

Option A Project Questions

[Organized by request of Professor Cannata]

Note: The design suggestions below might (probably doesn't) match every team's idea of how to handle user rights, but it's a start on a conversation with ARL. Also, I tried to keep every unique question from the original list, only removing duplicates, and organizing under categories.

#pin

Security/User-Rights

Two classes of users is implied, but the differences are not described in detail in the design document. We propose two types of users “standard” and “admin” with the properties described below. Is this the desired user access model?

An admin user would have full access to make changes to both the notice_lists table and the notice_list_members table while the standard user would only be able to view the notice_lists table but have full access to make changes to the notice_list_members table.

- **Standard-users**

- View/add/remove, ONLY self, ONLY to lists with matching division code

Not quite. A standard user should be able to edit memberships for any list belonging to their lab (regardless of division). They should be able to add/remove members that may belong to their own or any other lab or division.

- NOTE: The design document as written gives an unexpected level of power to standard-users (ex: changing list status to active/inactive, changing list division codes).

Sorry I wasn't clearer. A standard user may not make changes to the list itself (notice_lists table), they can only make changes to the membership. The only exception might be to allow a standard user to modify the list_description, but to keep is simple let's say notice_lists is view only for a standard access user.

And a standard user would have Management functions as copied from below:

- List management (add/remove members, see members not on list, search active employees, alert to inactive members on list, etc.) for lists that belong to their same lab.
- General list maintenance (clone membership, remove employee from all lists, view employee list memberships) for lists that belong to their same lab.

- **Admin-users**

- Create/edit/maintain ANY list

- Or limited to admin only over lists in matching division? If that's the case, and they change a list to another division, do they lose the ability to manage that list?

An admin user would have full access to create and change any list belonging to any lab or division.

- List management (add/remove members, see members not on list, search active employees, alert to inactive members on list, etc.)

Agreed.

- General list maintenance (clone membership, remove employee from all lists, view employee list memberships)

Agreed.

- We are proposing keeping an Admin table separate from the Active_Employees table to keep track of admins (who are active employees). Only admins can add/remove other admins to the admin table. Does this design match your expectations?

Our authorizations are actually handled by several tables as well as Oracle security policies. We don't expect you to emulate that so it is fine to keep the admin list in a separate table. You could also just add an admin flag to the active_employees table and leverage that instead (like is_admin Y/N). For that matter, you could also add a flag for is_notice_list_user or something like that to identify the standard users.

•Logistical

- Should authentication be done using APEX accounts or something like LDAP?

LDAP. We use network account ids – that field should be added to the Active_Employees table.

- How do we relate the current APEX user to his/her Active_Employees record?

See above

- Must every user of this tool be an Active_Employees? If not, how would be auto-populate the division code? (In other words, can there ever be “external” users who are not employees?)

No, at this time we would not allow external users or external addresses. Any user of the form must be an active employee and must have standard or admin authorization to run the form.

•External Table(s)

- What is the real/complete schema for Active_Employees?

The table sample I gave you is a view based on several larger employee tables we maintain. They are given as starting examples and can be modified as you see fit for the project.

- What's the purpose of "External View Name" in Active_Employees?

That text string is how we will tie the dynamic list of members to an existing email list recognized by our network. Actually tying the pieces is beyond the scope of this project as it would involve other ARL systems.

- Will there be separate columns for first and last names?

The table active_employees is derived from does have separate fields for first, middle, and last name. The employee_name field is then derived from the 3 separate fields. Since active_employees is essentially reference table for this project, there is no need to break out the pieces of the name, but if you find that makes something easier for you, you may certainly add the separate fields to the table.

- If this database is specifically used to manage notification lists of employees, why would we have employees with an inactive status?

Effectively an inactive employee is an employee who is no longer appointed to the lab (resigned, retired, etc.). The system will know not to attempt to email an inactive employee. But say a new list is created on 3/15 then on 3/25 a member on that list wins the lottery and leaves the lab. We can certainly automate eliminating them from the system entirely, but we've found it's been much easier for users to review and duplicate the existing access of the inactive employee when they hire a replacement than to have to start over from scratch.

- Are there tables for division and lab codes and will we have access to them?

- If not, should users be given the opportunity to specify any characters as division/lab codes or should they chose from existing lists?

I will attach sample data at the bottom of this document. Lab/div info should be treated as cross-reference data for this system.

- Can a user have more than one division or lab?

No, an ARL employee is currently only assigned to a single division within a single lab.

- Is division a subgroup of lab? (Or what, if any, is the relationship between divisions and labs that we should be concerned with?)

Yes, one lab may have one or more divisions tied to it.

•List Action Questions

- Can lists be deleted or do they live forever and just become inactive?

At this point, they will remain in the system as inactive lists for historical purposes.

- What (if anything) should be done with inactive lists?

An inactive list can be viewed by the user but cannot be changed and cannot be added as a member to another list. An admin should be able to clone an existing inactive list into a new active list, copying over all currently active employees from the inactive list to the new list for membership.

- Would it be good to have an undo/redo history for any list edits?

That is not a requirement, but it's always a helpful user feature.

- Should an inactive list be able to be a member of a list?

No, not at the time it is being added to a different list. As with employees however, a list may become inactive sometime after it was added to another list and the user should be warned. In fact, when an admin goes to inactive a mailing list, the form should warn them at that time if that list they are inactivating currently belongs to other active lists.

- Should any view that shows list members be a tree view to allow users to expand the tree? Or should the hierarchy be hidden and the list be made flat (while also removing duplicate entries)?

The user should be able to ultimately see the full list of members which means if one of the members is another list, the user should be able to expand it down.

- Do we need to absolutely disallow the creation of list cycles or would it be ok to let them be created and instead, should operations that traverse the directed graph just know when to terminate (so they don't recurse indefinitely)?

You should be able to prevent obvious cycles where List A is a member of List B then List A tries to add List B to its own membership.

- What are the most common forms of list modification (eg new users, migration between groups, migration of entire groups)?

Once a list is created and has initial membership the most common action by far is adding and removing single members. Currently this has to be handled by MISS staff which is why this project idea was on our list. Brand new lists aren't created that often, and there currently isn't a method to migrate entire groups automatically.

•Miscellaneous Questions

- The project specification says we can search employees by partial name. Does this mean we can search by first and/or last name or even by syllable?

Yes, the search should be a straight text string search that looks in employee_name (or the first, middle, last names if the fields are broken out) for any occurrence of the string the user enters. You may want to add a toggle to restrict the search to active employees only. Just for reference, we have over 7000 employee records, with over 700 being considered active employees.

- Are there any specific statistics regarding the state of lists that would be of interest to APL on an analytics page?

Nothing comes to mind.

- Of the optional features, which are priority?

I re-read the specs and I don't see any optional features in what I submitted. I have mentioned a couple of things earlier in this doc, but none have any priority over anything else. If there is something specific you're interested in, please let me know.

- How should transactional conflicts be handled in the user interface? (For example, user is adding members to a list that another user deleted at the same time.)

- Should there be a commit fail message? Would the user whose commit fails lose their work?

I'm not familiar with how APEX handles locking. Professor Cannata might be better able to answer. In general, I would expect that the first person to make changes would lock the list such that when the second person pulls it up and attempts changes they would get an error that the list was already locked. In that case, there wouldn't be lost data as the second user wouldn't have been able to make changes. Otherwise, I suppose if both are allowed to work, then yes, the first person to hit the commit gets the prize and the second/third/fourth user would lose their changes – and hopefully at least get some sort of graceful message as to what happened.

•Answered Questions

- Q: How do file uploads/downloads relate to the application? What should we expect the users to upload?

- A: Professor said users will be able to upload/download blob files like their own little dropbox. This might not be a feature ARL wants.

We have no need for file uploads/downloads for this project. However, if you wish to do that as an optional feature, that's fine. We do in fact have our own upload/download web interface to move stuff to and from a user's desktop, but perhaps APEX can give us something more user-friendly.

- Q: Is there a hierarchy of Notification Lists?

- A: Sort of, it is an acyclic directed graph which could be a tree if the edges were undirected (if I'm remembering the graph theory right)

There is no explicit hierarchy to enforce in the structure.

- Q: Will we have to provide an easy solution to connect our tables to the actual Active_Employees table?

- A: I believe Professor said we should work against our local 'dummy' implementation of Active_Employees. Actually connecting to ARLs system is probably beyond the scope of this project and something 'they' will do over the summer.

You are correct that directly connecting to any of ARL systems will not be possible. The active_employees table should be populated with dummy data for various active and inactive employees, then used more as a cross-reference table for the rest of the project.

- Q: When removing an employee from all mailing lists, does this imply we are removing the employee completely?

- A: No, we will not have any ability to modify the Active_Employees table, therefore “removing the employee completely” is not something we can do..

Agreed (with this answer), just because an employee is removed from a list one day doesn't mean another user won't want to add him to a list the next day.

- Q: Does having someone on your mailing list imply you are on their mailing list?
 - A: No, question doesn't make sense given the design specification
Agreed (with this answer)
- Q: Are there multiple root notification lists, or can any list be connected to any other list?
 - A: Short answer, any list can be connected to any list, but cycles will have to be avoided.
Agreed (with this answer)
- Q: Is there a requirement for exporting/importing data? If so, what information should be imported/exported.
 - A: Professor said we should be able to import/export to our 'dummy' Active_Employee table. We might also want to import/export list membership information, but Professor wasn't sure about the need for that.
As mentioned above, we have no need for this feature in this particular project but if you have the time and want to add it as an option, that's fine.

Sample Lab/Div data:

Lab	Div
ADS	ACC
ADS	ADS
ADS	CON
ADS	PUR
ATL	ASD
ATL	ATL
ESL	ESL
ITS	CSD
ITS	ITS
ITS	MISS
SGL	SAT
SGL	SGL
SISL	SPD