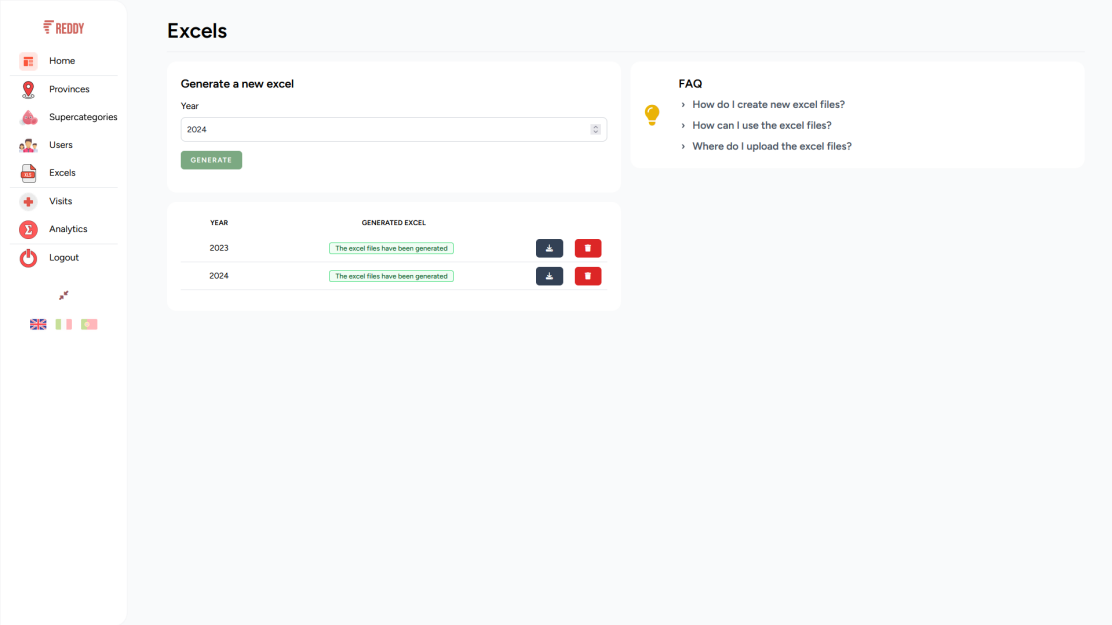

Figure 3.4: Excel management

3.6.6 Visits

This page lists all visits created, identified by the village and the date of performance. Each visit has one of two statuses:

- **Pending:** Scheduled for a past date but not yet documented by the healthcare provider
- **Complete:** All modules completed by the healthcare provider

Each module within these categories has a status showing whether it has been filled. If a module is not filled, it displays the number of missing indicators.

3.6.7 Excel Management

The Excel Management page provides the interface for handling Excel files that serve as templates for data collection. This page enables users to generate, download, and delete these templates while ensuring strict adherence to user roles and permissions. All users can access and download the necessary files, but only authorized personnel have the ability to modify the templates. To maintain data integrity, the system enforces specific rules regarding file generation and deletion. An Excel file cannot be generated for a particular administrative year if a file for that year already exists in the system. To generate a new file for the same year, the existing file must first be deleted. However, deletion is only allowed if there are no visits already associated with the existing Excel version. This precaution ensures consistency and prevents data loss. Once the Excel files have been downloaded and filled with relevant information, they can be reuploaded either as folders or as single files. Upon reupload, the files are parsed and the measures are inserted into the database. If characters are detected where numbers are expected, the measure is not saved and the visit page will signal the missing data.

3.6.8 Aggregation Page

Upon landing on the page, users are greeted with an intuitive interface that highlights the main features and functionalities available for report generation and data analysis. They can then select the desired parameters for their report, including the geographical aggregation (province, district, or area) and administrative year, using easy-to-use dropdown menus. Once the parameters are set, users can initiate the report generation process, with the system processing the request and automatically downloading the report in PDF format.

4 Usability Tests

After the development phase, we initiated the user testing phase. This chapter outlines the methodology employed for usability testing and discusses the results obtained. The tests performed included the System Usability Scale (SUS), the Computer System Usability Questionnaire (CSUQ) and the User Experience Questionnaire (UEQ).

4.1 System Usability Scale (SUS) Test

The **System Usability Scale** is a robust, reliable tool used to measure the usability of various systems, such as software applications, websites and hardware devices. Developed by John Brooke in 1986, SUS has become a standard in usability testing due to its simplicity, versatility and effectiveness. It provides a quantitative measure of usability, which will be crucial for comparing different iterations of the Freddy web app. The SUS comprises a 10-item questionnaire. Each item is a statement regarding the system under evaluation and respondents need to indicate their level of agreement using a five-point Likert scale ranging from *Strongly disagree* to *Strongly agree*. The statements address different aspects of usability, including ease of use, system complexity and the user's overall satisfaction. A SUS score above 68 is generally considered above average, while scores below 68 suggest there may be usability issues that need addressing.

4.2 Computer System Usability Questionnaire (CSUQ)

The **Computer System Usability Questionnaire** is a widely utilized tool for assessing user satisfaction with the usability of computer systems. The CSUQ is designed to evaluate various dimensions of a system's usability by gathering feedback directly from users after their interaction with the system. The questionnaire comprises 19 items that are categorized into three distinct scales: System Usefulness (SYSUSE), Information Quality (INFOQUAL) and Interface Quality (INTERQUAL). These categories collectively contribute to an Overall Satisfaction score. Each item is rated on a 5-point Likert scale ranging from *Strongly disagree* to *Strongly agree*, providing an understanding of the user's experience. In addition to its comprehensive nature, the CSUQ is straightforward to administer and analyze. The 5-point Likert scale allows for detailed responses while maintaining simplicity in scoring and interpretation.

4.3 User Experience Questionnaire (UEQ)

The **User Experience Questionnaire** is a comprehensive tool designed to assess the user experience of interactive products. Unlike traditional usability metrics that primarily focus on effectiveness and efficiency, the UEQ captures a broader spectrum of user experience aspects, containing both pragmatic and hedonic qualities. The UEQ is structured to provide insights into six key scales: Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation and Novelty. Participants interact with the product and then complete the User Experience Questionnaire (UEQ), which consists of 26 items. Each item presents a pair of contrasting characteristics (e.g., "attractive/unattractive") and participants rate their experience on a 7-point scale. The scale ranges from 1 to 7 but the goal is for participants to select a score that is closer to the characteristic they feel best describes the product. The UEQ is particularly valuable for its ability to provide a complete view of the user experience. By addressing both pragmatic qualities (such as efficiency and dependability) and hedonic qualities (such as stimulation and novelty), it offers a comprehensive assessment that can guide product development and improvement.

4.4 Testing Environment

The usability tests were conducted in an online environment. Each participant performed the test on their personal computer, which they regularly use for their daily operations related to the REACH project. The online approach was chosen for its efficiency, especially given the geographic distribution of testers in Italy and Mozambique. By conducting the tests online, users remained in a familiar environment, reducing the impact of unfamiliar variables on their performance and feedback. Participants connected to the test sessions via Microsoft Teams, with each session allocated a 1.5-hour time slot. During the sessions, they were required to share their screens with the observers. This screen-sharing setup allowed observers to view the participant's interactions with the system in real-time, providing valuable insights into their usability experience. Observers play a crucial role in the testing process. At the beginning of each session, their primary responsibility is to explain the purpose and process of the test to the participants. During the actual test, observers remain passive to avoid influencing participants' actions and responses, intervening only if necessary. Observers also gather additional data by listening to participants as they think aloud and explain their reasoning while completing tasks.

4.5 Produced Documents

For each participant, three documents were produced:

- **Informed Consent Form:** Each participant signed an informed consent form before the test commenced. This document outlined the purpose of the test, the procedures involved and the participants' rights, ensuring that they were fully informed and had agreed to participate voluntarily.
- **Initial Questionnaire:** This questionnaire was administered at the beginning of the test session and collected demographic and background information about the participants. Data such as age, gender, professional role and prior experience with similar systems were gathered to contextualize the usability test results and analyze the participant profile.
- **Final Questionnaire:** After completing the usability test, participants were asked to fill out a final questionnaire. This document contained the questions related to the SUS, CSUQ and UEQ tests, along with additional questions tailored to the specific features and functionality of the system.

These documents collectively provided a comprehensive overview of each participant's background, the usability issues encountered and their overall experience with the system. The informed consent form ensured ethical compliance, the initial questionnaire helped in understanding the participant demographics and the final questionnaire provided critical data for assessing the user experience with the system.

4.6 Participant Profile

This section details the profiles of users selected for the usability tests of the REACH project. The participants were chosen by the Medicus Mundi team, who were responsible for coordinating their availability. We requested a diverse and representative sample to ensure that the range of users interacting with the system in various roles was accurately reflected. A total of five participants were made available for the tests.

Age Range and Sex The majority of the selected users fall within the 25-34 age range, closely followed by the 35-44 age bracket. This reflects a relatively young group, consistent with the current demographic trends of individuals working on the REACH project. The gender distribution shows a balanced representation, with both male and female users participating equally. This balance is crucial as it ensures that feedback is diverse and representative.

Education Level and Profession A significant portion of the selected users hold a Bachelor's degree or higher, with many possessing Master's degrees. This high level of education reflects the specialized nature of the tasks within the REACH project, which requires advanced skills and knowledge. Professions vary, including nurses, industrial engineers and preventive medicine technicians. This diversity in professional backgrounds underscores the multidisciplinary approach of the REACH project, which necessitates collaboration across various fields to achieve its objectives.

4.6.1 Technical Proficiency and Usage

Computer Usage All selected users report using their computers for both work and personal use, indicating a high level of familiarity with basic digital tools. The hours of computer usage per day vary, with some using it for 1-4 hours and others for 5-8 hours. A notable portion mentions that they do not use it every day, which could suggest varying roles and responsibilities within the project.

Browser Preferences Google Chrome is the most frequently used browser among participants, alongside Safari and Mozilla Firefox. This preference aligns with global trends, as Chrome is well-known for its speed, reliability and extensive range of productivity-enhancing extensions. Although our application is designed to function uniformly across all browsers, this information is valuable to note.

Website Usage and Web Application Expertise Participants commonly visit websites related to work or professional activities, online shopping, social media and education. Their expertise with web applications varies from beginner to expert levels. Some participants have experience with similar applications to the one being tested, particularly *Epi Info and Access*, which are used for data collection and analysis. This variance in expertise is critical for understanding the training and support needs within the REACH project. Beginners might require more comprehensive onboarding and ongoing support, while experts could contribute to refining the system and mentoring less experienced users.

4.6.2 Data Management Experience

Roles in Data Management The data management roles encompass data collection in the field, data entry into Excel files and data control and analysis.

This section of the questionnaire connects to the subsequent one on satisfaction with the current system. Professionals involved in data entry are generally satisfied with the Excel system, while those responsible for data control and analysis typically express dissatisfaction with it.

Satisfaction with Current Systems Feedback on the Excel-based data management system is mixed, as anticipated. While some users are satisfied, many express dissatisfaction. The primary reasons for this dissatisfaction include Excel's limitations in handling human error and its lack of advanced features compared to specialized data management software.

Additional Opinions on the Current Excel System Before proceeding with the usability tasks, professionals were asked to share their views on the positive and negative aspects of the current system. The most commonly cited positive aspects were the users' familiarity with Excel and its ease of use. One participant noted that, despite not being fully satisfied with the current system, using Excel files was a significant improvement over paper forms. On the negative side, participants highlighted issues with data validation, problems with formulas not working correctly and accidental alterations to the sheet format.

4.7 Test Results

4.7.1 System Usability Scale (SUS)

The SUS score achieved was 76.00, indicating above-average usability. This score is above the commonly accepted threshold of 68.00 for average usability, indicating that while there is room for improvement, the system is generally well-received by its users. Based on the mean responses provided for individual SUS items, users generally agree that the system is easy to use and they feel confident using it. They also agree that the system is well-integrated and quick to learn. Users strongly believe that the system is not unnecessarily complex. Additionally, they find the system relatively easy to navigate and generally find its functions consistent. While users are neutral or slightly positive about needing minimal support to use the system, there are some reservations or difficulties. This suggests that there is still room for improvement in terms of usability, but overall, the feedback points towards a positive user experience.

4.7.2 Computer System Usability Questionnaire (CSUQ)

The CSUQ results indicate a strong positive reception of the system across various dimensions. The overall satisfaction rating of 4.36 suggests that users are quite pleased with their experience. This score is in the upper range considering the 5-point scale. This indicates that the system not only meets but potentially exceeds user expectations. The system usefulness rating of 4.22, while slightly lower than the overall satisfaction score, still reflects a high level of perceived utility, suggesting that users find the system to be beneficial in achieving their tasks. This indicates that the system is well-designed to support user workflows and provides valuable functionalities that users need. Information quality received the highest score at 4.46, which is exceptionally positive. Information quality is a critical factor for users, as it encompasses the accuracy, relevance and comprehensiveness of the data provided by the system. A score of 4.46 suggests that users find the information to be reliable and useful. Finally, the interface quality rating of 4.36 indicates that users find the system interface to be both user-friendly and aesthetically pleasing. Users appreciate the intuitive navigation and visual design, which enhances their overall experience and efficiency when interacting with the system. In summary, the CSUQ results reveal a successful system with notable strengths in information quality and interface design. While system usefulness is slightly lower than the other metrics, it is still above average and suggests minor areas for enhancement. Overall, these results reflect a well-received system that performs well across all evaluated dimensions.

4.7.3 User Experience Questionnaire (UEQ)

The User Experience Questionnaire (UEQ) results for this study provide insights into six different scales, each assessed on a range from -3 (extremely bad) to +3 (extremely good). Scores between -0.8 and 0.8 are considered neutral, while scores above 0.8 indicate positive evaluations and those below -0.8 indicate negative evaluations. The attractiveness scale received a highly positive mean score of 2.133, indicating that participants found the system appealing. The relatively low variance of 0.38 suggests a consistent evaluation among the participants regarding the system's attractiveness. In terms of perspicuity, the scale achieved a mean score of 1.750, reflecting a generally positive perception of clarity and ease of understanding. However, the higher variance of 1.34 points to a broader range of responses, indicating some variability in how participants perceived this aspect of the system. Notably, the UEQ analysis tool highlighted a potential error in this scale, pointing to one evaluation concerning the system's comprehensibility. This particular evaluation was inconsistent with other evaluations on similar characteristics made by the same tester. Upon further verification, it was

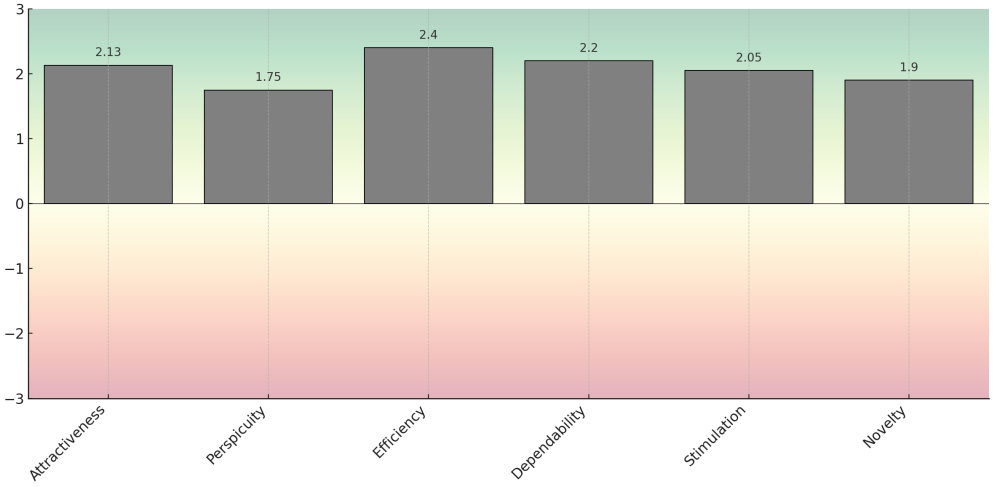

Figure 4.1: UEQ summary

confirmed that the user had indeed made an error. Despite this, we decided to retain the initial score. This decision reflects the realistic constraints of testing, where it is often not feasible to contact testers and acknowledges the likelihood of a small number of errors. Efficiency was highly rated with a mean score of 2.400, suggesting that users found the system to be very efficient. The low variance of 0.3 indicates strong agreement among participants on the system's efficiency. Dependability received a mean score of 2.200, showing that participants considered the system to be highly reliable. The variance here was also low at 0.29, indicating consistent evaluations among participants. The stimulation scale had a mean score of 2.050, signifying that users found the system engaging and exciting. The variance of 0.67 suggests a moderate level of agreement among participants. Novelty was rated positively with a mean score of 1.900, indicating a favorable perception of the system's innovativeness. However, the higher variance of 0.93 indicates some variation in how users perceived the novelty of the system. In summary, all six scales scored positively, with means well above the neutral range, suggesting that the system was well-received. The variances indicate varying levels of agreement among participants, with efficiency and dependability showing the most consistent evaluations. Notably, the primary area of concern appears to be the learning curve and overall difficulty of the system. Some testers indicated that the system was not extremely easy to learn. This feedback is understandable, given their familiarity with the existing Excel-only system, but highlights an area that needs to be addressed.

4.7.4 Benchmark

The measured scale means are compared to values from an extensive benchmark dataset, which includes data from 21,175 individuals across 468 studies, covering various products such as business software, web pages, web shops and social networks. This comparison allowed us to conclude the relative quality of the Freddy web app. Overall, the web app performs well across most scales, frequently ranking within the top 10% of results. This is evident in the scales of Attractiveness, Efficiency, Dependability, Stimulation and Novelty, all of which received an "Excellent" rating. These results indicate that the product is highly regarded in terms of user experience. The Perspicuity scale, while still performing well with a "Good" rating, suggests that there is potential to further enhance its clarity and intuitiveness.

4.7.5 Additional Insights Gathered from Observing the Tests

During the testing phase, several key insights were gathered that can enhance the system's usability and user experience. These insights are based on observed interactions and user feedback, particularly focusing on internationalization, data handling, user feedback, UI placement and error messaging.

- **Complete Internationalization**: Complete internationalization is essential for improving the usability of the system across different languages and regions. During testing, it was observed that Portuguese operators found the system difficult to use, as only automatic translations were available, and required the help of an

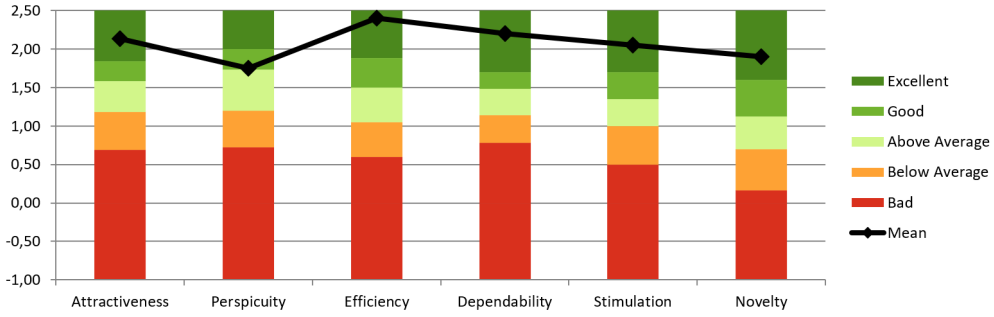

Figure 4.2: UEQ Benchmarking

interpreter. This underscores the need for a fully internationalized and localized interface to suit non-English/Italian speakers effectively.

– **Preserve Input Data on Error**: Preserving input data on error is important for enhancing user experience. For instance, when an admin enters an incorrect password while creating a new user, the system should retain the previously entered username and email address. This functionality should be generalized across other forms to minimize repetitive data entry and improve user convenience.

– **User Feedback for Flag Changes**: User feedback for all changes is another important aspect. Some operations, such as checking checkboxes on the indicator management page, do not provide sufficient feedback to the user. It is important to ensure that users receive clear and immediate confirmation when a flag is changed, thereby improving transparency and user satisfaction.

– **Clarify Error Messages for Linked Entities**: It is essential to make error messages clearer for non-technical users. For instance, when an entity cannot be deleted because it has linked child entities, the error message should be easy to understand. Simplifying these messages will help users grasp the issue and take the right action without needing technical expertise.

– **Handle Empty Supercategories**: The creation of an empty supercategory has caused the website to break. Although the cause for this problem has already been identified and is easy to fix, it highlights the need for careful testing to prevent similar issues in production.

5 Conclusions

The REACH project has successfully established a robust foundation for enhancing healthcare access and quality in the Inhambane province of Mozambique. By developing a user-friendly web application for managing clinical data, the project has now the basis to improve its data management and analysis processes. This transformation enables more informed decision-making and widespread access to synchronized data. The developed system allows healthcare providers to continue using familiar Excel tools while seamlessly integrating with the new web system. The Excel import feature, requested by the Medicus Mundi team, ensures a smooth transition and maximizes the system's usability. Additionally, its reporting tools provide analysis and reporting capabilities, empowering healthcare professionals to derive meaningful insights from the data. The choice of Laravel as the development framework for the web application has been instrumental in achieving the project's goals. The framework's built-in tools for database management, such as migrations and seeding, streamlined the integration of current clinical data into the system, enhancing data integrity and consistency. Furthermore, Laravel's support for modern PHP practices facilitated the development of an intuitive user interface for the involved personnel, which was paramount considering the varying levels of technological proficiency of the professionals involved in the project. Deployment was notably straightforward, thanks to Laravel's organized project structure, which facilitated a smooth transition from development to production. Tests showed that the system has been appreciated by everyone involved. The usability testing phase revealed that users found the system efficient, reliable and user-friendly. The System Usability Scale (SUS) score indicated above-average usability, highlighting strengths in ease of use and integration. The Computer System Usability Questionnaire (CSUQ)

results showed high overall satisfaction, with particularly high scores for information quality and interface design. Similarly, the User Experience Questionnaire (UEQ) results reflected a positive user experience across various dimensions, with high ratings for attractiveness, efficiency and dependability. While there were some areas identified for improvement, such as learnability and the addition of innovative features, the overall feedback was positive, demonstrating the system's effectiveness and user acceptance. Beyond addressing the errors identified during testing, additional enhancements can be considered to further optimize the system's performance and expand its capabilities. Integrating more advanced analytics and reporting tools would enable deeper insights from the collected data. By developing better charts and visualizations, trends can be more easily identified, facilitating a clearer understanding of the data. Developing an API that provides external access to the collected data would facilitate research by third-party projects, fostering collaboration and potentially leading to further healthcare enhancements. Translations are currently available in Italian and English, but ensuring Portuguese localization remains a priority to enhance accessibility and usability for local healthcare providers. Finally, due to the current state of internet connectivity in Mozambique, we were compelled to adopt a hybrid approach, integrating Excel with the web application. However, as internet stability improves and users become more familiar with the application, there is potential to transition data entry directly onto the platform, gradually reducing reliance on Excel files and increasing overall operational efficiency.