

mUzima PHR documentation for version 1.0

This application will be tethered to OpenMRS, provide decision support features through notification and alerts, provider will be able to view patient entered data, the application will be patient controlled and it will be developed on native android platform. The whole system will be comprised of three components namely;

1. mUzima PHR application
2. OpenMRS PHR Module
3. OpenMRS API

1. mUzima PHR application

An android application that is part of the large mUzima suite. Prior to patients installing this application, they will be required to register with OpenMRS system administrator to be assigned a username and a password. Consumers will then install this application on their mobile devices and login using the credentials issued to them.

First time login

User enters username and password then clicks login button, this action hits phr module authentication endpoint with two parameters; username and password. If the two parameters match a record in phr_user table then authentication is successful. After successful authentication it will run an initial setup to download patient specific forms and data before loading the application.

Subsequent Login

The username and password will be stored locally on the device and so authentication will be done at device level, if successful application will be loaded.

Saving patient entered data

mUzima phr will post the data to phr module to the post obs end point that validates authorization before forwarding the request to OpenMRS REST.

Synchronization

Synchronization will be automated through two CRON jobs for mUzima PHR; one that will run once an hour to check if there is any unsynchronized data on the device, the second runs daily to check if there is new data on the server like new forms and obs.

2. OpenMRS PHR Module

Intermediary between mUzima PHR app and OpenMRS API. Performs the following administrative processes; managing phr users, managing programs eg assigning patients to programs, managing mUzima forms eg assigning forms to programs. This module will also be responsible for consumer authentication. When mUzima PHR requests reach this module, it will check the authenticity if valid, forwards them to OpenMR REST that exposes OpenMRS API. The phr module will expose the following end points for mUzima phr;

- GET encounters
- GET obs
- GET concepts
- GET forms
- GET encounters
- GET demographics
- GET providers
- POST obs

Access control

The PHR module will have two layers; web application which consists of the angular2 web pages and the API consisting of the background services to retrieve and persist data.

Web application

Configured with a require tag class that will be included in web pages to check if a user has the privilege(s) to view a page.

API

This will contain services handled through interfaces. @PHRAuthorized annotation will be defined to manage service layer access. This will make use of spring framework method interceptors to implement authorization advice. An authorization interceptor bean is defined in application-context.xml file of spring and listed among service interceptors . By preceding a method definition by @PHRAuthorized annotation will compel spring framework to check if the user accessing a resource is eligible, if not it throws authentication exception.

Data model

The following tables will be required;

1. phr_user table

id

uuid

person_id

program_id

username

password

2. phr_program table

id

uuid

name

3. phr_program_forms table

id

uuid

form_uuid

program_uuid

3. OpenMRS API

No changes will be made on this component except the addition of a single phr_user in openmrs user table. This API will expose all services, communication between OpenMRS and the phr module will be through the phr_user credentials. PHR module will also make use of existing mUzima framework for managing forms (e.g uploading form templates) and processing encounter form data.