

Grouper UI redesign v2.2

Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources
---------------------------	---	--------------------------------	--	---	--

[See the documentation of the Grouper 2.2 UI, released July 2014](#)

- [Grouper UI csrf xsrf prevention](#)
- [Grouper UI favorites and preferences user data](#)
- [Grouper UI redesign v2.2 issues for Michael Girgis](#)
- [Grouper UI text proposal](#)
- [Grouper UI v2.2 dojo](#)
- [Grouper UI v2.2 error handling](#)
- [Grouper UI v2.2 linking to and from the admin and lite UI](#)
- [Grouper UI v2.2 tasks](#)
- [Grouper UI v2.2 testing](#)
- [Grouper UI v2.3 developers guide](#)
- [New Features \(to be added to UserVoice forum\)](#)
- [Sample UI Redesign Process](#)
- [UI Requirements](#)
- [Weekly UI Chat - 1.2.2013](#)

[Go to Community Requests Table](#)

[Requirements page](#)

Goals

- Optimize the user experience across all browsers and mobile devices.
- Design an interface which meets the needs of Basic and Intermediate users and which is intuitive enough such that no formal training is required to use the Grouper UI for common tasks.
- Meet WCAG 2.0 AA requirements for accessibility.
- Replace the Lite UI and ship v2.2 with the existing Admin UI and the new UI.

Measurable Objectives

- Users with no formal training are able to complete a series of 5-10 tasks within ten minutes.
- Users rate their experience at five or higher on a seven point scale.
- A user using assistive technology is able to perform any and all tasks.

Phases/Process

- Initial research
 - User interviews
 - What tasks do users perform most often? Do all users follow the same workflow to perform those tasks?
 - What are the pain points in the existing interface? What are the most frustrating tasks?
 - What do users wish they could do with the current interface but are unable to?
 - Where/when/how do users interact with Grouper?
 - How much functionality needs to be supported on small screens?
 - What's the true breakdown of expert vs intermediate users?
 - How much time, if any, should be spent testing the current interface against new users? existing users?
- Gather requirements (concurrent with Initial Research)
 - Identify core functions
 - Identify existing functionality to be preserved
 - Identify new features requested by the community
 - Vote on new features?
 - Finalize list of new features and freeze requests
- Low-fidelity wireframes - desktop and mobile
 - Initial sketching and low-fi electronic wireframe development
 - Test common tasks against wireframes
 - Rinse and repeat
- High fidelity wireframes - desktop and mobile
 - Develop richer wireframes
 - Test common tasks against wireframes
 - Report results
- Proof of concept of accessible widgets and write framework code (custom tags, common javascript etc)
 - Some of this could be done concurrently with steps outlined above
 - autocomplete
 - tree control
 - popup window
 - menu
 - layout (frame-like)
 - tooltips

- Graphic Design
- Development (iterative) (see the [Developers Guide](#))
- Testing

Key Tasks for User Testing

- Find a group and add/delete members
- Add a new administrator to an existing group
- Update an existing member's privileges to include ADMIN
- Create a new folder
- Create a new group
- Edit attributes for an existing group
- Add a group as a member of another group
- View the default privileges for a group
- INTERMEDIATE/ADVANCED: Create a composite group
- ADVANCED: Create and edit attributes for an entity
- ADVANCED: Create a role, a permission, actions, and assign permissions

Architecture

- In order to reuse work done in the existing Lite UI, the same architecture will be used: Javascript framework on top of jquery/dhtmlx. There is no custom javascript per function, just server side java ajax which controls the screen via a Javascript layer
- The main screen design will be centered around a tree control on the left pane which will browse the repository, sort of like Windows Explorer or Eclipse or other similar software
- We could have two UI paths in the same UI application: one which is a rich application which might not be accessible, and one which is a very light application which will be accessible and mobile friendly. The first page of the application should be a splash page which is accessible and mobile friendly where the user can set a preference about which UI they want associated with their account or which to use for this session. Browser detection could help here too. The accessible/mobile part could be purely based on Grouper WS-like operations...
- Favorites or recent activity could be associated with the user in the database (e.g. with membership attributes) so that it is easy for the user to setup tasks so they can quickly do their work the next time they use the Grouper UI

Guidelines

Usability (from Initial Research)

- Provide context for where you are and what you are doing, whether it is browsing or editing. Be consistent with navigation elements (especially breadcrumb).
- Question defaults in every step make the user aware of the default settings.
- Provide more data by default and let users filter down from there (better list management).
- Always provide a way out (cancel and back buttons).
- Upon completion of a task, provide confirmation of success.
- Hide/displace expert clutter.
- Use search consistently.

Technical

- Reduce the number of queries for actions
- Make sure most of the logic is in the API, not the UI code. Would be nice if the gaps between what is used and the WS are noted
- Have the UI be usable by keyboard without requiring the mouse
 - Note, could use accesskey for buttons (probably need to add this to the UI framework)
- Do not keep stuff in session, just cache globally, and use request. The only things in session would be things that can be cached for the user but figured out if needed. e.g. authentication information. The app should be able to be used in a load balanced environment with session clustering. No unserializable stuff in session
 - Make sure app works in different tabs in same browser (it will since no data in session)
- The tree control on the left of the UI prevents the URL from changing with clicks, however, we can put anchor URLs periodically as the user uses the app, so that bookmarks, back, forward, etc still work. This is already somewhat in the UI framework, need to revisit

Security

- All methods should be POST, though if GET is required, have a whitelist
- Prevent CSRF by having a key (SESSIONID?) which is transmitted with each request in a form variable (will this work for dhtmlx GET requests?). Have a switch that turns this off

Ideas

- Overall search screen should allow search for all grouper objects
- Comboboxes should have filters (e.g. for which source)
- Comboboxes should show results with a <space> if there arent too many of them (maybe also if there is invalid text?) See the action permission combobox
- [Have recently used objects available](#)
- Have a screen with each type of UI widget it in it for browser testing
- Hide dangerous operations (e.g. delete object) behind menus so they are less likely to be accidentally pressed
- Might need more custom tags to help display objects easier. for instance a subject display custom tag...
- All searches should be case-insensitive (e.g. group searches, attribute searches, group searches in subject results), should search name, display name, and description, alias, etc

- Maybe if there arent too many options in dropdowns, it could be a select instead of combo
- Get query count for each operation and make sure it is minimal :)

Help framework

- Investigate using this: <http://embedded-help.net/>
- Perhaps could use some youtube movies too...

Requirements

- See [UI Requirements](#)

Sample UIs

These sample UIs may still be in development and were submitted as examples when soliciting community requests. This is not a comprehensive list.

- University of Washington's [prototype Groups Service UI](#). This UI was demonstrated at the Grouper working group session at 2012 SMM in Arlington, VA. It was agreed that this prototype UI has a clean look and is easy to navigate.
- [SURFteams UI](#) - Includes examples of handling invitations to join a team (using federated email addresses), creating and deleting teams, adding and deleting members, managing team roles (member, manager, admin), and requesting access.
- [Uppsala University](#)

Community requests

We ask community members to add their recommendations, requirements, or other functional, technical, process, or participation requests for the v2.2. UI to the following table. If you have more to say than fits into a Description cell, please create a child page and put a link to that in the Description cell.

If you see an existing entry that reflects a concern or request that you share, please add your name and/or institution name to the Requester(s) cell.

Request name	Description	Category	Requester (s)
Use Spring MVC and Web-Flow	Use same underlying technologies as Jasig CAS, thereby reducing skill sets necessary to deploy and customize a CAS-protected grouper installation.	Backend	U Montreal
HTML customization	Use CSS themes (perhaps along the lines of csszengarden) and customizable JSP header and footer includes.	Markup	U Penn Cru (CCCI)
Internationalization (i18n)	One language could be globally configured. Supporting multiple language would be better, but it's not a top priority. It could be detected from the browser (which is nice, but a default value would also work) but it has to be user-changeable using a drop-box or link of some kind since browser language might not be what the user prefers to see.	Backend	U Montreal Cru (CCCI)
Display group relationships	Visual display of groups impacting membership of a given group. Would illustrate subgroups and composite factors, perhaps expanded by levels.	UI	U Montreal
Images of people in person picker	We would like a future Grouper UI to be able to display an image/photo of a subject (person) when choosing subjects to add as members of a group.	UI	LIGO
Why isn't subject a member search /display	Would be nice to have a way to search a group's membership and display why a particular subject is not a member of a group. Example: given group x:y:allowed I'd lookup a subject and the UI would display that the subject is not a member of the group because of membership in x:y:deny or the subject is not a member of any groups which make up :allowed. This would be very helpful for debugging a particular subject's membership issues when there are a lot of composite groups in play affecting the final group in question.	UI	U Chicago
Roles /Permissions Area (Lite UI)	Roles/Permissions work flow is somewhat complex and counter intuitive. The hierarchy graph and role/permissions display widgets are cool, but data entry flow could be improved. Maybe better documentation would be helpful. Are there any plans for this?	UI	UCLA
Action Timer	Would be great when viewing details of a group to see when changes to a group will next be provisioned to downstream systems (or updated from external source)	UI/Backend	U Chicago
Mobile-Aware	As mentioned in the architecture section above as a possibility, an interface optimized for use with mobile devices (smartphones and tablets)	UI	Stanford Cru (CCCI)
Clone	Astem:AGroup exists including attributes and values, want to clone it to Bstem:BGroup with attribute names changing "namespace" from A to B. Users often want a new group that is "just like this existing group but different".	UI/Backend	LIGO
Attribute handling at objects	It would be really nice to handle attribute usage for groups, memberships and so on the page that handles the object. See the UI being developed at Uppsala University.	UI	Uppsala University Cru (CCCI)
Metadata about provisioning	Be able to attach built in attributes to identify on a UI where the group is used, and how long the provisioning takes (maybe range?)	UI/Backend	Working group meeting
Encapsulated composites	Would be nice if the UI treated include/exclude lists and require groups differently (maybe if they follow a naming convention) so the user doesn't need to know which group is the whitelist etc, the UI makes it easy. Maybe this is in phase 2... If looking at the full group the UI knows to edit the whitelist etc.	UI	Steven C. Chris

SSO to Grouper UI	Here at the University of Hawaii we also use CAS for authentication. SSO to the Grouper UI would be very beneficial	Backend	University of Hawaii (taken from comment below) Cru (CCCI)
Batch membership update from UI	Be able to easily assign multiple groups to a subject. Right now, there doesn't seem to be a way to manage membership from a subject point of view. Everything is done through the group page. It would be great if there was a way to manage membership from the subject info page. That way, we could easily add a single subject to multiple groups. It would be even better if there was a mechanism like the "Import members" from the Lite UI's group page where you can import a CSV file or input members directly on screen. That could be used to list the groups in which we want to add a subject. Also should be easy to remove multiple groups from a subject at once.	UI	U Montreal
Moved to "Guidelines". Hide complexity by default	Hide all of the "internal" attributes (such as ID, ID Path, UUID, etc.). Only show the user friendly names, paths, and descriptions. The details will only become visible through a special setting. The Lite UI does this now, and I think the entire UI should do this going forward. I'd recommend that each user have a "profile" where they can set this kind of option globally... so that power users can always see the IDs, etc., but "normal" users will never be bothered by them.		Cru (CCCI)
UX participation	User eXperience professional can help with process to ensure good UI usability and workflow.		U Chicago
GroupType Attributes	Add functionality for GroupType attributes to have a possible list of items a user can select. This will include showing a default value if one is presumed. There are of course cases in which a value needs to be supplied by the user, but there are many in which the possible answers are known and can be shown to the user for selection even if it is a boolean. To use the mailingList attributes as an example. The alias will need to be provided by the user. The allowAttachments and moderated are true/false questions. There can be a drop down list to select true/false. A default value of say false can be specified in the metadata. For attributes that are lists of users, the user should be able to click on the attribute or an obvious button next to it to engage the selection process.	UI	UC Berkeley

Input-Gathering Process

We'll aim to conduct the input gathering process as follows:

Date (s)	Activity
April 1 - May 31	Gather community requirements and requests in table above. Weekly reflection on grouper-dev list of new requests and any discussion of them. Counter proposals? Acceptable? Relative priority? Do others want to add their names as Requesters?
April 23	I2 SMM Grouper WG F2F
May 1	Float initial proposal for technology stack to be used to build v2.2 UI. Post to grouper-dev.
May 1-14	Drive discussion on grouper-dev of tech stack, with intent to settle as much of it as possible during this period.
May 18?	IAM Online focused on Grouper v2.2 UI planning & requirements
May 15	Float initial proposal for UX process to grouper-dev
May 15-31	Drive discussion on grouper-dev of UX process with intent to settle it during this period.
June 1-15	Finalize acceptance status of each request by grouper-dev and assign each to a priority class.

Screen mockups

This is the intro screen which is like the confluence main screen. It introduces Grouper (optionally), shows favorites, recent activity, links to do various tasks, etc.

Note, the text, colors, etc are only to put something on the screen to show functionality, it will actually look different than this. Also, this was discussed with one of Penn's experience designers.

My School



Welcome to Grouper

Grouper is... intro help mobile/accessible version

Services Favorites My Services

Show details

File sharing

Mail lists

Wikis

My recent activity Favorites All

Managed group aibic

Viewed permissions bicid

I want to...

Browse the repository

Search

Manage groups

Manage permissions

After the intro screen is the browse screen. This shows a folder structure on the left, filtered potentially to make it easy to navigate. The screen is laid out like frames, though they are not HTML frames. The panels are able to be resized or hidden. There is a context menu on each item in tree (e.g. create group in folder)

My School Browse Repository

Group
Menu

Filter

- ☐ it
- ☐ library
- ☐ school of business
- ☐ apps
- ☐ file sharing
- ☒ admins
- ☒ users
- ☐ school of engineering

Group Summary

Current location: ☐ school of business ☐ apps ☐ file sharing ☒ admins

Name	admins
Path	-----
ID	admins
Description	-----

Manage members

Privileges

Other actions

Currently the UI has a combobox control. The new UI could incorporate recently selected results, favorites, and an advanced modal search dialog

Add Member

Enter search text...

John smi

Advanced search

Add member

If you search for one char, it will say to enter more chars, but will also show the recently selected and favorites (if matches by search text... space or star or percent matches all)

Add Member

Enter search text

John Smi
Recently selected John Smithereen (js) (not active) Student (also: staff)
Favorites John Smitherton (jsmi) (active) Faculty (also: alumni)
Search results John Smith (jsm) (active) Staff John Smithe (jsmit) (active) Faculty ... too many results, narrow your search

Can browse or search if applicable to the combobox. Result will populate the underlying combobox.

Add

Enter

Advan

Advanced Subject Search

Search Browse

Search text	John smi
Source	☒ people ☐ groups ☐ principals ☐ external people ☐ local entities

Search:

Results:

Recently selected
John Smithereen (js) (not active) Student (also: staff)

