caCORE SDK CSM/UPT Enhancement Tasks

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **LOE** | **Impact** | **Tracker** | **Feature Request** | **Comments** | **Presented to TRB** | **TRB Decision** | **TRB Comments** | **Expected completion date** |
| 1 | Configurable UPT | 20 | High | [CSM-692](https://tracker.nci.nih.gov/browse/CSM-692) |  | Configurable ability to prevent access to specific screens (and therefore functionality) based on the application and user privileges  Major issue for caArray since group creation through UPT corrupts the installation  Related trackers: [CSM-336](https://tracker.nci.nih.gov/browse/CSM-336), [CSM-378](https://tracker.nci.nih.gov/browse/CSM-378), [CSM-692](https://tracker.nci.nih.gov/browse/CSM-692)  Related trackers submitted by Rashmi Srinivasa (caArray) on 2011-09-26, on behalf of caArray support on 2011-01-12, and by Steve Stowers on 2006-05-01.  CSM-378 previously approved by TRB as GF 30433  Reusable page components, branding, embedding, etc.  See attachment below |  |  |  |  |
| 2 | Support for compliance with NIH security-related policies--primarily NIH Password Policy  <http://ocio.nih.gov/nihsecurity/pwd_policy.pdf> | 20 | High |  |  | Ability to change password, require change on first use, scriptable character combination checks, configurable lockouts, etc.  Support deactivating (and reactivating) users instead of deleting them.  Encrypt Personally Identifiable Information in database  Related trackers: [CSM-651](https://tracker.nci.nih.gov/browse/CSM-651), [CSM-409](https://tracker.nci.nih.gov/browse/CSM-409), [CSM-380](https://tracker.nci.nih.gov/browse/CSM-380), [CSM-418](https://tracker.nci.nih.gov/browse/CSM-418), [CSM-345](https://tracker.nci.nih.gov/browse/CSM-345)  Related trackers submitted by Zhong Li (MAT KC) 2009-06-28, unknown 2009-07-27 (caArray), caNanoLab, and on behalf of Jerry Eads (caArray) on 2006-09-28 |  |  |  |  |
| 3 | Need stronger access control | 10 | Medium | [CSM-360](https://tracker.nci.nih.gov/browse/CSM-360) |  | Security flaw allowing unauthorized access to records  See attachment  Reported by Mark Grand (caGrid KC) on 2009-11-24 |  |  |  |  |
| 4 | UPT GUI installer should validate database connection info | 3 | Medium | [CSM-368](https://tracker.nci.nih.gov/browse/CSM-368) |  | Installation improvement suggested by caArray KC based on CDL install  There is no validation of the database connection information. UPT must be completely installed and run before a user finds out that the dbconnection info is invalid.    Posted 2009-11-19 |  |  |  |  |
| 5 | Section 508 compliance improvements | 5 + documentation team | Medium |  |  | While the score from the Accenture Digital Diagnostics Engine scan of hosted UPT is 95%, it doesn’t look like the scan was thorough. Additionally, peripherals (such as the installation GUI and scripts) need to be reviewed and the manual converted to the wiki.  A review using the WAVE Firefox plugin showed hundreds of errors over 24 screens.  Errors look easy to correct and primarily fall into these four categories:  • Missing image alternative text  • Menu Spacer Images missing alternative text  • Orphaned Form Labels: A form label is present, but is not associated with any form input field  • Form Labels missing: A form <input>, <select>, or <textarea> label is missing |  |  |  |  |
| 6 | Enterprise authorization service (suite security) | 5 | Medium |  |  | Ability to provide HTTP based centralized access to CSM provisioned security.  Could be used for CTS centralized security (not discussed with CTS). |  |  |  |  |
| 7 | Password recovery | 5 |  | [CSM-331](https://tracker.nci.nih.gov/browse/CSM-331) |  | Initial request was for retrieval of superadmin password when using RDBMS authentication. (Password is encrypted. If there are no other superadmins, it cannot be reset; although jamming in a known encrypted password might work.)  Could be done via email or using security questions. (Clear-text email would violate security policy.)  Suport for this feature should be configurable by application.  Submitted by xuegong wang of UPMC on 2006-02-22  Related trackers: [CSM-408](https://tracker.nci.nih.gov/browse/CSM-408) |  |  |  |  |
| 8 | Auto-email new user or associated user | 4 |  | [CSM-346](https://tracker.nci.nih.gov/browse/CSM-346) |  | When adding or modifying a user, consider adding a checkbox control that, if checked, would automatically send an email to the user containing the user's login information.  (Use without encryption would violate password policy.)  Submitted on behalf of Stephen Reckford (C3PR) on 2006-09-28 |  |  |  |  |
| 9 | Support datasources for RDBMS authorization (instead of requiring login information directly) | 3 |  | [CSM-602](https://tracker.nci.nih.gov/browse/CSM-602) |  | See attachment  Requested by caTissue on 2009-07-15 |  |  |  |  |
| 10 | Better LDAP integration when adding users to UPT | 5 |  | [CSM-381](https://tracker.nci.nih.gov/browse/CSM-381) |  | Scripted creation of users in CSM that will authenticate using existing LDAP accounts.  Suggested by caArray KC posted 2011-01-12 |  |  |  |  |
| 11 | Implement paging for UPT (result sets) | 10 |  | [CSM-387](https://tracker.nci.nih.gov/browse/CSM-387) |  | For every search result page that returns a lot of results, allow the user to view the result set one page at a time. This includes next and previous options and a display of the current (active) page and total number of pages.  Not sure of original reporter. 20-Nov-2008  Related tracker: [CSM-339](https://tracker.nci.nih.gov/browse/CSM-339)  Related request by Steve Stowers on 2006-05-01 |  |  |  |  |
| 12 | Documentation improvements | 5 |  | [CSM-382](https://tracker.nci.nih.gov/browse/CSM-382) |  | No specifics: “The UPT documentation can be improved with better organization and illustration by examples”  Suggested by caArray KC posted 2011-01-12 |  |  |  |  |
| 13 | User-friendly UPT | 15 |  |  |  | Free text style searches  Improved look and feel  Fine-grained even-driven GUI  Improved error messages  Improved validations  See attachment  Related trackers: [CSM-379](https://tracker.nci.nih.gov/browse/CSM-379)  Reported by caArray KC on behalf of MAT KC posted 2011-01-12 |  |  |  |  |

**Attachments:**

**Item 1, Configurable UPT**

**Issue details:**

caArray has many complaints about CSM UPT protection group creation page. caArray has extended CSM ProtectionGroup to provide group level security permissions in its application. caArray manages groups through its web application and it synchronizes and group creation with CSM through CSM API. There are synchronization issues when an UPT admin creates protection group through UPT application. caArray want to disable protection group feature of certain administrators.

**Issue background:**

UPT has been a standalone web application with security provisioning capabilities using CSM API. Any applications implementing CSM API would either need to use UPT or develop their custom UI within their application to provide security provisioning capabilities.

For example, UPT provides capability to add and update a user into CSM RDBMS schema. If an application implementing CSM API would want to have same capability to add and update a user with in their application, it would need to develop those screens from scratch all again.

**Issue resolution:**

UPT would need to re-architect its web application to support customizable, extendable and flexible UI, actions, event handling mechanism. An application should be able include UPT web application components into its application and make use of UPT functionalities.

Clean separation of business logic with presentation

Ease the burden of developing and maintaining applications that run using UPT

Ease of UI state management, event handling, input validation, page navigation and accessibility.

Facilitate customizable branding with UPT pages

Support customizable and extendable actions and user interface to include within an existing web application

**Fix impact:**

Entire UI, navigation, messages

**Tasks:**

CSM API:

1. Update schema to introduce new table(s) for feature level permissions
2. Provide APIs to access, create, update, delete feature level permissions
3. Provide APIs to access, create, update, delete feature functional level permissions
4. Externalize error/warning messages to be customizable
5. Update APIs to be extendable.
6. Update/add Junit test cases.

CSM UPT:

1. Update UPT to use CSM APIs and enforce feature level security
2. Introduce new page(s) to manage feature level permissions
3. Introduce new page(s) to manage feature functional level permissions. For example, Protection Group page may not have delete permission for a particular admin user.
4. Display or hide UPT features based on permissions
5. Display or hide UPT features functions based on permissions
6. Provide ability to brand CSM pages with custom images and style sheets.
7. Design and implement a workflow for an application to include UPT pages. For example, an application with user creation page should be able to include UPT user creation page to take advantage of it.

## Item2, Support compliance with NIH Password Policy

**Issue details:**

Password security policies help ensuring the confidentiality, integrity, and availability of NIH information by minimizing the vulnerability of NIH information technology (IT) systems. Passwords are the first line of defense against potential intruders. A strong password policy prevents hackers from gaining unauthorized access into a system and any resources available to an authenticated user. CSM has to follow NIH password policy and provide APIs to let applications use or extend them. CSM UPT would need to implement these policies using CSM APIs.

**Issue resolution:**

CSM is an open source application sponsored by NCI CBIIT, a division of NIH/HHS. CSM has been in use within NCI community and outside as well. Being part of NIH, CSM has to follow NIH password policy to ensure security. NIH password policy can be broken down into following sections:

Password definition: This addresses min/max length of password and allowed characters and their minimum combination.

Password usage: this addresses changing passwords, minimum password age, account lockout and its duration

Password Administration: this addresses password policy for administrators, activating or inactivating a user, communicating passwords, password auditing

CSM will provide a configurable and extendable solution to implement NIH security policy. These policies should be turned on or off through CSM configuration. Applications following different security policies should be able to easily customize or extend default CSM implementation. These changes will be implemented with CSM and will be used by UPT to enforce them. This would require schema changes within CSM. These changes would be applicable to all applications configured with a particular CSM instance.

**Issue impact:**

## CSM API, CSM UPT, All applications using CSM security

**Tasks:**

CSM API:

1. Provide configurable way to enable or disable policy settings through properties file.
2. Validate password main/max length, allowed characters, combination of characters and display appropriate error messages.
3. Update login module to enforce password expiration rules.
4. Encrypt CSM PII information storing into the database
5. Provide ability to active or inactive a user. Remove delete capability.
6. Update/add Junit test cases.

CSM UPT:

1. Update UPT to use updated CSM APIs to its login page. Display message about password expiration.
2. Provide ability to update password during expiration window.
3. Introduce a new page to change password on first time use. Enforce policies to not use previously used passwords.
4. Implement user activation or inactivation feature. Remove user delete feature.

## Item 3, Need stronger access control

[CSM-360](https://tracker.nci.nih.gov/browse/CSM-360)

**Issue details:**

Queries can probe for the existence of records containing specific attribute values in records that a user is not authorized to access. Consider this scenario:

Suppose that the OpenClinica data service has two users named X and Y. Among the things that OpenClinica tracks are studies and subjects, which have a many-to-many relationship to each other.

User X is supposed to have access to study A and all subjects associated with study A. User Y is supposed to have access to study B and all subjects associated with study B. We do not want user X to be able to tell which subjects in study A are also in study B. Similarly, we do not want user Y to be able to tell which subjects in study B are also in study A. However, CSM does not allow us to prevent this.

Here is how it works. For the Study class we set up a filter that will only match a specified set of study names. For the Subject class, we set up a filter that will only match Subjects that are associated with a Study that matches a specified set of study names. For user X we specify that the study name to filter for is A. For user Y we specify that the study name to filter for is B.

The problem is that when we issue a query, CSM does not use both filters. It just uses the filter that matches the class that we want to return an instance of. If user X issues a query asking for all of the subjects that are associated with study B, CMS only uses the filter for Subject. It ignored the filter for Study. For this query, there will be no filtering on Study, so the Subject filter will remove subjects that are not associated with study A and the query is free to find subjects among those that are also associated with study B.

**Issue resolution:**

At an abstract level, the difficulty we have with CSM is that it only applies those constraints to a query that are associated with the type of object that the query will return. If CSM were to apply all constraints related to all objects referenced in an hql query, then it would meet our use case's needs.

This difficulty stems from the fact that CSM filters are implemented as hibernate filters and hibernate does not manage filters in a way that matches our needs. If CSM were to implement filters as hibernate Criterion objects, then it would be possible for CSM to explicitly manage the filters.

The way that I would expect this would be packaged is as an extension to the existing API. To use it, the application would create its query as a hibernate Criteria object. The application would then pass the Criteria object to one of two new methods in the CSM API that might be called addUserCriteria or addGroupCriteria. These methods would add to the given Criteria hibernate Criterion objects that encapsulate the relevant filters. The application will then ask hibernate to execute the Criteria query.

Hibernate will execute the query applying all of the filters at once. The result of such queries will be consistent with our use cases.

In addition to this extension to the CSM API, it will also be necessary to modify the caCORE SDK to use the new API extension.

As a work-around for use in caGRID data services for the previously described problem, I have written a CSM-aware replacement (attached) for the default CQL processor used in caGRID data services. This replacement CQL processor directly interprets information in CSM tables by applying filters for all tables referenced in a query, not just the target table.

Follow the recommendation given in the issue details and refer to the example attached in GForge.

**Fix impact:**

Instance level security

**Tasks**:

CSM API:

1. Update CSM API to look at all query objects and apply filters in a particular query.
2. Provide HQL Criteria API to extend flexibility in using CSM instance level security
3. Update/add Junit test cases

**Item 4, UPT GUI installer should validate database connection info**

**Issue details:**

**Tasks:**

1. Update GUI installer to provide test database connection button.
2. Update GUI installer to provide validation message.

**Item 6, Enterprise authorization service (suite security)**

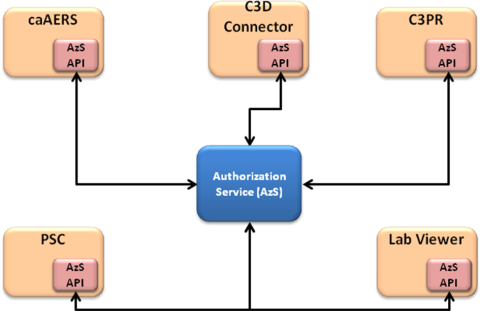
**Issue details:**

Applications query provisioned security using CSM API’s. This creates a tight coupling between the application and CSM. Any changes in CSM API’s would force applications to update their software. This is especially troublesome with multiple applications (suite-wide) trying to implement a centralized authorization policy using CSM.

**Issue resolution:**

An enterprise authorization service would solve the tight coupling between CSM and implementing applications. Application would make a request to this secured authorization service to verify permission for necessary security elements. This enterprise service would be a façade to multiple complex CSM APIs. Any version or API changes would be abstracted by this service so that applications depend on CSM would have minimum or no impact with changes. This enterprise authorization service would be based on HTTP request/response model, stateless and leverages HTTP security implementation approaches. A standard request and response model will be defined as part of this service creation.

Following is an example use case.



## Tasks:

1. Provide standardized input and output formats
2. Provide RESTful service to expose authorization security functions. This would support GET only.
3. Provide ability to decrypt HTTP header to read user/password

**Item 9, Support datasources for RDBMS authorization**

## Authentication LoginModule configuration should support DataSource

[CSM-602](https://tracker.nci.nih.gov/browse/CSM-602)

**Issue details:**

RDBMS login module should support authentication using a configured data source. Currently, RDBMS login module requires database connection settings to perform authentication.

**Issue background:**

The CSM authentication service provides a simple and comprehensive solution for user authentication. Developers can easily incorporate the service into their applications with simple configuration and coding changes to their applications. Currently, the authentication service allows authentication using LDAP and RDBMS credential providers only.

RDBMS login mechanism uses JAAS based configuration for authentication.

As of v4.2, following properties are required to establish a connection to the database includes:

* Driver - The database driver loaded in memory to perform database operations.
* URL - The URL used to locate and connect to the database.
* User - The user name used to connect to the database.
* Password - The password used to connect to the database.

The following property provides the database query to be used to retrieve the user:

* Query - The query that will be fired against the RDBMS tables to verify the user ID and the password passed for authentication.

**Issue resolution:**

RDBMS login code will need to be changed to support Data Source. It should continue supporting database connection settings as well.

gov.nih.nci.security.authentication.loginmodules.RDBMSLoginModule

**Fix Impact:**

* UPT login using RDBMS login with data source settings
* UPT login using RDBMS login with database connection settings
* Unit test APIs

**Tasks:**

1. Update configuration files to accept datasource.
2. Update CSM API to access configured datasource.

## Item 13, User friendly UPT

**Issue details:**

Free text style searches

Improved look and feel

Fine-grained even-driven GUI

Improved error messages

Improved validations

**Issue background:**

UPT web interface was designed and developed 5 years with latest technologies available at that time. There has been number of technological advancements in recent years helping to enhance user experience with web applications.

**Issue resolution:**

UPT will make use of latest technical advancements to enhance user experience with look & feel, navigation, error messages, validations, notifications etc. throughout the application.

**Fix Impact:**

Entire UI, navigation, messages

Tasks:

1. Update UPT to provide free text style search boxes in all searchable screens. This would include search user, application, and protection element.
2. Update UPT to use latest technologies supporting dynamic, event driven and asynchronous application
3. Update UPT to format, validate and display user centric messages.
4. Update UPT to use externalized messages
5. Update UPT to improve its navigation between pages.