Tests for Access Management

Background:

(From Saturday, Sept 24 2016 email)

The 'toggle access' functionality was only working in the context of a single unit -- click a unit once to give all contained users access to a room, and click again to remove access. That was working fine.

But, if you clicked one unit to add access, and then a second unit with overlapping membership, users in both units would end up having their access revoked -- not what we want.

The new system first tallies all the units that have been granted access to a room. It then generates two lists -- one of all the users in those units who are routable, and a second of users who presently have access. The final steps involve comparing the two lists using array_diff(). If someone is in the first list (potential users) but not the second, they're given access to the room. If they're in the second list (those who presently have access) but not the first, their access is revoked.

The nice thing about this approach is that it's a general solution to several different scenarios. It's code that runs as part of all of the following now:

Access is given to a unit
Access is removed for a unit
A user is removed from a unit
A user is added to a unit
A user's 'routable' status is changed

All this functionality is so critical to the system, and I think it's imperative that we do some thorough testing. Could you design a few tests to run to ensure that unit/user access is behaving as expected? We want to make sure that everything from basic management (click a unit to give it access to a room) to advanced cases (remove access to all rooms when a user has been removed from all the units that gave him/her access to those rooms).

Expectations for Consideration:

Could you design a few tests to run to ensure that unit/user access is behaving as expected?

1. Sharon Expectation: The routing should work without any chat rooms assigned.
2. Sharon Expectation: The routing should work with any combination of chat rooms assigned.

3. Is it clear to users that access and notification are tied? Should they be?

Factors for Consideration:

A well–performed experiment may provide answers to questions such as:

- What are the key factors in a process?
- At what settings would the process deliver acceptable performance?
- What are the key, main and interaction effects in the process?
- What settings would bring about less variation in the output?

Source:
http://asq.org/learn-about-quality/data-collection-analysis-tools/overview/design-of-experiments.html
Accessed 26 September 2016

Key Factors:

Individual Components:

Access
Unit
Routability
Status
Adding
Removing
Room
Individual
(?)Unofficial versus official organizations

Settings:

Browser:

Chrome Windows
Firefox Windows
IE Windows
Chrome MAC
Firefox MAC
IE MAC
Safari

Networking Concerns:

Certificate/Performance Issues on the following:

VPN Online.usfsp.edu
Local host
USF Guest Network
USF Gold (No longer active, but are people trying to use it?)
Eduroam

Interaction Expectations:

By User:

Student
Router
Recipient

By WorkFlow:

1. staffRouterEditor, and put people in appropriate units.
    a. Any problems because of collapsed “alphabetical” menus?
2. As someone who is in a chat room, routing to the proper authorities. (Question: should there be an alert or other indicator if a unit has not been authorized by the main router? Is a disabled checkbox enough?)

By Page:

“Ask A Question” page
chat.php
staffRouterEditor.php
chatManagementEditor.php
Courses.php -- just check to make sure you CAN get to chat and ask-a-question page from main menu

By Role:

Role of 1, 5, 10
Role of Other?

By Type of Edit

Editing Chat Room
Editing Staff Access
Editing Unit Access
Editing Routability

Notes:

27 September: I had to return staffRouterEditor.php to the version dated 26 September 2016; Sharon Austin (SA) code broke Casey Frechette (CF) code on that page.   Testing continued without the expanded alphabet configuration on the right hand side.

To Test:

Basic Tests:

Access is given to a unit
Access is removed for a unit
A user is removed from a unit
A user is added to a unit
A user's 'routable' status is changed

Basic Tests with Room ID:

Access is given to a unit in a room
Access is removed for a unit in a room
A user is removed from a unit in a room -- currently not set up
A user is added to a unit in a room -- currently not set up
A user's 'routable' status is changed in a room -- currently not set up

Combinations “By Order of…”:

Note: Although below reflects cases in which the same “functionality” is being tested, the order is also being tested to ensure that the order in which one makes the changes does not affect the expected outcome.

A user is added to a unit then, access is given to a unit
Access is given to a unit then, a user is added to a unit

Access is given to a unit and routability is changed to 1
Access is given to a unit and routability is changed to 0

Access is removed for a unit and routability is changed to 1
Access is removed for a unit and routability is changed to 0

A user is removed from a unit and routability is changed to 1
A user is removed from a unit and routability is changed to 0

Access is removed for a unit and routability is changed to 1
Access is removed for a unit and routability is changed to 0

A user is added to a unit and routability is changed to 1
A user is added to a unit and routability is changed to 0

A user's 'routable' status is changed and routability is changed to 1
A user's 'routable' status is changed and routability is changed to 0

Routability is changed to 1 and Access is given to a unit
Routability is changed to 0 and Access is given to a unit

Routability is changed to 1 and Access is removed for a unit
Routability is changed to 0 and Access is removed for a unit

Routability is changed to 1 and a user is removed from a unit
Routability is changed to 0 and a user is removed from a unit

Routability is changed to 1 and a user is added to a unit
Routability is changed to 0 and a user is added to a unit