PHR Module

Overview

The PHR module (i.e. personalhr) allows you to create a patient controlled health records application. It gives the patient the full control of his/her own health records and other personal information, and enables him/her to share part or all of those information to any one in his/her social network such as a family member, a doctor, or any other caregiver he/she trusts.

Architecture

The PHR module defines the following (static) OpenMRS roles / privileges:

- PHR Patient / PHR Single Patient Access
- PHR Restricted User / PHR Restricted Patient Access
- PHR Administrator / PHR All Patients Access

, and is composed of the following functional components:

1. PHR Patient Registration
2. PHR Invited Self Registration
3. PHR Relationship & Sharing
4. PHR Authentication & Authorization

PHR module provides basic user interfaces such as PHR Patient Dashboard, PHR Restricted User Dashboard, and PHR Administrator Dashboard, which can be extended by other modules to create a fully blown PHR application.

Sharing Process

- When Patient adds a new relationship by entering name, relationship type, email address and sharing type and submitting, System creates a new record in phr_sharing_token table
- System sends an email to the authorized person containing a link to OpenMRS PHR Login window and the sharing token string
- When the targeted Person receives the invitation and clicks the URL with Sharing Token embedded to access the information shared by the Patient
  - System pops up a login window (embedded with Sharing Token)
- When the targeted Person (Sharee) clicks Sign Up button on the login window
  - System pops up a self registration window not pre-populated with Patient entered info (still embedded with Sharing Token)
  - The Person entered his personal information that can be different from Patient entered information and provides username and password to be created
  - System creates a new OpenMRS Person object and a new OpenMRS User account with PHR Restricted User role
  - The Sharing Token record is updated with the proper share_activate_date, and relation_person_id
  - Generate warning if user entered name or email address do not match those stored in the phr_sharing_token table
- When the targeted Person (Sharee) clicks Login button after supplying his username and password created before on the login window
  - The Sharing Token embedded (if there is one) is checked against phr_sharing_table and corresponding record is updated if and only if relation_person_id is null.
- When the targeted Person clicks the Person link in his relationship list after logging in
  - To display the relationship list, System looks up the phr_sharing_token table with the current user's Person ID to find all of the patients he has relationship with
  - After clicking the Patient link in his relationship list, System checks if the current user has a Data sharing Relationship with that clicked Patient by looking up the Relationship table with the clicked person's Person ID and the current user's Person ID and all of its matched Person ID's found in the person_match table

PHR Security

PHR authentication and authorization is implemented by PHR security checking at URL Level, Page Level, Controller Level, and Service Level. The Pro & Con of various levels of security are listed as follows:

<table>
<thead>
<tr>
<th>PRO</th>
<th>CON</th>
<th>Detail</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page only security</td>
<td>1. No Java code change is needed to bring in new modules into PHR application</td>
<td>1. JSP page change is needed to bring in new modules into PHR application 2. Unable to check privilege on direct DWR or Web Service calls</td>
<td>1. Add <a href="">personalhr:require</a> and <a href="">personalhr:hasPrivilege</a> tags in each jsp page 2. Map privilege text (registered or not) to PHR dynamic role(s)</td>
</tr>
<tr>
<td>URL only security</td>
<td>1. No Java code change is needed to bring in new modules into PHR application</td>
<td>1. URL privilege mapping is needed for every page and DWR call 2. Data identification not always possible (e.g. if naming convention not followed) 3. Unable to check the privilege on embedded JSP script 4. Very hard to check the privilege on a POST request, DWR, or Web Service call</td>
<td>1. Check if URL is allowed for given user in URL filter 2. Check if data (as identified by patientId, personalId, encounterId, etc) is allowed for given user in URL filter 3. Map URL to PHR dynamic role(s) 4. DWR filter is needed to check if DWR call is allowed</td>
</tr>
</tbody>
</table>

Disclaimer

This module is still under construction. This information may not be accurate
<table>
<thead>
<tr>
<th>Controller only security</th>
<th>Service only security</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. No need to hold temporary privileges for a prolonged time</td>
<td>1. Protect the right data at the right time</td>
</tr>
<tr>
<td>1. Java code change is needed to bring in new modules into PHR application including controllers, servlet, dwr, etc.</td>
<td>1. Java code change is needed to bring in new modules into PHR application and to secure existing base OpenMRS services</td>
</tr>
<tr>
<td>1. Non-PHR pages are sufficiently protected by (&lt;)openmrs:require&gt; and (&lt;)openmrs:hasPrivilege&gt; tags</td>
<td>1. Non-PHR pages are sufficiently protected by (&lt;)openmrs:require&gt; and (&lt;)openmrs:hasPrivilege&gt; tags</td>
</tr>
<tr>
<td>2. PHR users do not have permanent openmrs privileges</td>
<td>2. PHR users do not have permanent openmrs privileges</td>
</tr>
<tr>
<td>3. PHR users are given temporary openmrs privileges when making openmrs requests from PHR compliant functions and give away those privileges when finishing the requests</td>
<td>3. PHR users are given temporary openmrs privileges once logged in</td>
</tr>
</tbody>
</table>

Instructions for installation and configuration of basic PHR modules (Personal Health Toolkit)

1. Install OpenMRS 1.8.3: [https://svn.openmrs.org/openmrs/tags/1.8.3](https://svn.openmrs.org/openmrs/tags/1.8.3)
2. Check out and build the general messaging module from: [https://svn.openmrs.org/openmrs-modules/messaging/branches/messaging4all](https://svn.openmrs.org/openmrs-modules/messaging/branches/messaging4all)
3. Check out and build the phr (personalhr) module from: [https://svn.openmrs.org/openmrs-modules/personalhr/branches/0.0.3](https://svn.openmrs.org/openmrs-modules/personalhr/branches/0.0.3)
4. Check out and build the messagingphr module from: [https://svn.openmrs.org/openmrs-modules/messagingphr](https://svn.openmrs.org/openmrs-modules/messagingphr)
5. Check out and build the messaging module from: [https://svn.openmrs.org/openmrs-modules/phrjournal/trunk](https://svn.openmrs.org/openmrs-modules/phrjournal/trunk)
6. Login as openmrs administrator, load the messaging, personalhr, messagingphr, and phrjournal modules into openmrs in sequence
7. From the messaging module’s Manage Messaging Gateway administrative page, Start Omail server; Configure Outgoing Email Server and Start the Gateway
8. From the messaging module’s Manage PHR Security Rule and Manage PHR Authorized URL pages, make sure security rules and URL’s are loaded
9. Create a PHR admin user with PHR Administrator role
10. Create an internal user with PHR Restricted User role (Used for self-registration purpose, must have username: Temporary, password: Temporary8)
11. Add the following custom person attribute: Name: Email, class: java.lang.String, description: A person’s email address
12. Login as a PHR admin user, create a demo PHR Patient user using the Manage Patients/Add Patient command and the Manage Users/Add User command
13. Login as a PHR Patient user, you’ll see three tabs in the patient’s dashboard: My Relationships, My Email, My Journal
14. As a PHR patient, you can add a relationship with a valid email address. An invitation will be sent to the provided email address.
15. The recipient of the invitation can click a URL link embedded in the email to self-register with openmrs as a PHR Restricted user.
16. After logging in as a PHR Restricted user, you’ll see in the dashboard My Relationships tab, where the patient who sent you the invitation is listed and whose health information and journal entries may be available depending on the Data Sharing Type granted to you; and My Email tab, where you can send openmrs internal email (OMail) to that patient.
17. You can extend PHR Patient dashboard with additional tabs (e.g. My Encounters). Be sure to configure/authorize the new URLs using the PHR Authorized URL form, and guard the jsp pages with \(<\)personalhr:require\> or \(<\)personalhr:hasPrivilege\> tags.
18. The default data sharing types include “Sharing Medical”, “Sharing Journal” and “Sharing All”. Additional sharing types are configurable by adding them to the PHR Security Rule form.
19. Please Note: at this point of time, the installation of messagingphr module and phrjournal module may not be stable due to a recent migration from openmrs 1.7.x to 1.8.3. The PHR (the messaging and the personalhr modules) will still function without these two modules though.

Frequently Asked Questions:
1. In the current PHR code, for the "restricted users" what is the process of obtaining a userid/password?
   Answer: Patient adds a relationship with a person -> An invitation is sent to that person's email address -> The person clicks a link embedded in the email invitation (with an embedded sharing token, which is similar to a "unique id") to accept the invitation -> The person sees PHR login page and clicks the "First Time User Registration" link -> The person enters basic information to create a PHR restricted user account, which has a link to the patient through the sharing token he received.

2. If the "restricted user" is a physician whose email is drtanmaymahapatra@yahoo.com and two patients share his/her medical record whether the physician need to get two separate userid or one userid will work?
   Answer: the physician only needs to do the registration once. The second time he received an invitation, he can simply clicks the link and login to OpenMRS with his previously created user account, and a link to the new patient will automatically be added to his account.

3. It will be ideal if the physician use one single id. If the physician use one single id then he/she should be able to use "find patient" functionality?
   Answer: He will be able to "find" all patients who he has a relationship with, as those patients will all be listed under his relationship tab.

4. Currently, in openmrs (@ sign and as such emailid) is not allowed as userid. There is a ticket to fix this in core OpenMRS. We will surely like to fix this or at-least make it a part of this module, so that email id is allowed as user id.
   Answer: Good point.

5. I am assuming it is possible to make a omod file for this module and this module can be imported in an existing OpenMRS implementation. Please confirm.
   Answer: Yes.

6. When the phr omod file is imported in an existing OpenMRS implementation what will be the login URL for the patients?
   Answer: There are two login pages, one for base openmrs users (/openmrs/index.htm), the other for PHR users (/openmrs/phr/index.htm). You can login as a PHR user or non-PHR user through either login page, the program will direct you to the correct pages based on your role.

7. Current login URL for the Admin and Physicians are: http://117.194.129.238:8080/openmrs
   This one already have a link of "I forgot my password". Whether it will be the same urls, with the additional links of "First Time registration to request an unique id" and "Create a userid if you have an unique id"?
   OR
   will it be a separate URL like: http://117.194.129.238:8080/openmrs/phr/ with the three links as "I forgot my password", "First Time registration to request an unique id" and "Create a userid if you have an unique id"
   Answer: The two extra links ("First Time registration to request an unique id" and "Create a userid if you have an unique id") should be added to the /openmrs/phr/ page.