StockStatusMonitoring_Technical Documentation

Release 1.0

SCI-ezzy

August 05, 2015

Contents

This documentation should provide an overview of the entire stock status monitoring tool and a description of the scope of the system and its intended usage. The scope should also describe external interfaces to the system, external dependencies and provide a brief overview of the 'characteristics' of the system, commenting on aspects such as deployment of the system, requirements of the system for it to run etc...

There is a downloadable PDF version of this documentation, a mobile friendly EPUB version and a downloadable HTML version.

CHAPTER 1

Table of contents

1.1 Overview

The Stock Status Management tool analyzes data from various sources including:

1. Facility and County Level Stock Data from DHIS.
2. National Level Stock Status data from the Supply Chain Agencies (KEMSA)
3. Pipeline information based on incoming shipments per funding agency

This data is aggregated and analyzed for the different malaria commodities and a 2 pager report generated that indicates to management, the months stock status and pipeline status and pipeline status of malaria commodities.

This tool is meant to amend and enhance the current 2 pager tool used by MCU to accomodate additional requirements including representation of Central and county level data for malaria commodities, and to color code the diffeent pending shipments based on the timelines for delivery.

1.2 Software Environment Tools and Components

The Malaria Stock Status Management system needs to be installed on a web server.

1.2.1 Web server software requirements

1. PHP 5.3.2+
2. One of the following operating systems:
    - Linux/Unix/BSD
    - Microsoft Windows XP SP3, Vista, Windows 7, Server 2008, Server 2008 R2
    - Mac OS X 10.4+

    (a) **Database:**

    - MySQL 5.0+

    (b) **Web server:**

    - Apache 2.0+

1.2.2 Web server hardware requirements

Hardware requirements vary widely depending on the traffic to the system, the complexity of its logic (i.e., PHP), and its size (i.e., database.) By default, all pages are dynamic, and thus access both the database and execute PHP code to generate.

The recommendations for the hardware are:

1. Atleast 50 GB disk space in the server.
2. Atleast 5 GB RAM

1.2.3 Client side requirements

The malaria stock status is designed to work well with Google Chrome, Mozilla Firefox and Internet Explorer 8+. The application works well across Windows, Linux, and Mac operating systems.

1.3 Frameworks

The system is built using **Codeigniter**, a php framework.

CodeIgniter is an Application Development Framework - a toolkit - for people who build web sites using PHP. Its goal is to enable you to develop projects much faster than you could if you were writing code from scratch, by providing a rich set of libraries for commonly needed tasks, as well as a simple interface and logical structure to access these libraries. CodeIgniter lets you creatively focus on your project by minimizing the amount of code needed for a given task.

1.3.1 Source Code Structure

The source code is organized into **Model-View-Controller** development pattern. This is because CodeIgniter is based on the Model-View-Controller development pattern. MVC is a software approach that separates application logic from presentation. In practice, it permits the web pages to contain minimal scripting since the presentation is separate from the PHP scripting.

- The **Model** represents your data structures. Typically your model classes will contain functions that help you retrieve, insert, and update information in your database.
- The **View** is the information that is being presented to a user. A View will normally be a web page, but in CodeIgniter, a view can also be a page fragment like a header or footer. It can also be an RSS page, or any other type of "page".
- The **Controller** serves as an intermediary between the Model, the View, and any other resources needed to process the HTTP request and generate a web page.

CodeIgniter has a fairly loose approach to MVC since Models are not required.

CodeIgniter framework was chosen for:

- maximum performance
- capability, and
- flexibility in the smallest, lightest possible package.

1.4 Deployment of the system

These instructions show you how to install the malaria stock status monitoring system on any web server. Please check out operating systems specific guides for Linux server, Windows Server and Mac OSX.

1.4.1 Installation Steps

step 1

- Make sure the webserver has MySQL and PHP support.

Note: e.g in Windows server,

Step 1: **Download the installation files**. You will need to download and install the windows version of the installers for each of the below componments:

1. Apache. Apache is the HTTP (Web) server software.
2. PHP. PHP is the general-purpose scripting language, especially suited for Web development, that will be used.
3. MySQL Server. MySQL is the database server/software will be used.

Step2: Install Apache. Run the Apache install file. You will need to follow the prompts and respond accordingly. When you reac

1. Network Domain: localhost
2. Server Name: localhost
3. Administrator Email: (any email address).

Step 3: **Install PHP**. Run the PHP installation file which you downloaded in Step 1.

Step 4: Install MySQL. Run the MySQL installation file which you downloaded in Step 1.

- Select the option for the "Typical" isntallation.
- You will then click on "Install".

Step 2

- Put the project into the webroot of the server. For Ubuntu servers, the webroot is in *var/www/html* while in Windows server, its located in xampp(WAMP)/htdocs.

Step 3

Configure the database settings to fit the application that is to be deployed. Visit the domain or IP address in your web browser. After a couple of minutes, the web application will be set up.

1.4.2 Deployment on Linux (Unix) servers

It is important to ensure you check the Server Requirements list before deploying on your unix server.

At a high level you will need a:

- Web server ie Apache

- Database ie MySQL
- PHP

You should have a working LAMP (Linux + Apache + MySQL + PHP) installation before the actual deployment of the system.

Note: The user has to be root when installing applications by typing *sudo su* on the terminal.

For Ubuntu servers, visit this page for LAMP installation.

Deploying the system on Ubuntu server

- Once all of the above requirements are installed, the application folder is uploaded into the web directory.
- Upload the database file into the phpMyadmin of the web server.
- Configure the databse for use by the application.
- The application can now be accessed throught the browser using the IP address or the defined url.

1.4.3 Windows server with WAMPServer 2.5+

Installing WAMP

1. Go to the WampServer download page.
2. You will first need to download and install the suggested Visual C plus plus Redistributable for Visual Studio 2012 Update 4 BEFORE installing WampServer.
3. Next, download the WampServer installer and the run the installer. By default, it will install to C:wamp. You can choose your own install path if you wish; however note we will refer to the c:/wamp in the test of this tutorial.
4. Once WampServer has been installed and launched, you will see a small "W" in the task bar, next to the clock. If everything is working, then it will be green. If it's orange or red, then something is likely misconfigured. See the Troubleshooting section below. If you can't see the "W", then WampServer hasn't been started and you should start WampServer from the start menu.
5. Left-click the "W", then select Apache -> Apache Modules -> Rewrite Module. The "W" will flick to orange, and then return to green.
6. Left-click the "W", then select MySQL -> my.ini. At the very bottom of the file, and add the following to a new line without the quotes): "lower_case_table_names = 2". Save the file, close #. Notepad and left-click the "W", and select 'Restart all services'. This is used to ease the transition between a Windows-based install and a Linux-based install where database case-sensitivity is important.

1.5 Project structure

The malaria stock status monitoring tool contains three parts:

1. Assets
2. Client
3. Database

Each part forms an important part of this project.

1.5.1 Assets

These are the resources needed in this project.

It contains:

- CSS
- Bootstrap
- PHP - CodeIgniter framework
- JavaScript

1.5.2 Client

This is the presentation and user interface logic.

It contains scripts that will display on the browser. The client side is linked to the server side so that the user can make use of the server side logic.

1.5.3 Database

This is the database logic. It contains scripts for database authentication and connection creation as well as those for inserting, fetching and updating items on the database.

The project's landing page is at: http://41.89.93.232/stock-monitoring/

1.6 Configurations

- Upload the System folders and files to the server. Normally the index.php file will be at your root.
- Open the application/config/config.php file with a text editor and set your base URL. If you intend to use encryption or sessions, set your encryption key.
- Open the *application/config/database.php* file with a text editor and set your database settings.
- If you wish to increase security by hiding the location of your CodeIgniter files you can rename the system and application folders to something more private. If you do rename them, you must open your main index.php file and set the *$system_path* and *$application_folder* variables at the top of the file with the new name you've chosen.
- For the best security, both the system and any application folders should be placed above web root so that they are not directly accessible via a browser. By default, **.htaccess** files are included in each folder to help prevent direct access, but it is best to remove them from public access entirely in case the web server configuration changes or doesn't abide by the .htaccess.
- If you would like to keep your views public it is also possible to move the views folder out of your application folder.
- After moving them, open your main index.php file and set the $system_path, $application_folder and $view_folder variables, preferably with a full path, e.g. '/www/MyUser/system'.

1.7 Development tools

Below is a description of the development tools and technologies used to develope the malaria stock status monitoring tool.

1.7.1 Database - MySQL

- A MySQL database is used to store data.
- However, another database like PostgreSQL can also be used.
- The choice of the database was purely due to:
  - The rapid turn around time.
  - It manages memory very well: MySQL server has been thoroughly tested to prevent memory leaks.
  - It runs on many operating systems: MySQL runs on many operating systems, including Novell NetWare, Windows,Linux, many varieties of UNIX* (such as Sun* Solaris*, AIX, and DEC* UNIX), OS/2, FreeBSD*, and others.
  - It supports several development interfaces: Development interfaces include JDBC, ODBC, and scripting (PHP and Perl), letting you create database solutions that run not only in your NetWare 6.5 environment, but across all major platforms, including Linux, UNIX, and Windows.

1.7.2 PHP and Apache2 for the server side

- The scripting language PHP and web server Apache2 were used for the server side logic.
- **CodeIgniter** PHP framework was used to develop the system.

1.7.3 JavaScript

JavaScript was used in the client side for validation and for responsives purposes.

1.7.4 HTML5, Bootstrap and JavaScript for user interface

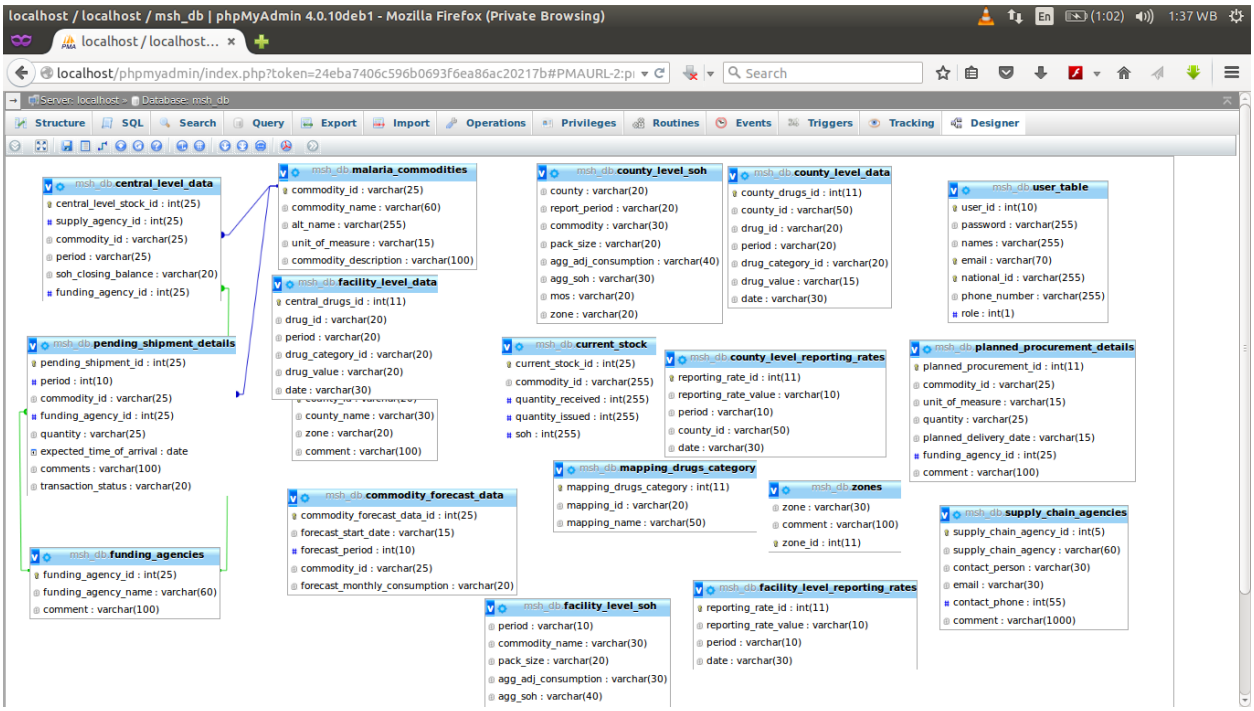
Since this is a Web application, the interface is coded in HTML5 with Bootstrap framework incorporated to provide a better user experience JavaScript is also used to make the interface responsive and event/data driven.

1.8 The Database Schema

1.8.1 Introduction

This documentation set describes the data characteristics of the tables and columns in the malaria stock status monitoring database: including datatypes and sizes, nullability, index, sequence, key and constraint information. Additional commentary is also provided for key tables and columns.

Here is the screenshot of the database schema:



1.8.2 Database Tables

1. User_table
2. County_level_soh
3. Commodity_forecast_data
4. Current_stock
5. Pending_shipment_details
6. Counties
7. Central_level_data
8. Facility_level_data
9. County_level_reporting_rates
10. Zones
11. Supply_chain_agencies
12. County_level_data
13. Mapping_drugs_category
14. Facility _level_reporting_rates
15. Funding_agencies
16. Facility_level_soh
17. Malaria_commodities
18. Planned_procurement_details

1.8.3 Tables attributes and design

User_table

This table consists of a list of the system users' attributes with the following columns:

- user_id
- password
- names
- email
- national_id
- phone number
- role

Field	Type	Null	Key	Default	Extra
user_id	int(10)	NO	PRI	NULL	auto_increment
password	varchar(255)	NO		NULL	
names	varchar(255)	NO		NULL	
email	varchar(70)	NO	UNI	NULL	
national_id	varchar(255)	NO	UNI	NULL	
phone_number	varchar(255)	NO		NULL	
role	int(1)	NO		0	

County_level_soh

This table consists of the county level SOH attributes. It has the following columns:

- county
- report_period
- commodity
- pack_size
- agg_adj_consumption
- agg_soh
- mos
- zone

Field	Type	Null	Key	Default	Extra
county	varchar(20)	NO		NULL	
report_period	varchar(20)	NO		NULL	
commodity	varchar(30)	NO		NULL	
pack_size	varchar(20)	NO		NULL	
agg_adj_consumption	varchar(40)	NO		NULL	
agg_soh	varchar(30)	NO		NULL	
mos	varchar(20)	NO		NULL	
zone	varchar(20)	NO		NULL	

Commodity_forecast_data

This table consists of the commodity forecast attributes. It has the following columns:

- commodity_forecast_data_id
- forecast_start_date
- forecast_period
- commodity_id
- forecast_monthly_consumption

Field	Type	Null	Key	Default	Extra
commodity_forecast_data_id	int(25)	NO	PRI	NULL	auto_increment
forecast_start_date	varchar(15)	NO		NULL	
forecast_period	int(10)	NO		NULL	
commodity_id	varchar(25)	NO	MUL	NULL	
forecast_monthly_consumption	varchar(20)	NO		NULL	

Current_stock

This table consists of the current stock attributes. It has the following columns:

- current_stock_id
- commodity_id
- quantity_received
- quantity_issued
- soh

Field	Type	Null	Key	Default	Extra
current_stock_id	int(25)	NO	PRI	NULL	auto_increment
commodity_id	varchar(255)	NO		NULL	
quantity_received	int(255)	NO		NULL	
quantity_issued	int(255)	NO		NULL	
soh	int(255)	NO		NULL	

Pending_shipment_details

This table consists of pending shipment attributes. It has the following columns:

- pending_shipment_id
- period
- commodity_id
- funding_agency_id
- quantity
- expected_time_of_arrival
- comments
- transaction_status

Field	Type	Null	Key	Default	Extra
pending_shipment_id	int(25)	NO	PRI	NULL	auto_increment
period	int(10)	NO		NULL	
commodity_id	varchar(25)	NO	MUL	NULL	
funding_agency_id	int(25)	NO	MUL	NULL	
quantity	varchar(25)	NO		NULL	
expected_time_of_arrival	date	NO		NULL	
comments	varchar(100)	NO		NULL	
transaction_status	varchar(20)	NO		pending	

Counties

This table consists of counties attributes. It has the following columns:

- county_id
- county_name
- zone
- comment

Field	Type	Null	Key	Default	Extra
county_id	varchar(20)	NO	PRI	NULL	
county_name	varchar(30)	NO		NULL	
zone	varchar(20)	NO		NULL	
comment	varchar(100)	NO		NULL	

Central_level_data

This table consists of central level data attributes. It has the following columns:

- central_level_stock_id
- supply_agency_id
- commodity_id
- period
- soh_closing_balance
- funding_agency_id

Field	Type	Null	Key	Default	Extra
central_level_stock_id	int(25)	NO	PRI	NULL	auto_increment
supply_agency_id	int(25)	NO	MUL	NULL	
commodity_id	varchar(25)	NO	MUL	NULL	
period	varchar(25)	NO		NULL	
soh_closing_balance	varchar(20)	NO		NULL	
funding_agency_id	int(25)	NO	MUL	NULL	

Facility_level_data

This table consists of facility level data attributes. It has the following columns:

- central_drugs_id

- drug_id
- period
- drug_category_id
- drug_value
- date

Field	Type	Null	Key	Default	Extra
central_drugs_id	int(11)	NO	PRI	NULL	auto_increment
drug_id	varchar(20)	NO		NULL	
period	varchar(20)	NO		NULL	
drug_category_id	varchar(20)	NO		NULL	
drug_value	varchar(20)	NO		NULL	
date	varchar(30)	NO		NULL	

County_level_reporting_rates

This table consists of county level reporting rates attributes. It has the following columns:

- reporting_rate_id
- reporting_rate_value
- period
- county_id
- date

Field	Type	Null	Key	Default	Extra
reporting_rate_id	int(11)	NO	PRI	NULL	auto_increment
reporting_rate_value	varchar(10)	NO		NULL	
period	varchar(10)	NO		NULL	
county_id	varchar(50)	NO		NULL	
date	varchar(30)	NO		NULL	

Zones

The zones table has the following attributes:

- zone
- comment
- zone_id

Field	Type	Null	Key	Default	Extra
zone	varchar(30)	NO		NULL	
comment	varchar(100)	NO		NULL	
zone_id	int(11)	NO	PRI	NULL	auto_increment

Supply_chain_agencies

The Supply chain agencies table has the following attributes:

- supply_chain_agency_id

- supply_chain_agency
- contact_person
- email
- contact_phone
- comment

Field	Type	Null	Key	Default	Extra
supply_chain_agency_id	int(5)	NO	PRI	NULL	auto_increment
supply_chain_agency	varchar(60)	NO		NULL	
contact_person	varchar(30)	NO		NULL	
email	varchar(30)	NO		NULL	
contact_phone	int(55)	NO		NULL	
comment	varchar(1000)	NO		NULL	

County_level_data

The county level data table has the following attributes:

- county_drugs_id
- county_id
- drug_id
- period
- drug_category_id
- drug_value
- date

Field	Type	Null	Key	Default	Extra
county_drugs_id	int(11)	NO	PRI	NULL	auto_increment
county_id	varchar(50)	NO	MUL	NULL	
drug_id	varchar(20)	NO		NULL	
period	varchar(20)	NO		NULL	
drug_category_id	varchar(20)	NO		NULL	
drug_value	varchar(15)	NO		NULL	
date	varchar(30)	NO		NULL	

Mapping_drugs_category

The Mapping drugs category table has the following attributes:

- mapping_drugs_category
- mapping_id
- mapping_name

Field	Type	Null	Key	Default	Extra
mapping_drugs_category	int(11)	NO	PRI	NULL	auto_increment
mapping_id	varchar(20)	NO		NULL	
mapping_name	varchar(50)	NO		NULL	

Facility_level_reporting_rates

The Facility level reporting rates table has the following attributes:

- reporting_rate_id
- reporting_rate_value
- period
- date

Field	Type	Null	Key	Default	Extra
reporting_rate_id	int(11)	NO	PRI	NULL	auto_increment
reporting_rate_value	varchar(10)	NO		NULL	
period	varchar(10)	NO		NULL	
date	varchar(30)	NO		NULL	

Funding_agenciebles

The funding agencies table has the following attributes:

- funding_agency_id
- funding_agency_name
- comment

Field	Type	Null	Key	Default	Extra
funding_agency_id	int(25)	NO	PRI	NULL	auto_increment
funding_agency_name	varchar(60)	NO		NULL	
comment	varchar(100)	NO		NULL	

Facility_level_soh

The facility level SOH table has the following attributes:

- Field
- period
- commodity_name
- pack_size
- agg_adj_consumption
- agg_soh
- mos
- zone

Field	Type	Null	Key	Default	Extra
period	varchar(10)	NO		NULL	
commodity_name	varchar(30)	NO		NULL	
pack_size	varchar(20)	NO		NULL	
agg_adj_consumption	varchar(30)	NO		NULL	
agg_soh	varchar(40)	NO		NULL	
mos	varchar(20)	NO		NULL	
zone	varchar(30)	NO		NULL	

Malaria_commodities

The malaria commodities table has the following attributes:

- commodity_id
- commodity_name
- alt_name
- unit_of_measure
- commodity_description

Field	Type	Null	Key	Default	Extra
commodity_id	varchar(25)	NO	PRI	NULL	
commodity_name	varchar(60)	NO		NULL	
alt_name	varchar(255)	NO		NULL	
unit_of_measure	varchar(15)	NO	MUL	NULL	
commodity_description	varchar(100)	NO		NULL	

Planned_procurement_details

The planned procurement details has the following attributes:

- planned_procurement_id
- commodity_id
- unit_of_measure
- quantity
- planned_delivery_date
- funding_agency_id
- comment

Field	Type	Null	Key	Default	Extra
planned_procurement_id	int(11)	NO	PRI	NULL	auto_increment
commodity_id	varchar(25)	NO		NULL	
unit_of_measure	varchar(15)	NO		NULL	
quantity	varchar(25)	NO		NULL	
planned_delivery_date	varchar(15)	NO		NULL	
funding_agency_id	int(25)	NO		NULL	
comment	varchar(100)	NO		NULL	

1.9 MOS Maps

MOS maps have been used to represent MOS status both at the county and national level.

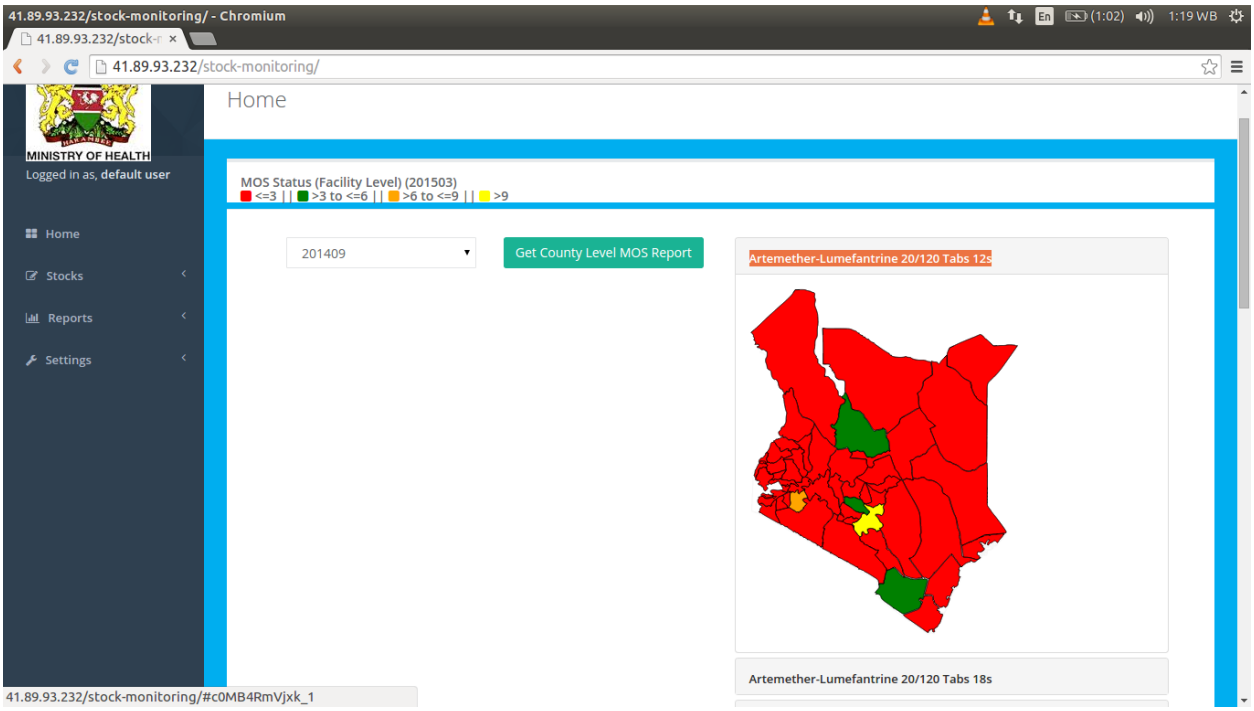
1.9.1 Color codes Key

1.9.2 County Level MOS Report

MOS report is generated of each drug based on the period selected at the county level.

Artemether-Lumefantrine 20/120 Tabs 12s

Here is the screenshot of Artemether-Lumefantrine 20/120 Tabs 12s MOS report:



Artemether-Lumefantrine 20/120 Tabs 18s

Here is the screenshot of Artemether-Lumefantrine 20/120 Tabs 18s MOS report:

Artemether-Lumefantrine 20/120 Tabs 24s

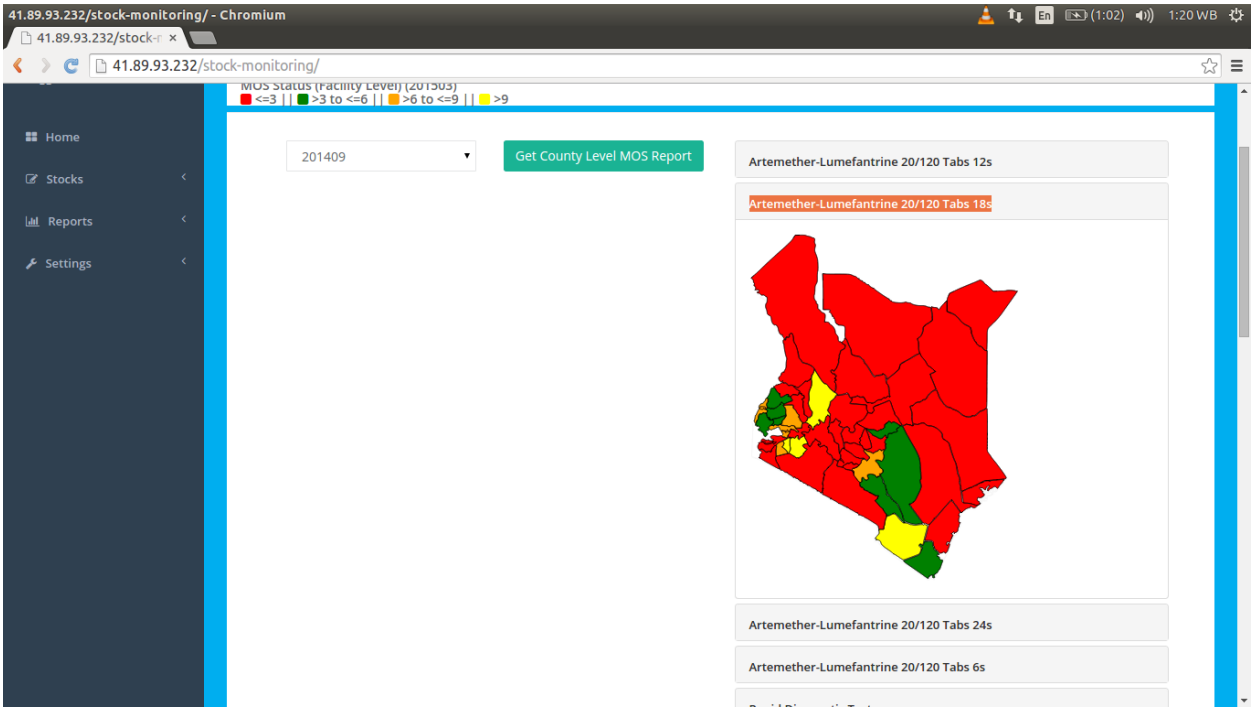
Here is the screenshot of Artemether-Lumefantrine 20/120 Tabs 24s MOS report:

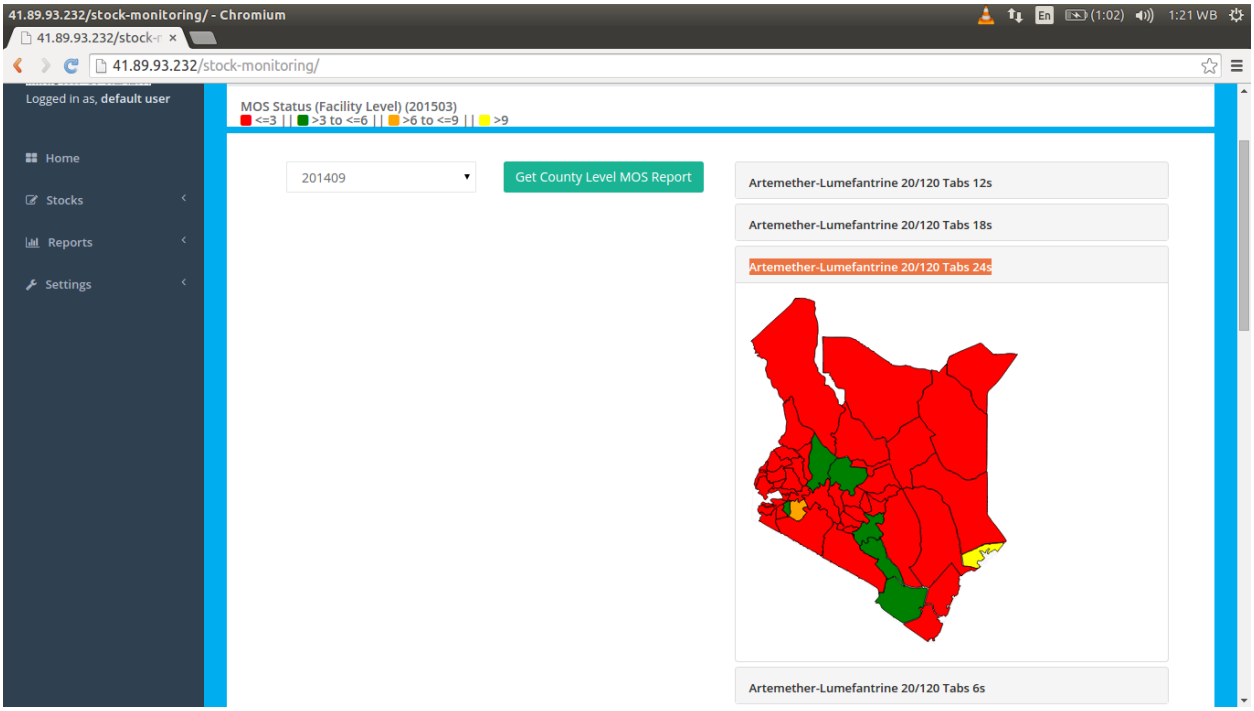
Artemether-Lumefantrine 20/120 Tabs 6s

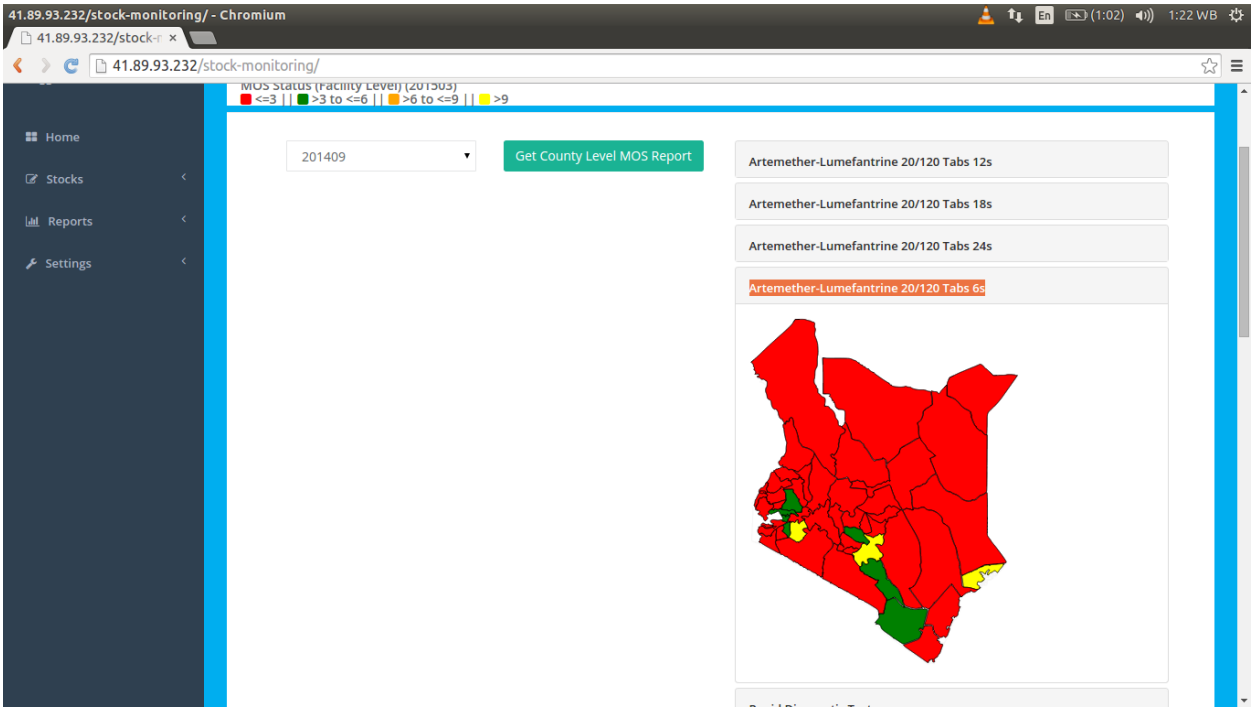
Here is the screenshot of Artemether-Lumefantrine 20/120 Tabs 6s MOS report:

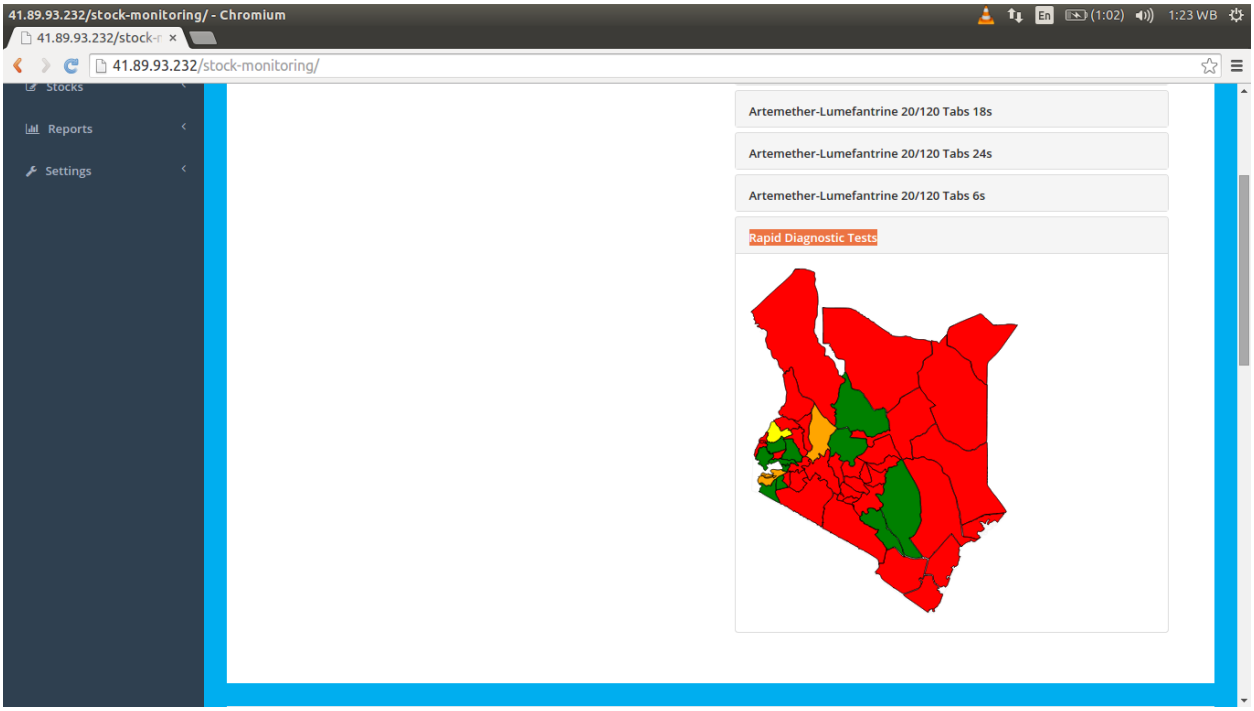
Rapid Diagnostic Tests

Here is the screenshot of Rapid Diagnostic Tests MOS report:


41.89.93.232/stock-monitoring/ - Chromium
41.89.93.232/stock-monitoring/
<=3 | | >3 to <=6 | | >6 to <=9 | | >9
Home
Stocks
Reports
Settings
201409
Get County Level MOS Report
Artemether-Lumefantrine 20/120 Tabs 12s
Artemether-Lumefantrine 20/120 Tabs 18s
Artemether-Lumefantrine 20/120 Tabs 24s
Artemether-Lumefantrine 20/120 Tabs 6s

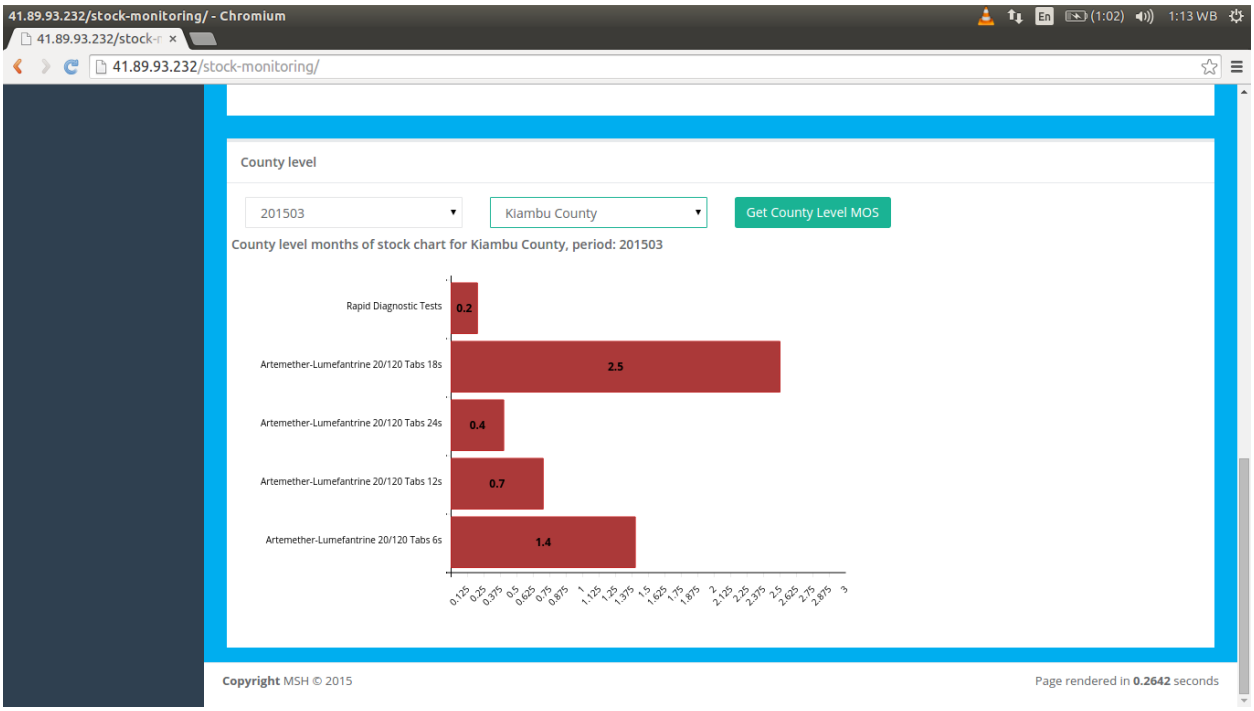

41.89.93.232/stock-monitoring/ - Chromium
41.89.93.232/stock-monitoring/
Logged in as, default user
MOS Status (Facility Level) (201503)
<=3 | | >3 to <=6 | | >6 to <=9 | | >9
Home
Stocks
Reports
Settings
201409
Get County Level MOS Report
Artemether-Lumefantrine 20/120 Tabs 12s
Artemether-Lumefantrine 20/120 Tabs 18s
Artemether-Lumefantrine 20/120 Tabs 24s
Artemether-Lumefantrine 20/120 Tabs 6s


41.89.93.232/stock-monitoring/ - Chromium
41.89.93.232/stock-monitoring/
<=3 | | >3 to <=6 | | >6 to <=9 | | >9
Home
Stocks
Reports
Settings
201409
Get County Level MOS Report
Artemether-Lumefantrine 20/120 Tabs 12s
Artemether-Lumefantrine 20/120 Tabs 18s
Artemether-Lumefantrine 20/120 Tabs 24s
Artemether-Lumefantrine 20/120 Tabs 6s


41.89.93.232/stock-monitoring/ - Chromium
41.89.93.232/stock-monitoring/
Reports
Settings
Artemether-Lumefantrine 20/120 Tabs 18s
Artemether-Lumefantrine 20/120 Tabs 24s
Artemether-Lumefantrine 20/120 Tabs 6s
Rapid Diagnostic Tests

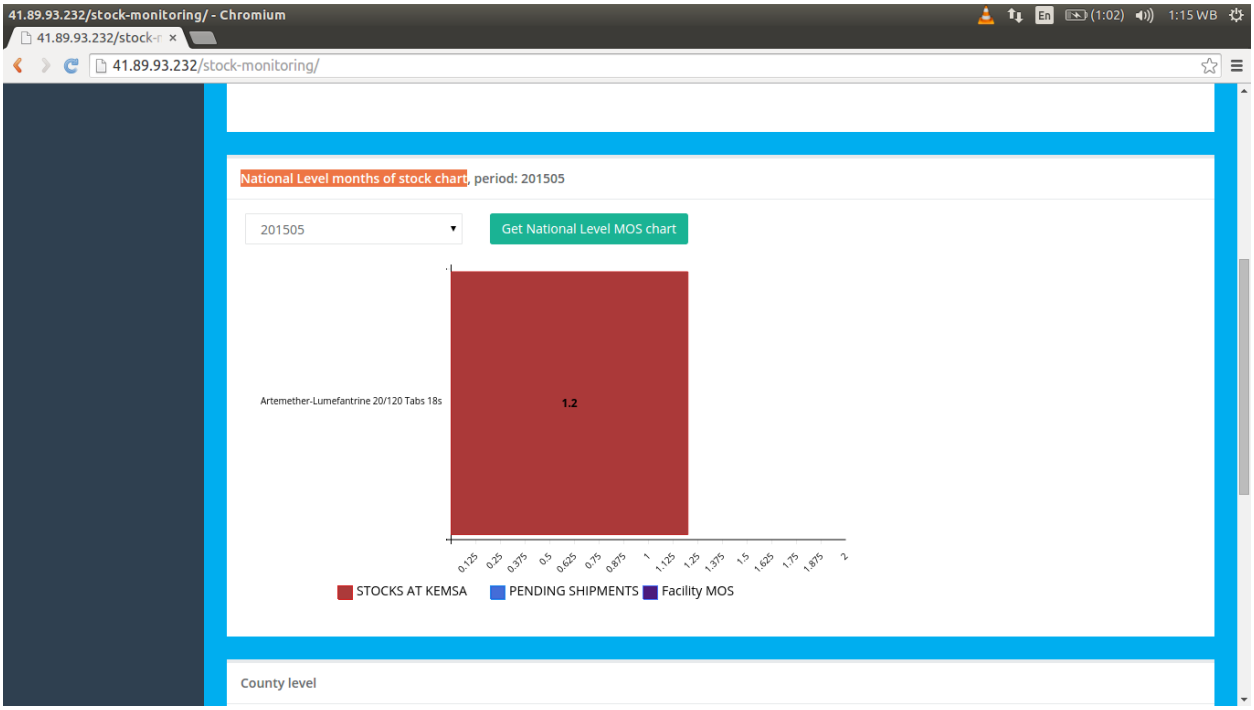
1.9.3 County level months of stock(MOS) chart

Individual county level months of stock(MOS) Charts are produced here as per the selected period



1.9.4 National Level months of stock(MOS) chart

National level months of Stock (MOS) charts are produced as per the selected period from the drop-down menu.


41.89.93.232/stock-monitoring/ - Chromium
41.89.93.232/stock-monitoring/
National Level months of stock chart, period: 201505
201505
Get National Level MOS chart
Artemether-Lumefantrine 20/120 Tabs 18s
1.2
0.125
0.25
0.375
0.5
0.625
0.75
0.875
1
1.125
1.25
1.375
1.5
1.625
1.75
1.875
2
STOCKS AT KEMSA
PENDING SHIPMENTS
Facility MOS
County level