MEETING TARGETS AND MAINTAINING EPIDEMIC CONTROL (EPIC) PROJECT

COOPERATIVE AGREEMENT NO. 7200AA19CA00002

Decentralized Drug Distribution Mobile Application: Overview and Technical Architecture Guide

SEPTEMBER 2021

PEPFAR

USAID FROM THE AMERICAN PEOPLE

EpiC Meeting Targets and Maintaining Epidemic Control

ACKNOWLEDGMENTS

This guide was drafted by Moses Bateganya, James Batuka, Oyibo Igagwu Idris, and Lirica Nishimoto. It was edited by Stevie Daniels with page layout by Jill Vitick and cover template design by FHI 360 Design Lab. This document was made possible by the generous support of the American people through the United States Agency for International Development (USAID) and the U.S. President's Emergency Plan for AIDS Relief (PEPFAR). The contents are the responsibility of the LINKAGES and EpiC projects and do not necessarily reflect the views of USAID, PEPFAR, or the United States Government.

Suggested Citation: EpiC. Decentralized Drug Distribution Mobile Application: Overview and Technical Architecture Guide. FHI 360; Durham (NC): 2021.

EpiC is a global cooperative agreement dedicated to achieving and maintaining HIV epidemic control. It is led by FHI 360 with core partners Right to Care, Palladium International, Population Services International (PSI), and Gobee Group. For more information about EpiC, including the areas in which we offer technical assistance, click here.

CONTENTS

PURPOSE

This guide facilitates the adaptation of EpiC's generic Decentralized Drug Distribution Mobile Application (DDD App) to the local context. Programs use the app to support implementation of decentralized drug distribution (DDD) and improve the exchange of data between public health facilities and community antiretroviral (ARV) medication pickup points. The guide provides an overview of use, key functions, and adaptability.

Part 1, intended for program staff, outlines the purpose and functions of the app, provides a detailed explanation of how it can be used by service delivery point staff, and illustrates how it has been adapted for various contexts (see case studies) and models.

Part 2, intended for IT system developers, health information system (HIS) staff, and programmers, details the app architecture to be used when making adaptations for specific contexts and needs.

INTRODUCTION

The fight against HIV globally requires collective effort from the government, private sector, and donor community. With a growing number of countries on the cusp of epidemic control, any effort toward improvement in client access to antiretroviral therapy (ART) refills to allow for flexible and cost-effective delivery is a step in the right direction, especially in the context of the COVID-19 pandemic.

Many countries have adopted several differentiated service delivery (DSD) models with clients established on treatment increasingly being enrolled in multi-month dispensing (MMD) with 3- or 6-month refills. The COVID-19 pandemic has necessitated the adoption of additional out-of-facility individual DSD models such as through private pharmacies and unmanned and contactless smart lockers which decongest health facilities and reduce the risk of COVID-19 infection for both clients and health care workers (HCWs). Twelve countries in sub-Saharan Africa have adopted the private pharmacy model, eight with EpiC support and four with other bilateral U.S. President's Emergency Plan for AIDS Relief (PEPFAR) partners' support. In this model, public health facilities (hubs) partner with more than one private pharmacy to serve as ARV refill points, or DDD outlets, for clients established on ART.

Sharing of client data between health facilities and ARV refill points, including private pharmacies, private clinics, and other community points, is important for seamless implementation of these DDD models and to ensure refill completion and track missed appointments in real time. However, the lack of electronic medical records (EMRs) for safe, seamless, and real-time sharing of client data between public health facilities (hubs) and refill points (outlets) limits the wide scale-up of these models.

PART 1: OVERVIEW OF THE DDD APP

The DDD App is an online and off-line platform using a smartphone, tablet, or laptop to enable real-time data exchange between the public health facility (with or without an EMR) and community refill points. It is designed to be used at both the hub health facility and the outlet refill points enabling interactive bidirectional communication between both service delivery points. In the long run, the DDD App can be adapted to include a client-facing portal, which would facilitate tripartite communication and client-centered care. DDD outlets can use the app to transmit data for timely updates to health facility records. Hub health facilities are notified in real time regarding refill completion and missed appointments of clients to pick up their refills.

DDD App: Key Features

- Seamless, synchronized, interactive communication between hub facilities and DDD outlets (refill points)
- Flexible and efficient data capturing to document services provided by DDD outlets
- Simple and easy-to-use interface
- Real-time data sharing between hub facilities and DDD outlets
- Deployable using smartphones, tablets, and computers
- Capacity for automated reminders

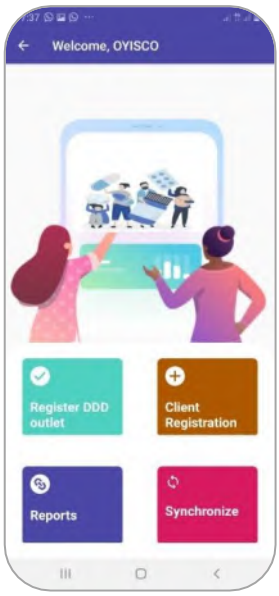
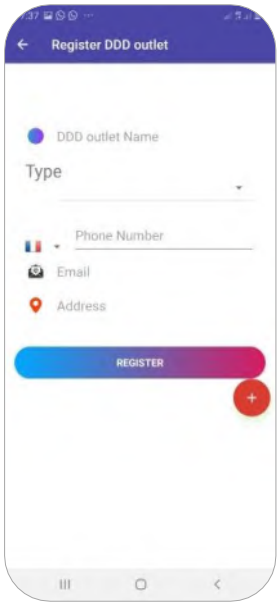
Facilities without an EMR system can use the DDD App to create a client profile and enter and record details. If an EMR system is in place, the DDD App can be linked to it and automatically pull information to develop client profiles within the app. The DDD App is not another EMR system but a platform that helps to create a seamless, secure, and interactive exchange of client information between hub facilities that use EMR systems and the outlets. The DDD App cannot replace existing partner or national EMR systems. Client profiles in the DDD App include:

- Client unique ID
- Client clinic number
- Date of birth
- Phone number for notification
- Regimen
- Height (once for adults) updated at every visit for children
- Weight (updated at each visit)
- Date of next refill and expected return date to hub facility
- Date of next viral load test
- Assigned DDD outlet

  (To switch outlets, client will go back to the health facility to be assigned to a new outlet.)

  (The DDD outlet can terminate or refer a client back to the hub facility.)

Figure 1. Visuals of the DDD App interface

Home Screen	DDD Outlet	Service Selection
		

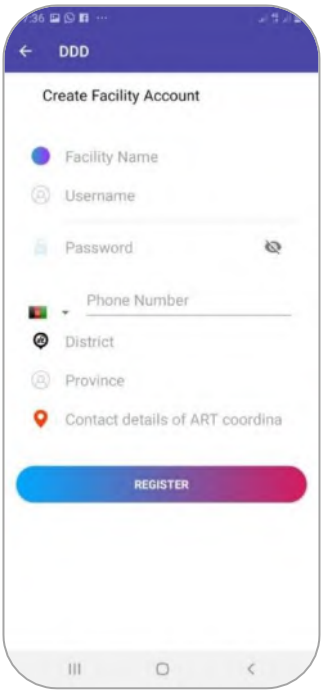
DDD App setup

Upon downloading the DDD App on a supported device, each hub facility will establish a password-protected account with multiple logins for staff responsible for DDD. The hub facility staff will then register the DDD outlet within the DDD App by entering the DDD outlet details including:

- DDD outlet name
- Outlet type: Pharmacy, clinic, home delivery, etc.
- Phone number for SMS notification
- Email address for email notification
- Physical address
- General Packet Radio Service (GPRS) codes will be selected automatically

Once the DDD outlet is registered, the hub facility staff can send an email or SMS with the activation link to the outlet through the app. Before clicking on the activation link, the outlet staff members should have downloaded the DDD App on their device. When DDD outlet staff click on the activation link, the app will automatically launch and prompt the DDD outlet staff members to

Figure 2. Facility account registration page



set up a password-protected account. They can then log in to their account from multiple devices. Once the account is set up, the hub facility will be able to devolve clients to the outlets, allowing the outlet to access and update the client profile.

Online trainings on how to set up and use the app can be conducted for the various cadres of providers. DDD App training is included in the DDD training materials. A YouTube video of the introduction to the DDD App is also available for self-training and orientation.

Uses and navigation

Once a client enrolls in DDD and chooses their preferred outlet, the hub facility can assign their profile to that specified outlet within the DDD App. At this point, the specified outlet will have access to the assigned client profile. The outlets only have access to client profiles that are assigned to them. The DDD App supports the following services:

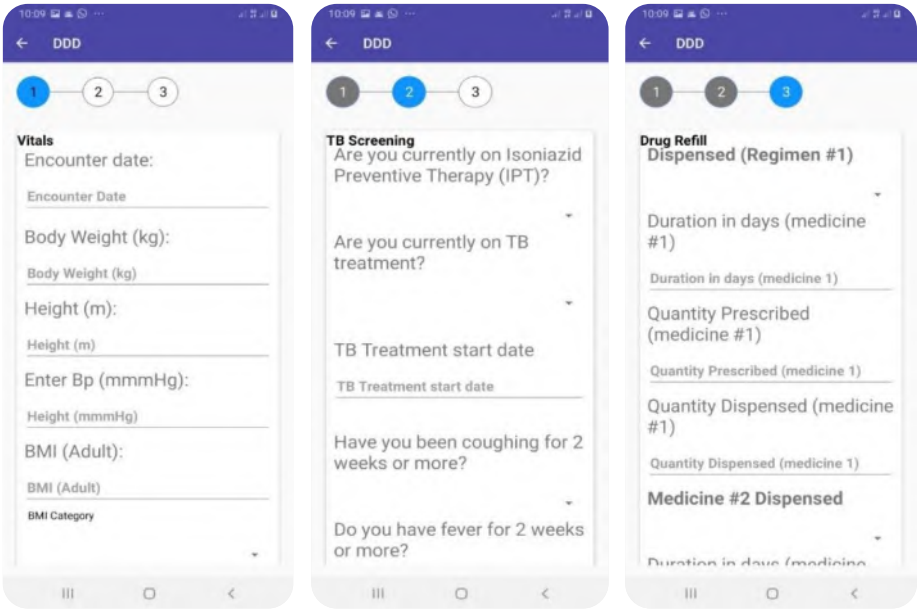
ARV dispensation

- Upon dispensing ARVs, the DDD outlet will enter the relevant information into the client profile in the app. This entry will provide confirmation to hub facilities that the refill has been completed or is incomplete, allowing real-time tracking of both clients and commodities.

Supporting provision of other services at DDD outlets

- Documentation of clients' vitals (weight, blood pressure, etc.)
- TB screening tool
- Adverse drug reaction screening tool
- COVID-19 symptoms screening tool
- Documentation of drug refills

Figure 3. Images of DDD App functions to support service provision



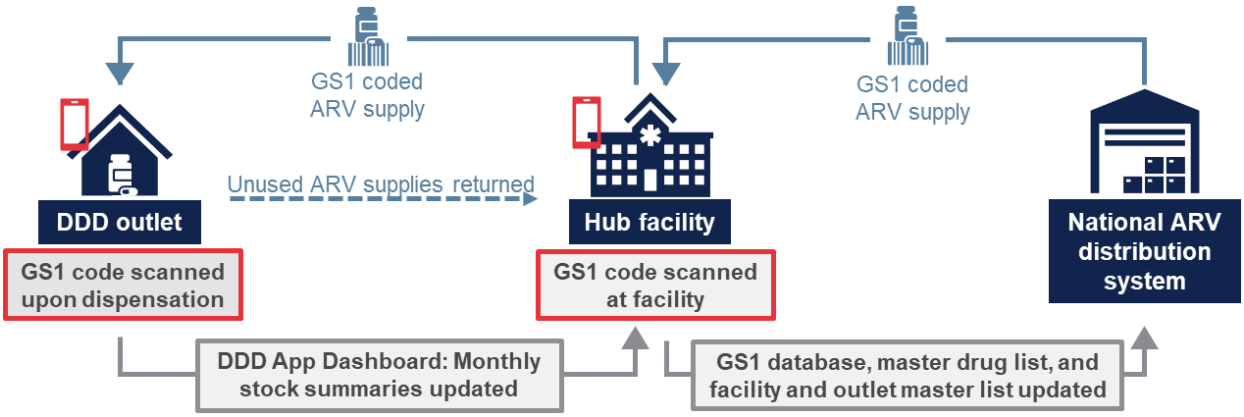
Communication with clients

- The DDD App automatically sends SMS reminders to clients before scheduled appointments. If appointments are missed, automatic SMS messages are sent to reschedule.
- DDD outlet staff can view upcoming refill appointments and contact clients through the mobile number saved in the DDD App to remind them about refill or viral load appointments.

Tracking commodities

The DDD App can also be used to track commodities. Once a client's three-month supply of ARVs is provided to a specified outlet, hub facilities can record the batch number, the unique code (e.g., GS1 barcode) of each ARV bottle, and the number of packs allocated to the specific outlet. The quantity of ARV bottles to be supplied to an outlet can be calculated based on the number of clients scheduled for pickup and their number of months of dispensation, which is also automatically calculated by the DDD App. Upon dispensing the ARVs to clients, outlet staff will record the necessary information from that visit in the app, including the quantity and the specific number of ARV packs dispensed to each client by scanning the bar code or documenting the unique code of each ARV bottle. The DDD App will then automatically subtract the number dispensed from the total number allocated to the outlet to keep a real-time record of the number of bottles of ARVs remaining at the outlet. If all clients scheduled for pickup during a month successfully pick up the correct number of months of ARVs, by the end of the month, the outlet's supply should be exhausted.

Figure 4. How the DDD App can be used to track ARVs



Hub facilities data updates

Updates made on the outlet staff DDD App account are automatically visible on the hub facility staff DDD App as soon as internet connection is available, and data is synchronized. The DDD App can also link with EMR systems allowing updates to occur. The DDD App works both online and off-line. When working off-line, users should regularly log in and connect to the internet to synchronize data. At health facilities without EMR systems, updates are available to the hub facility DDD App account, and paper records can be updated using real-time data from the app.

Program management and data analysis

- **Country-specific access:** Program managers and administrators have access to their country´s data and will not have access to view data from projects in other countries. The administrator login and additional access will be restricted to implementing partner staff for purposes of viewing reports.
- **Data visualization:** Field staff and backstops can view live dashboards such as number of refills over a specific period.
- **Data capture:** For sites with EMR systems, data for assigned clients will automatically be sent from the hub site to the assigned DDD outlet.

Reports

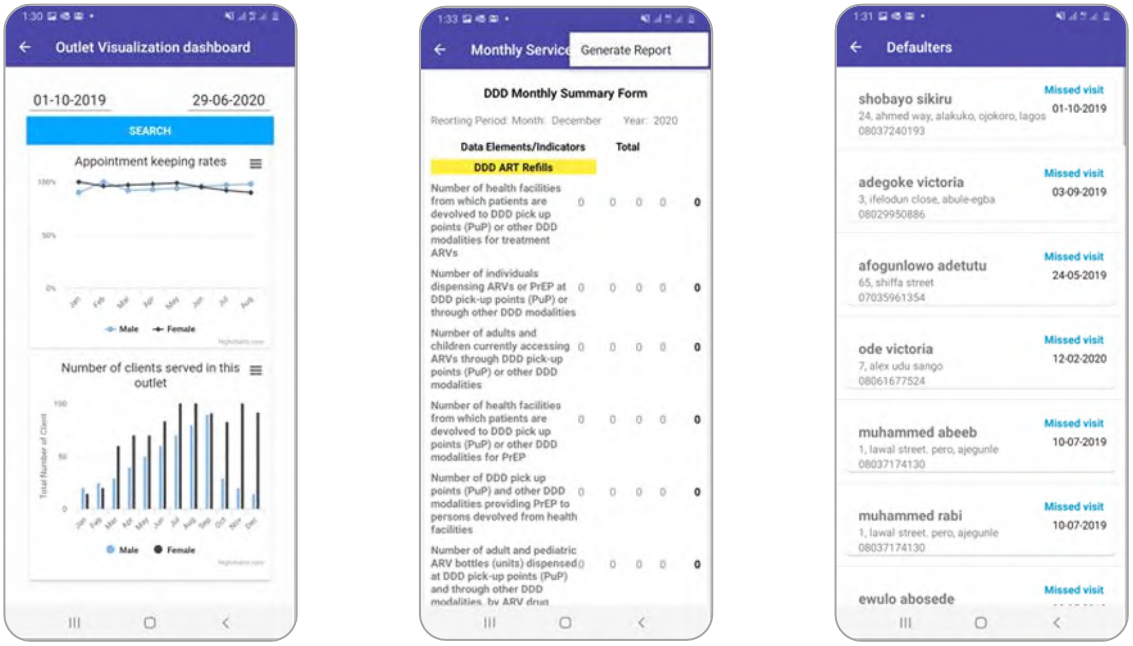
The DDD App autogenerates the following reports.

- Monthly summary by hub facility and DDD outlet
- List of registered DDD outlets
- Number of clients enrolled in DDD by hub facility
- Number of clients served, newly enrolled by DDD outlet
- Number of clients who missed appointments

Additional functions are available in the DDD App, and different functions are made available to hub facilities and outlets based on the needs in each context. Annex 1 outlines the functions of the DDD App and the service delivery point that would have access to each function. Adaptations can be made. For example, if needed, all functions above can be made available to all users, hub facility and outlet.

Figure 5. Data visualizations available on the DDD App

Outlet Records	DDD Monthly Summary	Appointments Missed



Data security and client confidentiality

Data recorded in the DDD App are stored on a server in each country, and the information generated is under the custody of respective countries either on a cloud-based or local server. The country manages their data, and the necessary security requirements should be installed on the server. The features below ensure data security and client confidentiality, and are reinforced by privacy procedures set up by different countries:

- Secure web hosting by recognized high-capacity cloud hosting vendors that ensure infrastructure is maintained with the newest versions of security patches.
- Secure sockets layer (SSL) encryption secures communication between the server and DDD providers and clients to prevent bots, hackers, or malware from intercepting data in transmission.
- App maintenance and software elements updated to newest versions and patches to help secure any emerging vulnerabilities.
- Limited identifying information recoded on the app, including client ID and basic demographics. However, clients' mobile numbers are required to provide follow-up services and notification (consent to use the mobile numbers is obtained at enrollment).

- Secure login portal securing client data with hidden phone numbers that are only accessed temporarily by case managers, and a data export sheet that replaces phone numbers with a unique identifier code thereby removing all personally identifying information in data exports.
- Informed consent obtained from clients before they are devolved explaining in clear terms what data are collected, why they are collected, and how the limited data may be shared securely with clinic staff and program staff.
- Staff user guides that outline user roles and staff trained in the crucial elements of protecting privacy and handling exported data responsibly.

Maintenance and sustainability

Local implementing partners and governments can roll out and scale up the DDD App to support other community DSD models. Since investment and maintenance costs are low, they should be able to support the app without additional funds. The DDD App:

- Has low maintenance costs, mainly for regular updates and patches
- Is open source with no requirement for annual license fees
- Is hosted by the Google Play and Apple App stores, which allows for centralized updates and easy maintenance managed by EpiC
- Is installed on smartphones, tablets, or laptop computers owned by the DDD outlet, which makes installation affordable and quick
- Does not require an additional server in countries with EMR systems
- Requires a cloud-based server or a local physical server in countries without EMR systems; a cloud-based server will require the country to pay for hosting of the data

Country adaptation

The DDD App has been developed in a generic form so that it can be easily adapted for each country. The relevant branding, including the national emblem, will appear on the app. Specific national, provincial, district, or local maps are incorporated to be used in relevant reports and dashboards. The DDD App can also be customized to offer multiple options for display languages, including local ones.

Aggregate data can be made available to country leadership through regular reports and dashboards. DDD summary reports can be added to national HIV websites or dashboards. The information is summarized at district, regional, provincial, national, and partner levels. Partners can view the reports and dashboards on a read-only basis. All information generated is under the custody of the country leadership. The DDD App has been and is planned to be adapted to many country contexts. In Cote d'Ivoire, it has been adapted and configured for use in French to support their pharmacy model; in Zimbabwe, plans are underway to adapt the app to support their pharmacy model commodity tracking using the GS1 code; and in Cameroon, the DDD app is included in the pharmacy model pilot plan approved by the Ministry of Health (MOH). The following are other examples and testimonials from Liberia and Nigeria on adapting and using the app.

Liberia experience

Liberia initiated a DDD pilot with one health facility in Monrovia,12 private pharmacies, and two CBO offices as DDD outlets. The DDD App was adapted for the Liberian context in consultation with health informatics representatives from the MOH by including the Liberia National AIDS Control Program logo and modifying data entry fields to match national reporting tools. The app is hosted on a local server. Health facility, pharmacy, and other relevant staff were introduced to and trained on the app to support real-time exchange of data, reporting, and monitoring of ARV dispensation between the health facilities and the DDD outlets. Internet-ready and toll-free tablets pre-installed with the DDD app were provided to the health facility and DDD outlets. The pilot was rolled out in May 2021 and, as of August, 55 patients had been devolved to pick up ARVs at pharmacies using the app.

Nigeria experience

In Nigeria, the DDD App has been deployed for various DDD models, including the community pharmacy ART refill program (CPARP), decentralized ART refill facilities (DARF), and community adherence refill groups (CARGs). Cumulatively, 7,683 clients on ART in Cross River, 34,101 clients in Akwa Ibom, 3,867 in Lagos, 1,904 in Edo, and 362 in Bayelsa have been devolved using the app. In comparing the app to previously used ones, all community pharmacies stated that the app was simpler and easier to use.

> Bez Pharmacy, Cross River State, stated the app is "… simpler and sleek."
>
> Siban Pharmacy, Akwa Ibom State, reiterated the app is "… easy to use; it's an easy way to capture my clients' refills."
>
> El-Charis Pharmacy, Akwa Ibom State, stated, "I have all my clients' details safe and well arranged in one application on my device."
>
> Wessa Pharmacy, Cross River State, stated the app "…has a great interface and easy to navigate unlike [the previous app]."
>
> Jomel Pharmacy, Akwa Ibom State, said the app is "…very user-friendly; I am able to monitor and update refills for the patients."

PART 2: ADAPTING THE DDD APP

Technical requirements

The DDD App is intended to be installed and used on a variety of platforms including Android and iOS. The app can also be used on laptops and desktop computers through an adaptation using Angular 8.0. Hence, the system is designed to meet industry standards for database management systems and application servers. When adapting the app, some attributes should not be changed in order to allow third-party and peripheral development efforts. The following attributes should be maintained for optimal use:

- Ability to run on any major database management system
- Ability to run on any mobile architecture component
- Extensibility to address local functional requirements
- Ability to be used both off-line and online

Architecture

The DDD App is written in Java and has a three-layer architecture: the presentation layer, the persistence layer, and the business layer. The application programming interface (API) allows the presentation layer to communicate with the other two layers: persistence and the business. The information below outlines the technical architecture of each layer and the API for the DDD App software. The DDD App is made up of JAVA classes and packages with several XML resources.

The Presentation Layer

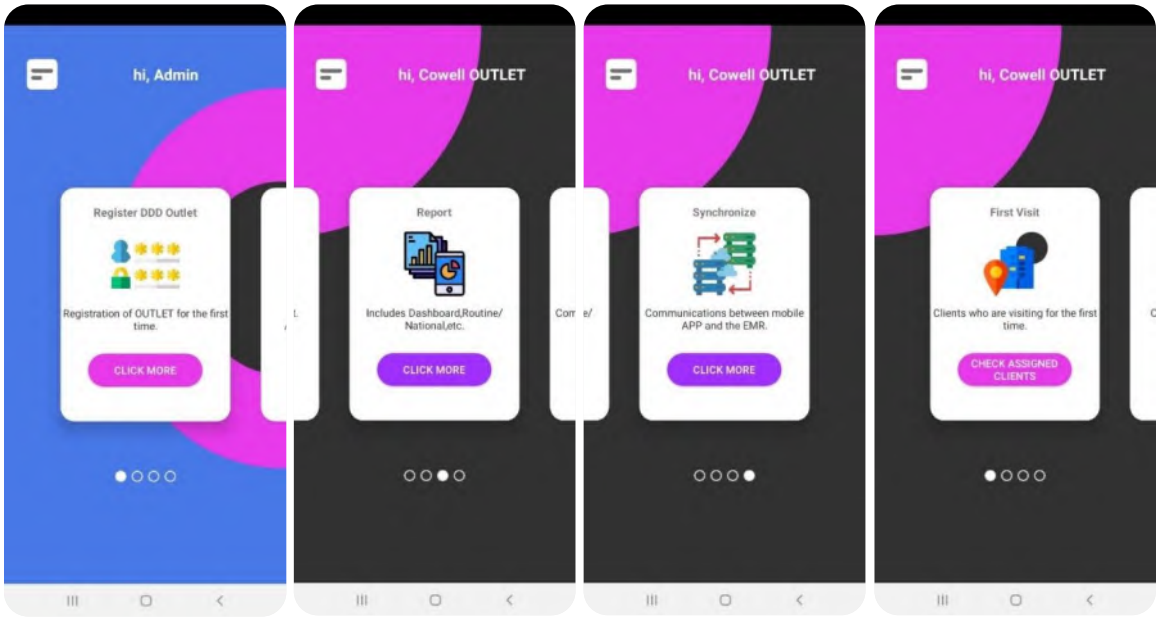
The presentation layer is the user interface (UI), which dictates the user experience and is developed using extensible markup language (XML). It is based on an Android mobile application, and the iOS follows the Android Architecture Room Database Component pattern. The architecture of the presentation layer of Android and iOS are in Annex 2.

Visual Design

The visual design of the DDD App is simple and easy-to-use with a common appearance that is achieved by using the XML page fragments in the requested pages. This design allows the DDD App to display the desired content, including the data entry fields, and to make navigation easy from one page to the next.

The main menu contains links to the various modules and functions including Registration of DDD outlets, Reports, Inventory, Synchronization, First Visit, Re Visit, and Language Configuration. The language configuration module allows the user to choose the language they prefer for the app. The options for the languages can be adapted. The options of the modules in the main menu are adaptable to the local context.

Figure 6. Images of DDD App main menu



Desktop User Interface Architecture

The DDD App can be adapted to use on a desktop or laptop computer. The website for using the DDD App on a computer is produced using the Angular 8.0 JavaScript framework (Annex 3), which conducts the following processes:

- **Event Binding:** a function that listens for and responds to the demands, actions, or inputs made by the user actions such as clicking “save”, entering data, moving to the next page, etc.
- **Property Binding:** is used to transmit data from the component class to HTML and allows you to interpolate values computed from your application data.
- **Services and Dependency Injection:** Angular 8.0 developers construct a service class for data or functionality that is not tied to a single view that they wish to share across components.

Application Programming Interface (API)

The DDD API is part of the Persistence Layer. This receives information from the UI and translates the information to the server. The API consists of the following:

- **Secure Sockets Layer (SSL) setup on the DDD server:** the security measure that prevents the API from being vulnerable to hacking or infringement of data.
- **Liquibase and Flyway for database management:** allows for easy updates to the variables and indicators that the DDD App collects.
- **JPA Persistence Layer Repository:** where Structured Query Language (SQL) coding is implemented.

- **Elastic Search for efficient client search:** library in the API that can pull entered data to provide a quick calculation of aggregate data (e.g., age using date of birth, total number of female clients, total number of clients devolved, etc.)
- **SMTP.gmail.com for client email reminders:** allows the DDD App to send emails (e.g., when sending the activation code to a DDD outlet, etc.)
- **SMS gateway:** allows the DDD App to send SMS (e.g., for auto-reminders to clients who miss appointments, etc.)

The Persistence Layer

The persistence layer is built on Room Database. This enables the DDD App to run on any relational database management system (RDBMS). Most important objects have their corresponding data type and data access object (DAO) store procedure implementation. A store implementation provides create, read, update, and delete (CRUD) operations and queries for DAO (e.g., client DAO, devolved DAO, etc.), which offers methods such as saving client data, deleting client ID, updating client data, and saving ARV regimen, etc.

The Business Layer

The services found in the business layer provide methods that delegate to a corresponding method in the persistence layer or contain simple and self-explanatory logic. Some services are more complex and use DAO connections and SQL queries to improve performance. These are elaborated in the following sections.

Data Synchronization Service

Data synchronization in the DDD App is based on REST (representational state transfer). Objects are transferred from the business layer in the JSON (JavaScript Object Notation) format and converters convert the Java objects for the persistence layer. Using Retrofit 2 Library, the business layer receives information from the presentation layer through the API and saves it in the Room Database.

In the second phase, the requesting computer receives data from the server and saves it in its database. Both phases alternate until the databases are synchronized. A computer receives data originating from only its facility, possibly from other stand-alone installations. This has the effect of equalizing the installations.

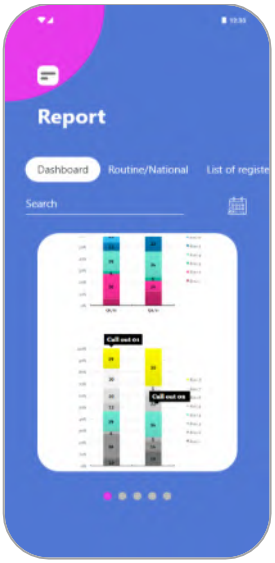
Reporting Services

The reporting services heavily utilize DAO connections and SQL statements for enhanced performance. The reporting is complex, and processing is delegated to specialized classes that collaborate. Each major reporting theme has an associated processor.

The reports are sent to JasperReports, which sets their form and format. The formats currently available are portable document format (PDF) and comma- separated values (CSV) through MS Excel but more can be configured. The Apache POI library is explicitly used to generate MS Excel (.csv) and PDF (.pdf) reports.

The charts and graphs are rendered with a JavaScript library called Highcharts. Java object collections are converted to JSON objects and passed to Highcharts.

Figure 7. Example of the reporting dashboard



SMS and email service

SMS and emails in the DDD App are implemented with the SMSLib library and the simple mail transfer protocol (SMTP) library through an internet gateway that can be used in place of a modem in a resource-constrained area.

Framework stack

The following frameworks are used in the Mobile Application Software Development Kit for both Android and iOS platforms. Application frameworks or dependencies can be used. The following are a few sample codes with UI:

- Hibernate (https://hibernate.org/)
- Apache Commons (https://commons.apache.org)
- Highcharts (https://www.highcharts.com/)
- Apache Ant (https://ant.apache.org)

ANNEXES

Annex 1. Summary of DDD App Functions for Hub Facilities and DDD Outlets

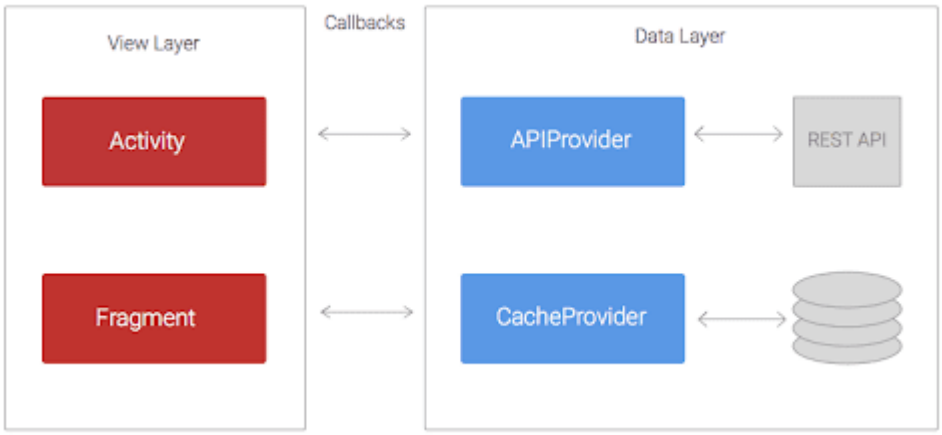
Functions	For hub facilities	For DDD outlets
Register participating DDD outlets	Yes	
Assign clients to specific DDD outlets	Yes	
Access the list of clients devolved from a hub facility to the assigned outlet (i.e., hub facilities can see the list of clients devolved; outlets can see the list of clients devolved to them)	Yes	Yes
Track clients' appointments, including previous and next refill, and viral load appointment dates	Yes	Yes
Keep a record of clients who discontinue services at the DDD outlet, including the reason for discontinuation	Yes	Yes
Screen for TB		Yes
Assess client ART adherence		Yes
Assess client ART adverse drug reactions		Yes
Compile a monthly list of dugs (ARV regimens and other medications) and the quantities needed by each DDD outlet based on the clients devolved to each outlet and the requests for drugs from DDD outlets	Yes	
Request for drugs from the hub facility		Yes
Track refill history for each client		Yes
Visualize real-time outlet service delivery data, including the number of clients served and the appointment-keeping rates	Yes	Yes
Send pre-appointment reminders	Yes	Yes
Track clients who miss appointments and send SMS reminders	Yes	Yes
Monitor drug inventory, including the number of drugs requested, received, dispensed, and remaining		Yes
Generate monthly program reports with customizable key indicators, including the number of clients devolved to outlet(s) during the month, total number of clients who picked up a refill at each outlet, number of clients who defaulted, and number of clients who were referred back to the facility	Yes	
Generate a customizable national or PEPFAR monthly summary report, including relevant key indicators	Yes	Yes
Synchronize data with the server	Yes	Yes
Tracking stock across different service delivery points	Yes	Yes

Annex 2. Presentation Layer Architecture of Android and iOS

Android Mobile User Interface Architecture

1. Activities
2. Fragment
3. API provider (rest API)
4. Cache provider

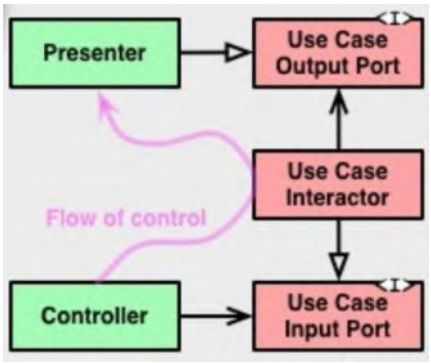
Presentation Layers of the Android Mobile APP



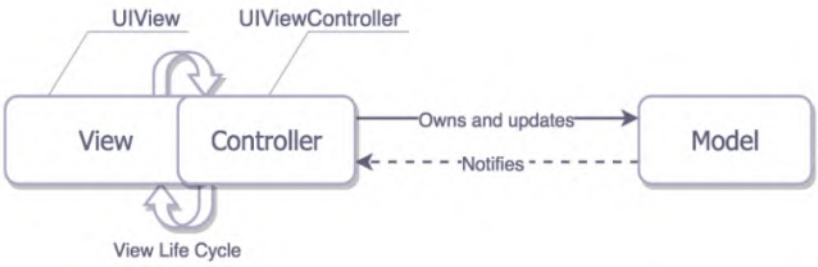
iOS Mobile User Interface Architecture

The standard iOS mobile app architecture can be divided into four blocks:

1. Kernel level (Core OS) — works with the file system, controls the validity of various certificates belonging to the applications; responsible for security of the entire system; contains low-level access to elements of the device.
2. Core services (Core Service) — provides access to databases and file controls.
3. Media level (Media) — contains tools that allow for processing most media data formats.
4. Interface level (Cocoa Touch) — has many elements for creating mobile interfaces and provides the remaining layers with information coming from the user.



Presentation Layer of iOS Mobile App



Annex 3. Implementing Angular 8.0 JavaScript Framework for DDD App Adaptation for Desktop

Router is an NgModule in Angular 8 providing a service that allows developers to design a navigation path among the various application states and view hierarchies in their projects. **It functions in the same way as a browser's navigation:**

- Enter a URL in the address bar, the browser takes user to that page.
- Click a link on a page, the browser takes user to a new page.
- Use a browser's back or forward buttons, the browser moves backward or ahead based on pages user has visited in the past.

Component Directives: Component directives are used in the main class. They describe how the component should be processed, created, and used during runtime.

Structural Directives: Structural directives begin with the symbol *. These directives are used to alter and change the DOM element structure. For instance, the *ngIf, *ngSwitch, and *ngFor directives.

- ***ngIf Directive:** allows user to Add/Remove DOM Element.
- ***ngSwitch Directive:** allows user to Add/Remove DOM Element. It is similar to the switch statement of C#.
- ***ngFor Directive:** used to repeat a portion of the HTML template once per each item from an iterable list (Collection).

Attribute Directives: Attribute directives are used to change the look and behavior of the DOM elements. For example, ngClass directive, and ngStyle directive, etc.

- **ngClass Directive:** The ngClass directive is used to add or remove CSS classes to an HTML element.
- **ngStyle Directive:** The ngStyle directive allows users to modify the style of an HTML element using the expression. The ngStyle directive can also be used to dynamically change the style of an HTML element.