

## Group Tasks - INF5750 - Autumn 2010

Please send an email to [joakibj@ifi.uio.no](mailto:joakibj@ifi.uio.no) and [morteoh@ifi.uio.no](mailto:morteoh@ifi.uio.no) before Friday October 15. 23:59 containing:

- A subject saying "INF5750 - Project Group Application - <username>".
- 3 tasks you would like to work on.
- Whether you are able to attend the lab session on Wednesday, Thursday, or both.

### 1. Port DHIS 2 web layer from Struts 2 to Spring MVC

The DHIS 2 web layer is based on Struts 2. Lately Spring MVC version 3 was released, which provides a compelling and sophisticated alternative. This task is about porting the web modules of DHIS 2 to use Spring MVC as web framework. This task has two challenges. First, the DHIS 2 web portal solution is based heavily on Struts and must be partly re-written. Second, the web modules are based around Struts 2 action classes and must be ported to Spring MVC controller classes. Info on Spring MVC can be found here: <http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html/mvc.html>.

### 2. Extend charting functionality in DHIS 2

DHIS 2 uses the JFreeChart library for producing charts. Currently only bar and line charts are supported. This task is about exploring which charts types are relevant for DHIS 2 and implement new chart types. Potential types are area, 3D bar, bubble, 3D line, pie and 3D pie charts. The DHIS 2 charting functionality is located in the Reporting module. More information on JFreeChart can be found here: <http://www.jfree.org/jfreechart/>. Recommended for people with a flair for visualization.

### 3. MSSQL support

DHIS 2 is using Hibernate which itself has support for MSSQL. However some parts of DHIS 2 are using SQL queries directly to improve performance. Currently PostgreSQL, MySQL, H2 and Derby are supported. This task is about implementing support for MSSQL. The SQL statements are externalized through StatementBuilder interface and a MSSQL implementation must be provided. Auto-increment identifiers must also be managed. MSSQL Express is free to use and will be used for this task: <http://www.microsoft.com/sqlserver/2005/en/us/express.aspx>. Recommended for people with interest in SQL.

### 4. Facility search

DHIS 2 needs improved functionality for searching for organisation units (facilities). Meta-data can be added to organisation units through groups and group sets. This functionality should provide dynamic search options based on such group sets, listing of facilities, all information for a specific facility and export options to Excel and PDF. This function is replicating the Master

Facility List in Kenya: <http://www.ehealth.go.ke/facilities/search.aspx>. Recommended for people with interest in Java programming.

## **5. DHIS 2 on Android**

Develop a DHIS 2 Android mobile application. This application should provide basic data entry functionality plus basic tabular reporting functionality. A REST/web-service on the DHIS 2 server side will also have to be developed, which must provide the necessary services for the mobile client. More info on Android development here: <http://developer.android.com>. This will be a large group split in two for client/server development. Recommended for people with interest in mobile applications.

## **6. Improve DHIS 2 security solution**

The security solution in DHIS 2 was developed four years ago on Acegi Security and was later ported to Spring Security, however without changing the design of the solution. This task is about reading up on Spring Security, documenting and understanding the current solution in DHIS 2, propose how the design can be simplified and how the new functionality of Spring Security version 3 can be leveraged, and eventually implement those changes. More info on Spring Security here: <http://static.springsource.org/spring-security/site/>. Recommended for people with interest in security.

## **7. Scheduling of aggregation operations**

DHIS 2 provides aggregate, statistical data and is capable of aggregating data in time (periods) and space (organisation unit hierarchy). The aggregation process is CPU and memory consuming and may take some time to complete when dealing with large data sets and should be possible to schedule, e.g. to run at night when the user load on the system is low. This solution should use the nice scheduling support in Spring 3, an intro can be found here: <http://blog.springsource.com/2010/01/05/task-scheduling-simplifications-in-spring-3-0/>. Recommended for people with interest in programming.

## **8. Improve DHIS 2 modularity**

Each web module of DHIS 2 is currently a stand-alone web application and package as a WAR-file. The portal solution assembles all modules into one application. This has some negative implications in that all WAR files are constantly unpackaged/packaged when assembled, and that changes to static content are not picked up at runtime during debugging. This task is about making Struts 2 (the web framework) serve the static content from the class path, which will make it possible to package the web modules as JAR files. This will dramatically reduce build time and ease debugging. More info here: <http://struts.apache.org/2.1.8/docs/static-content.html>. Recommended for people with interest in web application development.

## **9. Improve DHIS 2 dashboard**

DHIS 2 features a dashboard which is personal to each user. The dashboard contains charts

with live data and links to reports, GIS views, RSS feeds and more. The dashboard should be improved to support i) a customizable number of charts ii) customizable RSS feeds (the current are hardcoded) iii) fallback content, meaning that certain charts/links are displayed if the user has not added any or removed her own. Have a look at [demo.dhis2.org](http://demo.dhis2.org) (admin/district) to get an idea. Recommended for people with interest in programming.

## **10. Improve internal process management**

Processes that takes a long time to complete are run as internal background processes in DHIS 2. Typically these processes are performing aggregation or validation of huge data sets. DHIS 2 uses the Cave process framework for this. The first part of the task is to replace this framwork with the standard Java Concurrency API. The second part is to implement a internal process manager, which should include functionality for viewing all running, scheduled and finished processes, as well as canceling running processes. Intro on Java Concurrency API here: <http://download.oracle.com/javase/tutorial/essential/concurrency/>, Cave framework here: <http://www.amplecode.org/wiki/Cave/ProcessFramework>. Recommended for people with interest in multi-threaded and concurrent programming.