



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Modernizing Organizational User Account Management

A.N.Mendoza

September 3, 2014

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

ABSTRACT

The Environmental Restoration Department at Lawrence Livermore National Laboratory utilizes different web applications as part of their daily operations. Many of these web applications are in need of improvement using modern technologies and programming concepts. One such application is the User Account Request tool, which allows an employee, or employee sponsor, to request various user web accounts as well as equipment, such as a desktop computer. This tool is vital to the Environmental Restoration Department for getting employees and interns the tools they need in order to work. The need for a better web application is what this research is based upon.

The driving force behind this project was the need to create a more streamlined and efficient tool for employees to use. The main process involved combining the original three-page system into one page, using newer programming languages and concepts that were not available when the tool was first developed. Other project goals were to allow for data management (in regards to managing account requests) and modularity, which is a way to allow the code to be reused for other applications. To achieve this, research involved learning and implementing best practices for languages and concepts such as Perl, Javascript, jQuery, and Asynchronous Javascript and Extensible Markup Language (AJAX). General programming best practices were also studied and used throughout this project.

PROBLEM

The User Account Request web application used by the Environmental Restoration Department (ERD) at Lawrence Livermore National Laboratory is an essential tool for creating web, computer and systems accounts for employees and interns that work for the department. The previous tool used older web technologies and was cumbersome for employees to use. The code used to create the web application was also not well organized, making it difficult to update and manage. This project focused on completely recreating and restructuring this web application, with the goal of having a more efficient tool and added functionality.

The workflow of the original web application was very convoluted. The user was required to input and submit information using an HTML form. Upon submission, a second page would open and display the user's input, as well as any errors they may have made. If a user had made any errors, the user would be required to return to the first page and edit the input before submitting once again. This process would continue until the user received no errors on the second page. The input would finally be sent to different managers via email, through which the account request would be managed. This workflow was very inefficient and potentially confusing for the user.

SOLUTION

The new workflow was created with many goals in mind in order to eliminate the issues that the old workflow presented. One problem that needed to be addressed was the constant web page loading from navigating to different pages. Another issue was the error checking, which only occurred once the user navigated to the new page. Once the user gets through input validation, they would need to be able to submit this input and have it emailed to the appropriate managers. Besides all of these fixes, an added goal for this application was the ability to send this input to a database, where it can be managed and viewed by the management team.

The new workflow needed to use fewer web pages than the original web application. This issue was addressed by combining all the HTML form elements into one web page. Previously, the user had to input information on both the first and second web pages, which is a disorganized process. The new application allows the user to enter all input all at once. This allows the user to submit all information together, which is convenient for both the user and any developers who may need to work on the application in the future.

PROCESS

As the user provides input, the web application simultaneously checks for any errors that the user may have made. For this purpose, AJAX and jQuery have been utilized in this project. AJAX stands for Asynchronous Javascript and XML, and is a concept that uses multiple technologies that include (but are not limited to) HTML and Javascript. From the Mozilla Developer Network, “In a nutshell, it is the use of the XMLHttpRequest object to communicate with server-side scripts” (Mozilla Developer Network). AJAX is also asynchronous in nature, meaning that it is able to do this without reloading the page, making it a useful tool for validating input as it is entered. jQuery, a Javascript library, also contributes to the asynchronous error checking via the jQuery Validate plugin. jQuery Validate allows for client-side form validation using various methods and providing default error messages (jQuery Team). The jQuery Validate plugin is also useful due to its flexibility in allowing the developer to create custom methods and error messages for unique customization. The usage of custom error messages adds more convenience by not only notifying the user of what error they have made, but also where the error is located. By utilizing AJAX and jQuery together, the user is notified of any errors made within the form without needing to reload the page or navigate to a new one, making the new application much more efficient and convenient.

If the input contains errors, the user will not be able to submit it until the errors are fixed. After the input has been validated, it is ready for the submission process. The user simply clicks a button and the input will be sent out in an email through the use of Perl scripts. A basic overview of Perl from the Perl Programming Documentation, “Perl is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more” (Robert). Besides being used to load the HTML elements for this web application, Perl is used to take the information input on the client-side and send it to various managers via an email. This email is sent using a subroutine (a function in Perl), which is triggered when the user clicks the button to submit and the information is validated.

Additional requested functionality for this web application is the ability to send the account request to a database, where it can be viewed and managed. The original web application had no other means of management besides an email, which would be sent back and forth between various managers. It is very easy to lose information this way, and can also lead to confusion. By sending the information to a database, it can be viewed as well as edited at any time. The database of requests is far more convenient to manage and allows the managers to edit a request at any time. For instance, if an employee no longer requires a certain account, the managers would be able to remove that account from the request, while leaving the rest of the accounts in tact. This added functionality would create a more efficient account request system for ERD.

RESEARCH

This project has been a learning experience for many different reasons. Besides the need for research on Perl, Javascript, HTML, AJAX, and jQuery, there has also been a need for research on best practices. Best programming practices include concepts such as commenting, naming, and formatting. There are many

opinions on what best practices are, but commenting, naming, and formatting each have general guidelines that can be agreed upon. Comments are important for any code and should be updated as the code is updated. They should also be used to indicate what certain methods, functions, and subroutines are used for. In regards to names, according to Rob Caron from the Microsoft Corporation, “A name should tell ‘what’ rather than ‘how.’ By avoiding names that expose the underlying implementation, which can change, you preserve a layer of abstraction that simplifies the complexity” (Caron). In regards to formatting, there are many rules on different aspects, but this project relied heavily on standard indent sizes and bracket alignment. Best formatting practices say that the indent size should be standardized, in order to help align sections accordingly (Caron). There are also two different methods to bracket placement, but this project utilized a slanting style, in which the opening bracket is placed at the end of a line and the closing bracket is placed at the beginning of a line (Caron). An example of naming and formatting used in this project following best practices can be seen below:

```
if ($("#employeeForm").valid() == true){
    getEmployeeData();
}
```

Figure 1.1 Example of naming and formatting best practices within the User Account Request web application code.

CONCLUSION

The original web application shows the need for constant updates and utilization of best practices in development. The new web application is a way to streamline the workflow of a user account request and also created a more convenient, efficient application for both the user and the developer. By using best practices, this code will be easier to follow and update for other developers in the future. This application also leaves room for additional functionality and improvement, should the client request that. Other applications can benefit from

this research by utilizing programming best practices in order to create uniformity and clarity within code. This project is beneficial not only for use by the Environmental Restoration Department, but also for future developers interested in best practices. The project and research involved have been very enlightening and a useful means of learning development.

Works Cited

Caron, Rob. Coding Techniques and Programming Practices. Microsoft Developer Network. <[http://msdn.microsoft.com/en-us/library/aa260844\(v=vs.60\).aspx](http://msdn.microsoft.com/en-us/library/aa260844(v=vs.60).aspx)>.

jQuery Team. jQuery Validation Plugin. <<http://jqueryvalidation.org/>>.

Mozilla Developer Network. Getting Started-AJAX. <https://developer.mozilla.org/en-US/docs/AJAX/Getting_Started>.

Robert, Kirrily "Skud". perlintro. <<http://perldoc.perl.org/perlintro.html>>.