

System Performance and Utilization Module

Primary mentor	Pascal Brandt
Backup mentor	Jan Flowers
GSoC student	Vineet Kumar

Abstract

Extend the OpenMRS module "System Performance and Utilization Module" to supply additional information on system performance and utilization in order to monitor reliability and impact of the electronic medical record system installation. This module monitors OpenMRS and transfers specific system indicators, possibly using the Performance and Utilization Monitoring Program framework developed for an OpenMRS network implementation in Kenya. The expansion of indicators could include system indicators (CPU load, etc) and application level indicators (# of opened charts, etc), as well as, the further development of the visualization of these indicators (drill-down to view details within those statistics - which users logged in, which charts were opened, etc).

Theory

The major focus of the module currently is to show indicators of following type:-

- **System Level Indicators** - Memory Usage
- **Application Level Indicators** - Number of Login User

Basic Theory behind the Indicator Working, Which is require for future development and extension

1. Selecting Indicator purpose : - this could be a System or Application Level Indicator. our Aim is to consider what kind of System Usage and Application Usage we have to show.

2. Update And Syncing data from Database:- The data which have to shown on indicators is update and retrieved from database.

- Created HibernateDAO object of MetricTypeObject then called that in PerformanceMonitoringServiceImpl service to add/remove/modify MetricType values.
- Instead of mapping hbm.xml here used JPA annotations.
- Created MetricType object and annotate it with Table and column annotations as below. As we define table name, primary key, column names here there is no need to refer a .hbm.xml file Also the class needed to implement Serializable in order to define Id, which is primary key.
- When the module is loaded the new table is created now and you can add data there using DAO objects.

3. Updating Data Base:- Querying the used memory for this is done using the Java Management beans where it expose both use memory and free memory values with MemoryMXBean. The following link is explaining about the structure of the beans API.

1. <http://docs.oracle.com/javase/7/docs/api/java/lang/management/ManagementFactory.html>
2. <https://docs.oracle.com/javase/7/docs/jre/api/management/extension/com/sun/management/OperatingSystemMXBean.html>

Objectives

1. Add System Monitoring Indicators: System monitoring is mainly focused on gathering and presenting information on low level system indicators like CPU load, memory utilization etc.
2. Add Application Monitoring Indicators: The application monitoring is introduced to monitor what usage patterns are occurring number of opened charts during a set time period, number of modified charts during a set time period, number of active logins etc.
3. Advanced Visualization of Statistics: Ability to drill-down into the graphs for stratified views, such as, which roles/users were logged in, what kind of charts were opened, etc.
4. Export of Data: CSV file export, as well as, exchange of data into a larger framework for dashboard view across sites

Extra credit

1. Provide user interface to configure new indicators to collect

Suggested communication plan

During the project implementation period the student and mentors (Pascal/Jan) will hold a weekly one hour meeting to discuss the progress of the project and decide on the tasks to be finished for each week.

Mockups



tinypic

**This image is no longer available.
Visit tinypic.com for more information.**



tinypic

This image is no longer available.
Visit tinypic.com for more information.



tinypic

This image is no longer available.
Visit tinypic.com for more information.



In short this module requires

1. Add System Monitoring Indicators: System monitoring is mainly focused on gathering and presenting information on low level system indicators like CPU load, memory utilization etc.

System Monitoring can include:

- CPU usage
- Memory Usage
- Disk Usage
- System Information

Following resource will be helpful for getting CPU load information:

1. <http://docs.oracle.com/javase/7/docs/api/java/lang/management/ManagementFactory.html>
2. <https://docs.oracle.com/javase/7/docs/jre/api/management/extension/com/sun/management/OperatingSystemMXBean.html>

2. Add Application Monitoring Indicators: The application monitoring is introduced to monitor what usage patterns are occurring number of opened charts during a set time period, number of modified charts during a set time period, number of active logins etc.

Application Monitoring can include:-

- Login Number
- Encounter Count
- Visit Count
- Observation Count
- Patient Count
- Report Run
- Cohort Count
- Concept Count
- Form Filled

3. Advanced Visualization of Statistics: Ability to drill-down into the graphs for stratified views, such as, which roles/users were logged in, what kind of charts were opened, etc. This can be done using Google Charts API, highcharts library. While we already have the data of CUP Load, Memory Utilization, Members Login, Application data etc. we can frame those data easily on a web page.

4. Export of Data: CSV file export, as well as, exchange of data into a larger framework for dashboard view across sites. Not only CSV we can also export it to excel sheets.

According to My collected data the timeline should look like this:-

Timeline (May 25 – August 24):

0. Studying the complete Module and get use to it. (by May 30)

1. Add System Monitoring Indicators (by June 25) :-

- CPU Usage - CPU Usage, CPU Time, Maximum Hourly CPU Usage, Average CPU Usage
- Disk Usage
- Memory Usage

2. Add Application Monitoring Indicators (by July 15)

- Fields Count
- Improvement In Login User Indicators

3. Advanced Visualization of Statistics. (by July 30)

- Adding tabular Data Section
- All Fields Statistics

4. Export of Data (by Aug 3)

5. Testing phase and refine phase (by Aug 10)

6. Documentation and Demo (by Aug 18)

Resources

- **Module Repository** : <https://github.com/openmrs/openmrs-module-systemmetrics>
- **Project Blog** : <http://milindarukshan.blogspot.com/>
- **Mid term Presentation** : <https://talk.openmrs.org/t/gsoc-2014-system-performance-and-utilization-module-midterm-presentation/313>
- **Final Presentation** : <https://talk.openmrs.org/t/gsoc-2014-system-performance-and-utilization-module-final-presentation/488>
- Latest Documentation : [System Performance and Utilization Module](#)

Latest Documentation

[System Performance and Utilization Module](#)

Previous Work on Idea

- [Mozambique Implementation Packet](#) (see system monitoring section)
- [Kenya Workplanning- PUMP Demo.pdf](#) (this is just an example for inspiration)
- [KenyaEMR Appliance Setup](#)
- [PUMP Build Instructions](#)