Class Project Collaboration

The following is a general proposal for a collaboration between the Applied Research Laboratories (ARL) MISS office and a UT software development class for a class project. The class is led by Phillip E. Cannata, PHD, Adjunct Professor with the Department of Computer Science in the College of Natural Sciences at the University of Texas at Austin. The proposed project would be for a CS 347 Data Management class Dr. Cannata teaches on campus.

In general, this collaboration would mean that the ARL MISS office will provide Dr. Cannata with the outline and business rules of one or more data entry projects. If an MISS project is chosen by the class, the student developers will spend part of the semester creating a data entry system using an Oracle database for data storage and the Oracle Application Express (APEX) tool for developing the data entry screens. All primary interaction will be coordinated through Dr. Cannata at UT and MISS manager Carrie Woodworth at ARL. Specific dates and requirements for individual projects will be included with a copy of this document for each proposed project.

1) Visits to ARL
   Onsite ARL visits are not a requirement of this collaboration. However, should it be beneficial to the project and to the students involved, arrangements can be made to allow site visits to ARL. Such visits would be held in a front ARL conference room without full building access. Conversely, should it be beneficial to the project and to the students involved, appropriate members of the MISS development team will make themselves available for visits to campus to meet with the student developers. Should an onsite visit inside of the lab be desired, the request will be sent through the appropriate ARL authorities for approval and specific arrangements.

2) Access to ARL systems
   No project will require students to have access to any internal ARL systems. Proposed project descriptions provided by MISS to the class will not include any actual table names or data currently in production use at ARL.

3) MISS Deliverables
   MISS agrees to provide written project specifications for each proposed project. The specifications will be in any combination of Microsoft Word, Microsoft Excel, and/or Adobe PDF as appropriate to convey the information. The specifications will include a general outline of the project, a starting point for any data definitions, explanations of desired functionality, and any business rules that should be applied. In addition to providing descriptions of potential data entry projects, MISS agrees to make one or more staff developer available to answer questions or provide guidance as needed throughout the semester. MISS agrees to acknowledge and respond to all project-related questions and requests within 48 hours of receiving them during normal business hours.

4) Benchmarks and Progress Reports
   Specific dates will be included with specific proposed projects. In general, student developers will provide email updates on the project on a weekly basis, with a more detailed updated at the midpoint between accepting the project and the end of the semester.

5) Class Deliverables

Class Project Collaboration

At the conclusion of a project, Dr. Cannata and the student developers will provide a presentation of what they have completed. They will also provide an electronic version of all associated source code. All provided code must be open and available to MISS to review in detail. No pre-compiled or hidden encapsulated code will be accepted. All code provided must be able to run on a local installation of an Oracle Application Express server and Oracle database that MISS will provide and maintain internally. The final product must not have any dependencies on or references to any external systems, servers, or websites unless explicitly stated in the requirements provided by MISS.

6) Cost to ARL and to UT

The only cost to ARL for this collaboration will be in MISS staff time. The APEX software license and the hardware to run it are already available to ARL under current licensing and usage. There are no additional monetary charges to be incurred by Dr. Cannata or his students for these projects. Projects will be developed using tools Dr. Cannata already provides, leveraging the Oracle APEX online development sites.

7) Current Contact Information

Phillip E. Cannata, PHD
Adjunct Professor
Department of Computer Science
College of Natural Sciences
The University of Texas at Austin
2317 Speedway, Stop D9500
Austin, TX 78712
Office: TAY 2.124
512-263-3451
phil.cannata@oracle.com

Carrie A. Woodworth
Manager
Management Information Systems Services
Applied Research Laboratories
The University of Texas at Austin
10000 Burnet Road
Austin, TX 78758
512-835-3973
carrie.woodworth@arlut.utexas.edu

Notice List Project

Project Title:		Notice List Project
Proposed Start Date:		23-Mar-2015
Proposed Mid-point Review:		24-Apr-2015
Proposed Completion Date:		12-Jun-2015

Toolset

The class will be using apexea.oracle.com area to build the prototype. Oracle Application Express is expected to release version 5 in May or June of 2015. The class plans to provide a version 5 compatible solution, however MISS will also accept a solution based on the current production 4.2.x version of APEX. The final version of the product given to MISS should also be compatible with the Oracle 11gr2 database and should be able to run on current versions of the Internet Explorer, Firefox, Chrome, and Safari browsers.

Class Deliverables

The completed project should include all source code to generate, deploy, and run a data entry system that meets the Notice List Project requirements. This should include all necessary source code as well as any SQL or PL/SQL scripts to generate tables, functions, procedures, and/or packages required for the project.

Project Details

The Notification List project entails creating data entry screens to allow users to dynamically control membership to any pre-defined mailing lists they have access to. Mailing lists themselves can only be created by users with special access privileges. The final entry screens should allow for the following three distinct areas of functionality.

Maintain the Lists

Users with appropriate access should be able to create, edit, or change the status of a notice list through data kept in the notice_lists table. The user should be able to see/edit the list name, the list description, the ownership division code, the external view name, and the status. The user should be able to view but not update the creation date and status effective date. For a new list, the user is required to enter a name for the list as well as a text description of what the list will be used for. For each new list, the system should automatically pull a unique numeric identifier via a sequence to tie to the record, automatically set the list status to Active, and automatically set the creation date and status effective date to the current system date. The ownership division code should default to the user's division but the user should be able to override that and identify a different division code for ownership. Users should be able to search and display information on any existing list for which the ownership division matches their own division and should be able change the list status from Active to Inactive or vice versa. Any status change should automatically update the status effective date to the current system date.

Maintain Individual List Membership

Once a list is created and defined there should be a data entry screen to allow the user to pull up the current membership to add or remove members and otherwise maintain the data in the notice_list_members table. Members may either be individual employees or other existing defined mailing lists. In addition to viewing the current membership for edits, the user should also be able to see a list of active employees (from any lab or division) that do not currently belong to the list being reviewed, then be able to identify which employees should be added to the notice list via either a drag-and-drop method or a checkbox method. A user should be able to select more than one member at a time to add to a notice list. The list of active employees should be searchable and there should be filters available to screen on lab code, division code, and full or partial name searches. An employee must have an 'Active' status to show up on the list of available employees to add to a notice list. An 'Inactive' employee cannot be added to a list. If a list is displayed that contains an 'Inactive' employee in its current membership there should be some visual indicator to alert the user to allow them to resolve the issue. On the view of the existing list membership, the user should be allowed to remove any listed members or member lists either via a drag-and-drop method or a checkbox method. The user should be able to select more than one member at a time for removal.

The active_employees table is updated and maintained outside of the scope of this project. Student evelopers will need to create the table and fill it with test data for use in this project, but the active_employees data is for reference only and the user of the APEX screens should not be able to edit any data in the active_employees table.

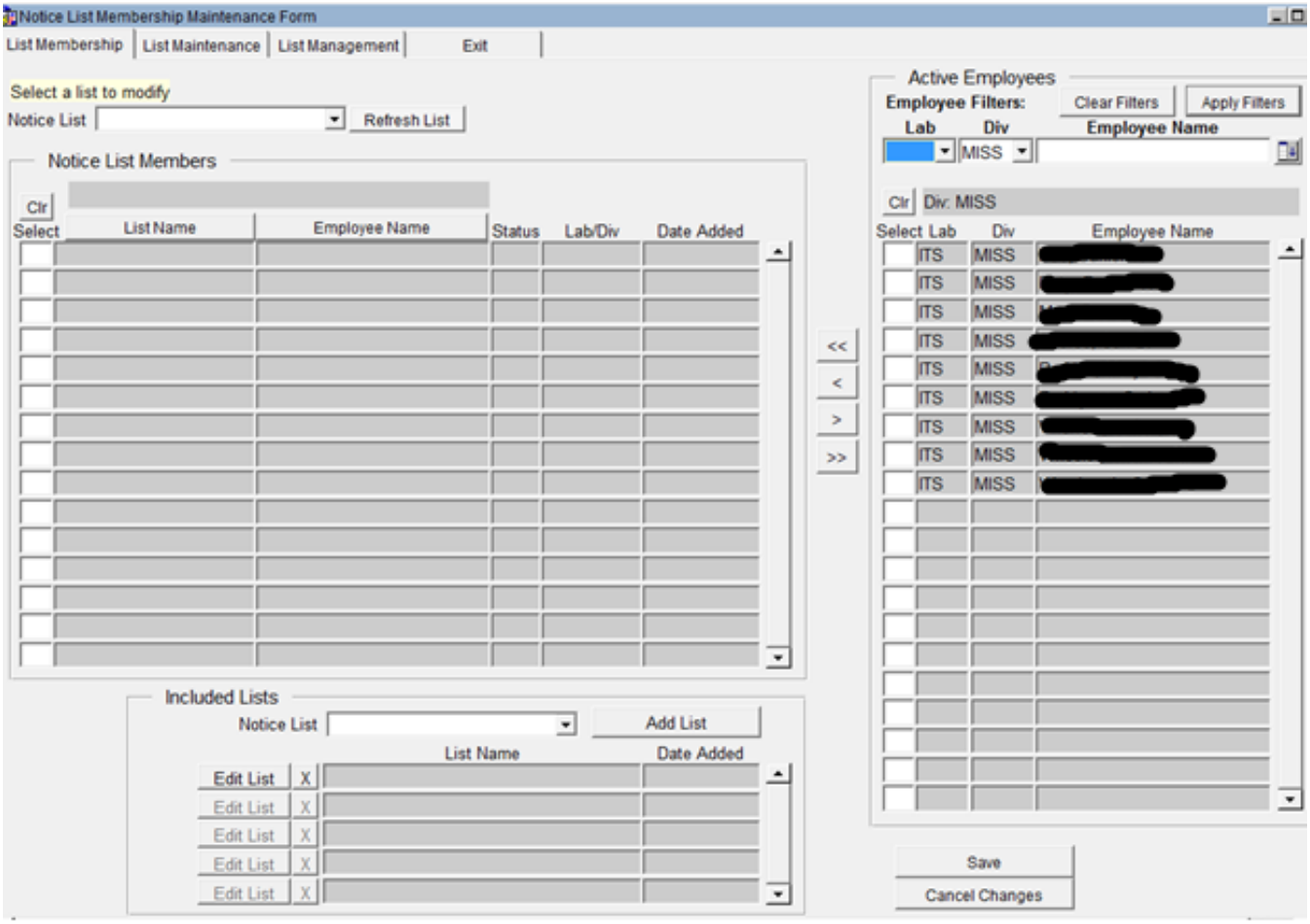
Allow General List Maintenance

There should be a List Maintenance screen to allow the user the following general functionality against any Active notice list: 1) Clone all current membership from one employee (either an Active or Inactive employee) to another employee (Active employee only). 2) Remove a specific employee (Active or Inactive) from all mailing lists that they currently belong to. 3) View membership such that the user can see all lists that a given employee currently belongs to.

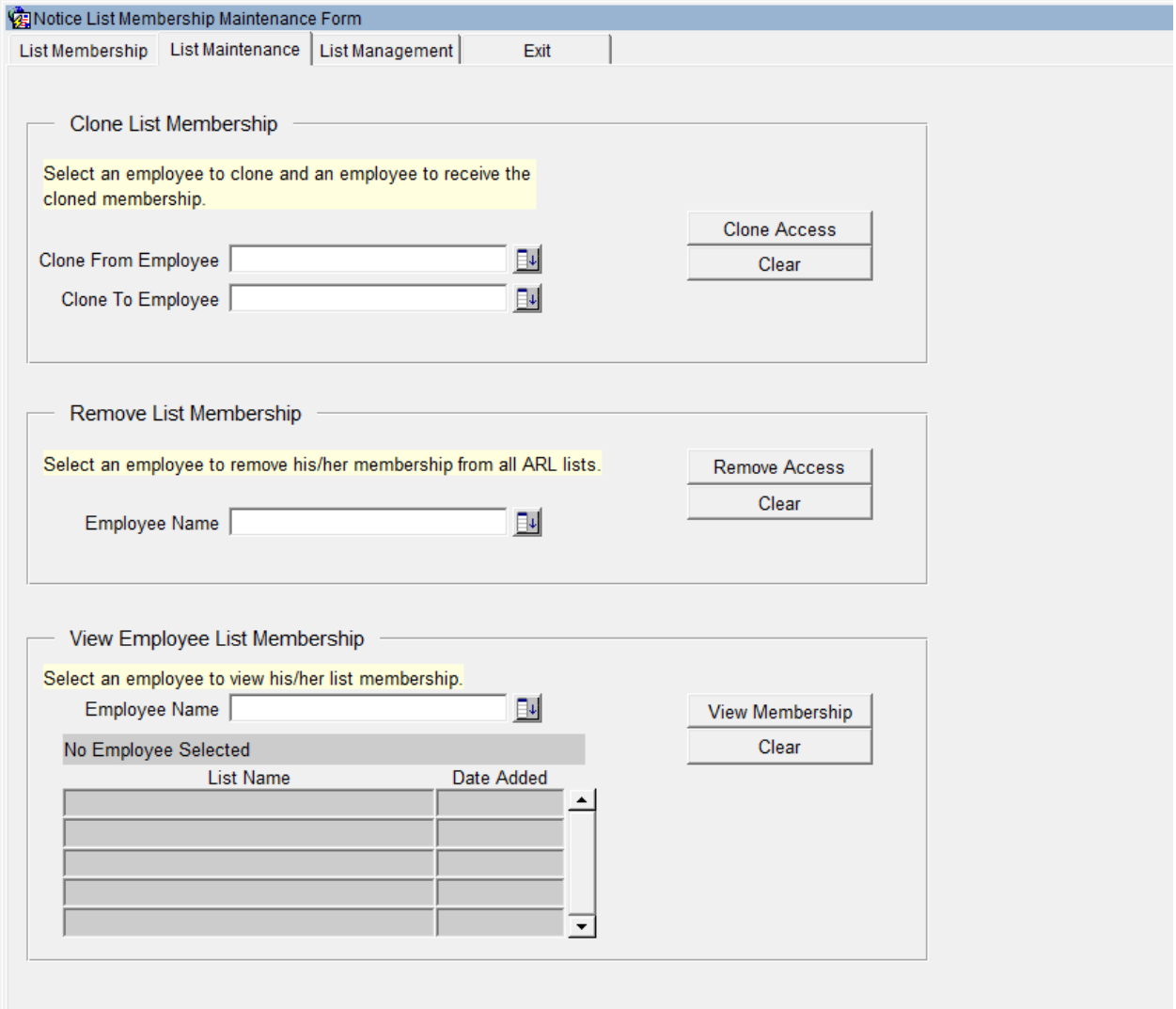
Sample Screens

The following are two screenshots of forms layouts that were generated as a proof of concept for this project. The layout itself does not need to be duplicated, only the functionality the screenshots represent.

Notice List Project



Notice List Project



Sample Table Layouts

The following shows sample data descriptions for the data tables mentioned in this project. Student developers may find that additional fields or additional tables may be needed to complete the project.

Notice List Project

Active_Employees:					
Name	Null	Type			Sample Data
EMPLOYEE_ID	NOT NULL	NUMBER	Unique identifier for an employee generated from a sequence.		4545
EMPLOYEE_NAME	NOT NULL	VARCHAR2(30)	Employee name in the format "LastName, FirstName MI."		Doe, John Q.
EMPLOYEE_LAB	NOT NULL	VARCHAR2(4)	4-character lab code identifying which lab the employee is appointed to.		ITS
EMPLOYEE_DIVISION	NOT NULL	VARCHAR2(4)	4-character division code identifying which division of a lab the employee is appointed to.		MIS
EMPLOYEE_STATUS	NOT NULL	VARCHAR2(1)	A for active or I for inactive		A
STATUS_EFF_DATE	NOT NULL	DATE	Date the employee_status became effective		1-May-2014
Notice_Lists:					
Name	Null	Type			Sample Data
LIST_SEQ_ID	NOT NULL	NUMBER	Unique identifier generated from a sequence		55
LIST_NAME	NOT NULL	VARCHAR2(40)	Unique name for the mailing list - no spaces allowed.		Hardware_Updates
OWNERSHIP_DIV_CODE	NOT NULL	VARCHAR2(4)	4-character division code		MIS
LIST_DESCRIPTION	NOT NULL	VARCHAR2(3500)	Text description for defining the usage of the list		Notify users of pending downtime.
EXTERNAL_VIEW_NAME		VARCHAR2(50)	50-character text field to reference an external view name.		hw_upd
CREATION_DATE	NOT NULL	DATE			1-Jan-2015
LIST_STATUS	NOT NULL	VARCHAR2(1)	A for active or I for inactive		I
STATUS_EFF_DATE	NOT NULL	DATE			1-Mar-2015
Notice_List_Members:					
Name	Null	Type			Sample Data
LIST_SEQ_ID	NOT NULL	NUMBER	Unique identifier tied to		55
EMPLOYEE_ID		NUMBER	Unique employee id tied to active_employees.employee_id. A record may either have an employee_id or a member_list_id but not both.		4545
MEMBER_LIST_ID		NUMBER	Unique list id tied to notice_lists.list_seq_id. A record may either have an employee_id or a member_list_id but not both.		
DATE_ADDED	NOT NULL	DATE	Date when the member was added.		1-Jan-2015