### Human Resources Module

<table>
<thead>
<tr>
<th>Role</th>
<th>GSOC 2011</th>
<th>GSOC 2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary mentor</td>
<td>Daniel Futerman</td>
<td>Sai Manohar Nethi</td>
</tr>
<tr>
<td>Backup mentor</td>
<td>Roger Friedman Carl Leitner</td>
<td>Roger Friedman</td>
</tr>
<tr>
<td>Assigned to</td>
<td>Sai Manohar Nethi</td>
<td>Sandeep Joshi</td>
</tr>
</tbody>
</table>

#### Background

Getting people with the right skills and qualifications to the right places to deliver health care is a difficult problem, particularly in Africa. Personnel rosters may contain employees who are on educational leave or assignment elsewhere, employees may be underqualified for the positions they occupy, and qualified employees are often attracted away from Africa to Europe or the US. Therefore, health systems must constantly keep tabs on their current workforce, what additional training is needed, who can move to positions of greater responsibility, how many newly trained staff are going to be needed, and how to ensure a flow of qualified people to less desirable posts. This is a joint effort at both the clinical and governmental levels, and developing competent systems to rationalize human resources management is a priority of many countries.

#### Purpose

The purpose of the project is to develop an OpenMRS module to maintain personnel information and generate reports useful at the clinical level and compatible with a national personnel system. The design is based on iHRIS (an open source personnel management and qualifications system written in PHP). Additional or modified requirements may come from other international projects currently underway. The project includes basic functionality, but could extend in two different directions – building a distributed communications capability to connect with a central iHRIS implementation, or building additional clinically-useful capabilities such as a staff scheduler or a patient education tracker. This project also can serve as an experiment in coordinating external registry data with OpenMRS tables, in this case, using the national provider registry to maintain the OpenMRS providers table. The desired product is a packaged module like the MDR TB module which comes with a data model, API extensions, data entry forms and reports. It would likely use the Role Based Homepage module as an entry point.

#### GSOC 2012 Project Description

As can be seen [here](#), in GSOC 2011, Sai Manohar was able to implement about half the tables of the complete project. In addition, some of the installation, administration and reporting features were functional but not yet user-friendly. Also, the HR Module was one of the first modules to use Maven, and the standards for doing so had not yet been established. For 2012, we would like to (1) implement data entry forms for the remaining tables; (2) improve the installation and reporting features; (3) update our Maven configuration; and (4) prepare to use our own base page with the Role Based Homepage module rather than use the Patient Dashboard and Admin pages to access the HR functions (the admin functions will remain on the Admin page).

#### Domain Expert(s) / User(s)

- Roger Friedman
- Carl Leitner

#### Required Skills

Coding in the OpenMRS environment generally requires the following skills: Java, JSP, HTML, SQL, Spring. The following skills are desirable: Javascript, Hibernate, Liquibase, Maven. In addition, report generation requires the use of some reporting features of OpenMRS and/or a report generator such as BIRT.

- Ability to create a module in the OpenMRS environment
- Ability to create reports in the OpenMRS environment
- Ability to create forms in the OpenMRS environment

#### Objectives

1. Detailed workflow and software design for and development of a human relations module:
   a. Add tables for job title, post, education, competency, evaluation, certificate, training, training_person
   b. Create admin functions to add/edit/inactivate/upload metadata tables job title, certificate, competency
   c. Create HR manager forms to assign/remove posts to locations, providers to posts, competencies to trainings
   d. Create HR clerk forms to provider's names/addresses, certificates, general notes, reviews (type of note), incidents (type of note)
   e. Create HR clerk reports showing providers by job title and location; selected provider name/address/certificates/post history/training history/note

2. Keep technical documentation up to date with design decisions and development process.

#### Extra Credit

1. Create finished version of project technical documentation
2. Create user documentation and training materials
3. Add staff scheduler capability
4. Add staff/patient education tracking
   a. Create form to add/edit/inactivate training instances
   b. Create form to assign/remove providers or patients to training instances
   c. Create reports showing trainees by training instance, training instance by trainee, competency counts by time period
5. Design iHRIS or similar interface
6. Develop iHRIS or similar interface

Resources

- iHRIS – http://www.capacityproject.org/hris/suite/
- iHRIS use cases (focus on Manage module) -- http://www.capacityproject.org/hris/hris-toolkit/tools/ihris_documentation.html
- CDC will provide connections with international projects for MOH HR functionality

Attachments
- provider.rtf -- narrative
- Provider.xls -- crosswalk with iHRIS
- provider.pdf -- draft data model
- Provider.sql -- draft data model

Additional attachments
- hr.mwb -- MySQL Workbench file with data model
- hr2.pdf -- revised draft data model
- schema2.xls -- revised draft data model

Revised version
- hr3.mwb -- revised version 22-Jun-11
  - check the notes for changes this version
  - note the changes to pay grade logic:
    - job_title contains a comma-separated, low-to-high ordered list of pay grades for that job title
    - post contains a job_title reference but no pay grade
    - post_history contains a pay grade, which should come from the list in job_title; step increases require a new record
    - assignment does not contain pay grade
- schema3.xls -- revised schema 13-Jun-11
- ExWkSched.xls -- demo data for loading 13-Jun-11
- hr_test_110622.sql -- MySQL dump of DB containing demo data 22-Jun-11
  - Requires concept_datatype and concept_class from openmrs sample DB
  - Should comment out drop table for openmrs standard tables (person, person_name, person_address, concept, concept_name, concept_name_map, location
  - Should change INSERT to INSERT IGNORE for concept, concept_map due to true/false concepts
  - Concept word has not been updated with HR concepts, not sure that's still being used
  - All HR IDs start with 77777 (except post_history and assignment, which go through 77778 and 77779 respectively
  - ISCO codes have been uploaded, may need some cleaning of tabs, cr/lfs etc. -- also note change in field names
- hr_test_110613.sql with slight modifications - done few minor changes to the dump file.
- HRPJPJPHA.sql -- a query against the DB joining post, job_title, post_history, assignment.
- HRPJPJPHA.sql modified - Query modified accordingly.

Project Spec
- HR Module - Project Spec.docx
- HR Role Privileges.docx

Mockups -- these files are mockup xml
- HRStaff1.xml -- Old tabbed mockup of Manage Staff
- HRJobTitle.xml -- old modal Job Title dialog
- HRPPost1.xml -- Old modal post dialog

Report Queries
- reports.sql

Presentation
- Human Resource Module Presentation

Project Plan

Milestones:

- Implementation of data model and the skeleton of the module
  - Creating required database tables
  - Creating required hibernate mappings and classes
  - Identifying and creating the required services for the module.

- Adding Job structure to the HR Module
  - Creating UI from the Admin page for the Jobs and Managing Job structure with a service
  - Create multiple positions for each job, each with a unique position code, department, supervisor, and office, facility or geographical location
  - Creating HRManager functions:
    - Assign an employee to a position
    - Record employment history and changes in the employee position

- Implementation of HRRReporting Services
  - Create and save customized reports to display aggregate data
  - Display reports as tables or charts
  - Create the following reports:
    - Individual Comprehensive Report (Content of all tabs)
    - Current post/assignment/leave list in alpha, location, cadre/job title order (filterable)
    - Vacant or on-leave posts
    - Posts by location at a time disaggregated by gender/vacant, age (DHIS 2 compatible)

Timeline:

IT1 - May 23 - June 5:
This iteration focuses on the first milestone i.e implementation of the data model and creating the basic HR module structure. This involves the following tasks:

Creating the basic structure of the module - Done
Creating the data model of the project. - Done
Creating the domain objects and hibernate mapping files. - Done

IT2 - June 6 - June 30:
Creating Job structure and related classes (services). - Done
Creating required pages to manage jobs (admin functions) - Done

IT3 - July 1 - July 20:
Creating HR manager functions such as employee positions and employment history. - Done

IT4 - July 21 - August 6:
Implementation of HR Reporting Services to store and display various kinds of reports - Done

IT5 - August 6 - August 15:
Documentation and Testing - Done