

Graduate Program in Information Science and Telecommunications and Networking

School of Information Sciences

University of Pittsburgh

Lab 3: Access Control for EHR

Version 1.2, Last Edited 3/24/2016

Group Members: _____

Date of Experiment: _____

Graduate Program in Information Science and Telecommunications and Networking

School of Information Sciences

University of Pittsburgh

Read the following guidelines before working in the lab

General Guidelines

Through this lab you will work on OpenMRS 2.3.1 Standalone Edition. It's a Java web application that can be installed on Linux, Mac OS X, and Windows. You can work with its demo that contains data for 5,000 sample patients. The demo can be downloaded and installed on your own system, or you can use its online web application. You must have Java 6+ installed on your system to run OpenMRS. Use username: **admin** and password: **Admin123** for logging into the demo.

A written lab report is required for this assignment. You should turn in a printed copy of the lab report. Space has been provided to answer some of the questions of the exercises in this lab. However, you should attach extra sheets of paper with your answers for some of the questions. Please label your extra sheets with your name and indicate precisely which questions you are answering.

Part I: Objective

The objective of the exercises presented here is to familiarize the students with the role based access control features available in OpenMRS, which is an open source EHR system.

Part II: Equipment/Software

The exercises of this lab are based on OpenMRS, which is an open source EHR system and is freely available at <http://openmrs.org/download/>.

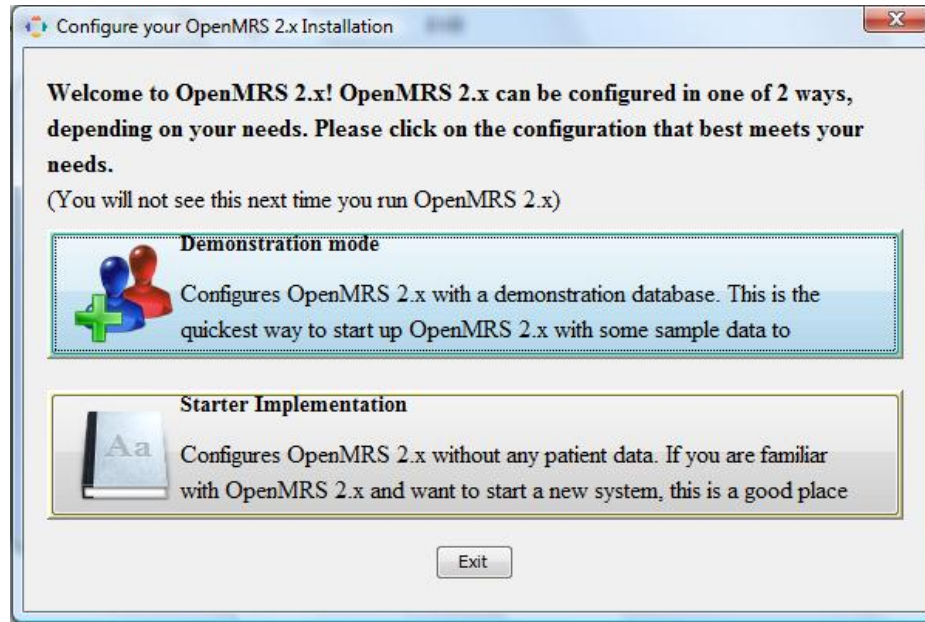
You can download and install the latest version of the demo, which is OpenMRS 2.3.1 Standalone Edition or you can use its online web application that is available at <http://demo.openmrs.org/openmrs/login.htm>. As the online demo is not stable and it's not available occasionally, it's recommended that you install the demo on your computer.

The database of the demo web application is automatically reset periodically, so you can make any changes that you want, but your modifications will be lost during automatic reset.

Find OpenMRS user guide at <https://wiki.openmrs.org/display/docs/User+Guide>.

Part III: Installation

You can download OpenMRS Standalone from the OpenMRS web site <http://openmrs.org/download/>. You must have Java 6+ installed on your system to run OpenMRS. To install the standalone version, download the ZIP file and extract it, then run the openmrs-standalone.jar file. During setup choose an option that installs the demo data, the first option in the following picture.



By default, the initial username and password are as follows: *Username: admin* and *Password: Admin123*.

When OpenMRS is running you can access the application by opening the following URL in your browser:

<http://localhost:8081/openmrs-standalone/>

If you want to stop the application, use the **Stop** button in the user interface. You can restart the application by clicking **Start**, or running the JAR file again.

Part IV: Access control for EHR

OpenMRS is an open-source EMR (electronic medical records) system that has been deployed in many developing countries, including South Africa, Kenya, Rwanda, India, Pakistan, the Philippines, etc.

OpenMRS is a Java web application that follows MVC (Model-View-Controller) model. It uses MySQL as a database engine, and implements **RBAC** as its access control model.

OpenMRS employs privileges and roles to control access to data in the EMR system. A privilege specifies what can or cannot be done in the system (e.g., *Add Patients* or *Delete Notes*). Roles are used to gather privileges into more manageable groups. Roles can inherit privileges from their parent roles.

Roles and Privileges are managed through the **Advanced Administration** page, under the **Manage Users** section.

Exercise 1:

Study different privileges in OpenMRS Demo and list four of them that seems most important to you.

Exercise 2:

Study different roles in OpenMRS Demo and name four main roles that have maximum set of privileges.

First, let's look at a simple example¹:

Assume there are several privileges related to patient data—e.g., *View Patients*, *Edit Patients*, and *Add Patients*. The *View Patients* privilege lets users look at patients in the system, the *Edit Patients* privilege lets users edit information about existing patients, and the *Add Patients* privilege allows users to create a new patient record within the system. Now imagine that you need to assign the proper roles to three people: *Mary* the Medical Student, *Bob* the Data Assistant, and *Erica* the Data Manager. You want medical students to be able to view patients, but not edit or add them. Data assistants should be able to not only view, but also edit patient data. And you want your data managers to be able to create new patients within your system.

In order to give these privileges to the relevant users, you must define a role for each of these types of user.

¹ This example is based on OpenMRS user guide.

Graduate Program in Information Science and Telecommunications and Networking

School of Information Sciences

University of Pittsburgh

Role	Privilege(s)
Medical Student	View Patients
Data Assistant	View Patients Edit Patients
Data Manager	View Patients Edit Patients Add Patients

After defining the main roles for the users of the system, add users *Mary*, *Bob*, and *Erica* and assign each user to its proper role.

Exercise 3:

Define the above mentioned roles and users in OpenMRS and attach the screenshot.

Now, let's take this process one step further. While it may not seem necessary in this simple example, as EMR system grows, it will likely end up with a large number of different roles. Very often, certain roles can be defined as a combination of other roles. In this example, a Data Manager oversees the Data Assistants and therefore should have all of their privileges plus some additional privileges. Now, redesign roles slightly to show how this might work.

Role	Inherit Privilege(s) from Role(s)	Additional Privilege(s)
Medical Student		View Patients
Data Assistant		View Patients Edit Patients
Data Manager	Data Assistant	Add Patients

You can see that the Data Manager role can be more clearly defined as a senior of Data Assistant role with the extra ability to add patients to the system. In addition, if you

Graduate Program in Information Science and Telecommunications and Networking

School of Information Sciences

University of Pittsburgh

should change or enhance the privileges of the Data Assistant role at any time in the future, the Data Manager will automatically adapt to those changes.

In a common deployment scenario, you will have several roles that use the same privileges with only a few differences. It is simpler to manage these privileges by defining a new role from which the others can all inherit. For example, you may have roles like **Clinician**, **Data Assistant**, and **Caregiver** that all have the same rules about viewing patient data. You might benefit from creating a new **Patient Data Viewer** role, assigning it to each of those other roles, and then managing the privileges in one place.

Exercise 4:

Now, you will work on a more complex scenario. Assume following roles are defined in an EHR system. Now, before defining these roles in OpenMRS, try to combine the roles that can inherit privileges from each other. You can define new roles if it makes managing your role hierarchy easier. Report your role hierarchy.

Role	Privilege(s)
Nurse	Add/ Edit/ View Patients Add/ Edit/ View Laboratory Orders Add/ Edit/ View Observations Add/ Edit/ View Reports
Doctor	Add/ Edit/ View Patients Add/ Edit/ View Laboratory Orders Add/ Edit/ View Encounters Add/ Edit/ View Visits
Health Secretary	Add/ Edit/ View Encounters Add/ Edit/ View Diet Orders
Physiotherapist	Add/ Edit/ View Encounters Add/ Edit/ View Physiotherapy Orders
Psychologist	Add/ Edit/ View Encounters

Graduate Program in Information Science and Telecommunications and Networking

School of Information Sciences

University of Pittsburgh

	Add/ Edit/ View Psychology Orders
Radiologist	Add/ Edit/ View Laboratory Orders Add/ Edit/ View Radiology Orders
Dentist	Add/ Edit/ View Encounters Add/ Edit/ View Diet Orders
Ambulance Personnel	View Patients Add Reports
System Administrator	Add/ Edit/ View Users Add/ Edit User Passwords Add/ Edit/ View Roles
Registration Clerk	Add/ Edit/ View Users Add/ Edit/ View Patients Add/ Edit/ View Appointments

Exercise 5:

Define your role hierarchy in OpenMRS. (You can add a new privilege if it's not defined in OpenMRS). Attach the screen shot of your defined roles in OpenMRS.

Built-in roles: There are some predefined roles in OpenMRS such as *Anonymous*, *Authenticated*, and *System Developer*.

Exercise 6:

Study the OpenMRS User Guide about these predefined roles and their privileges and report their usage. Create different users with these roles and check what the differences between their access lists are.

Exercise 7:

What are the challenges you encounter while working with access control model in OpenMRS? What are your suggestions for improving the access control model of OpenMRS?

Study HL7 Role-Based Access Control (RBAC) Engineering Process ([http://csrc.nist.gov/groups/SNS/rbac/documents/hl7_role-based_access_control_\(rbac\).pdf](http://csrc.nist.gov/groups/SNS/rbac/documents/hl7_role-based_access_control_(rbac).pdf)).

Two types of roles are defined in the document: **Functional Roles** and **Structural Roles**.

Exercise 8:

What are the differences between these two roles?

Exercise 9:

Define two Functional Roles and two Structural Roles in a healthcare environment and assign three proper privileges to each of them. Report your roles and their privileges.

Exercise 10:

Define your roles and their privileges in OpenMRS. Attach the screen shot.