CECH Equipment Management and SOCO Journal Solution

by

Chavez Kattick

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2014 Chavez Kattick

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

Chavez Kattick 4/17/2014

Chavez Kattick

Russell E McMahon 4/17/2014

Russell McMahon, Faculty Advisor  Date
University of Cincinnati
College of
Education, Criminal Justice, and Human Services
April 2014

Table of Contents

Abstract 3
Introduction 4
Problem 5
Solution 6
1. Solution Details 7
   1.1 Users
   1.2 Timeline
Technical Areas 10
2. Resources 11
   2.1 Software
   2.2 Hardware
   2.3 Budget
Proof of Concept 13
Testing 14
Conclusion 16
References 17
Abstract

Inventory management has long since been transferred over to computer systems in the form of user-friendly tools. Humans are still responsible for designing the functions of each system; however the computer, despite being a tool, is responsible for the storing of vast amounts of data. The goal of this project is to create a tool which is based on CECH IT equipment and how it is operated. Along with this goal, the tool has to be easily upgradable as is necessary for changing environments. While developing an equipment management tool may be simple for an average programmer; creating a user-friendly, secure, and upgradable program proved more challenging. After testing many platforms, we now have a tool that can manage CECH IT equipment inventory, along with other applications in one easy-to-use secure environment.
Introduction

Technology today has grown to the point where we rarely need to write things down the old fashion way. Technology’s main purpose is to make tasks easier for humans to complete. It is the big selling point for today’s newest products. Every day we look for new ways to improve our lives by modifying the way we do things. Technology gives mankind an alternative way of accomplishing a task; an alternative process is made possible. Things like record keeping goes from paper to ones and zeroes in a computer. Moving data to a computer not only helps with the management of that data, but provides a history of trends that can be used to better manage another processes related to that data.

In the business world, moving data to a more manageable, efficient system becomes a job in itself. This is why we have information technology (IT) departments or departments related to IT. One of the roles of the IT department is to improve the speed, efficiency, accuracy, and dependability of data.

We look at the personnel in the IT service industry as people who provide services where needed to improve processes related to technology. What we don’t look at is the processes within IT which then provides us with services. This is how the University of Cincinnati College of Criminal Justice and Education (CECH) Web Consolidation Project came about. A student worker noticed the need for an improvement on how CECH IT managed different applications on various servers. This project spanned nine different servers with even more applications on each server. Shortly after the initial project was established, it was split up into pieces with two pieces being put into one project. The first piece is to improve the management of equipment for CECH IT. This department has a growing inventory starting at around fifty pieces of random
equipment that need to be tracked. The equipment includes items such as laptops, tablets, sound equipment, computer accessories, and other tools that are loaned out to the faculty of CECH. The second piece of the project is to move an application off of one server, modify it to the user’s preference, and deploy it on another server. This application is a journal for all of UC to use. This two piece project will be called the CECH Equipment Management and SOCO Journal Redesign Solution.

**Problem**

The problem begins with the focus on how CECH IT loans out their equipment. There is no technology involved in the loaner process. All records are written down and stored in a file cabinet. Management of inventory turns into a long and time consuming process with no real reports or tracking involved. The problem continues with finding out when the equipment is going to be returned. There is no way to efficiently keep track of what you have available. For example, a faculty member needs to borrow a projector. It turns out that there are no projectors in inventory right now; all of the available units have already been loaned out. In order to find out when the next available projector will be back in inventory, you have to go through all of your hand written records to see when it’s coming back. You also have the problem of late returns. In order to keep track of what should be in inventory on any given day you would need to go through all of your records. There is also the condition of your inventory items to consider. How would you know when to order a replacement item if the current one is nearing end-of-life? Mistakes are more than likely going to happen with this old system of doing things. Hours of productivity are wasted on these simple tasks that can take seconds to manage on a computer. As Chris Jesse states in his book, a Journey through Oz:
“Information technology is a duck. In itself, a duck is a net consumer of resources; it must be contained, requires constant care, becomes less valuable with age, leaves a mess wherever it treads, and is counterproductive until converted into practical application. Caring for ducks requires partners, those who grow the grain, mill the feed, and make the fence wire. Ducks have a mind of their own. They are oblivious to their purpose, and require proactive management if they are to affect a positive result. . . As I look at us, those who consume and implement technology, I see that we have become focused on making faster ducks—whose meat tastes no better, whiter ducks—whose feathers are no softer, bigger ducks—with eggs no larger, and smarter ducks—whose intellect does nothing but make them more difficult to manage. Alternatively, we must become duck farmers. This requires that we change our focus from the duck to those consuming the duck, to those supporting our efforts to produce the duck, and to the management of the duck.” (1)

From that idea of caring for the duck or the process itself, I focused on making the program easier to use, implement, manage, and modify. My target user would be those who have to manage this system after it has been deployed. This system and process of loaning out equipment will not turn into an unmanageable duck.

The academic journal site is another part of this project which needs some modifications on its own. It’s already a developed application but needs to be moved to a new server with updated features. This site is used by all of UC so it carries the same priority as the equipment management problem.

Solution

The CECH Equipment Management and SOCO Journal Redesign Project plan set various goals in place to begin developing solutions to the issues stated previously. When developing a
solution from within IT, the system requirements have a habit of changing and that change; no matter what size, can impact the way that IT uses the solution. That leads us to the first goal of the project. That goal is to design an application that is easy enough to modify when needed. The same can be said for the relational database behind it all; this means that there needs to be a way to easily manage it from an easy-to-use graphical interface. A second goal for the application is to allow the administrator to change or customize the forms, reports, and layouts. The applications should be able to process data and store it to a database. The database should be able to store the data and run queries when needed and the administrator should be able to pull reports as needed through the application or application platform.

The main goals of this project is to allow the users to check in, check out, manage inventory, run reports on inventory, and be able to create other features if the need surfaces. Having everything in the virtual world makes it easier to manage and track. Another advantage of having this inventory process entirely on a computer is that an easy-to-understand user interface (UI) can be used. The final and most important goal of the project is to develop a modifiable, easy-to-use UI. Without a good UI, more time is wasted managing the inventory and defeats the purpose of the project.

The academic site is going to be placed on the same server as the equipment management solution and be run with the same software. Further improvements and upgrades will be made to the site if necessary throughout this process. By consolidating these two features to one server, they contribute to the overall goal of the Web consolidation project.
Solution Details

Users

The users in this system are broken up into two groups. The first group is the Administrator, who is granted full access and privileges to the system. The second group is the student workers, who have the ability to add items to inventory, check in, and check out items. The picture shown in Figure 1 is an early look at what a form in the application may look like.

Figure 1 Sample Inventory Page

The goal of developing the UI is to keep it clean and simple. Although, what you see in figure 1, it only just a piece of that UI.

The use-case diagram in Figure 2 shows both groups of users and what he/she interacts with on the system.
Figure 2 Use Case Diagram

Timeline

Development of this project started in December 2013 and took on another change in January 2014. The following timeline starts from the current project in 2014 shown in figure 3.
**Figure 3 Timeline**

Each week covers a section of the project starting with the core features and working its way to the user interface. There will be three sprints followed by the forth sprint which will be testing and upgrades if everything is on time. Then after development, there will be another round of testing in sections starting with each of the core features and working its way to the user interface.

**Technical Areas**

To approach a solution to these issues, the use of databases and some kind of programming language is going to be needed. There also needs to be either a virtual or physical server to run these two services. The database will have to be robust because a scalable system is crucial with the rapid changes in technology. The database will also have to be able to manage user privileges and security so that there are certain levels of controls put in place. The application itself will need to be written with a language that is able to communicate with a server and can be flexible enough to be modified and improved for future use. The language needs to be able to work side by side with other programming and scripting languages. The
operating systems requirements include being able to run the application and database services, 
transfer between different servers, handle multiple users, and provide sufficient security for 
CECH IT.

**Resources**

**Software**

The software being used for this project needs to hold multiple sites and applications. 
This is why a WordPress content management system is being used. It is easy to set up, very 
flexible when it comes to managing themes and moving systems, uses Hypertext Preprocessor 
(PHP) language, and various scripting languages. WordPress also uses its own database and 
makes it very convenient for development purposes. When developing multiple systems, having 
one easy to manage content management system saves time; (3) due to the short time frame of 
this project, time is very valuable. WordPress also has a feature called plugins. Plugins are like 
applications that you can be used to further manage your Web sites. This project will include the 
development of a plugin to manage these Web sites such as where their database is located, how 
the forms look, and more features will be developed if time will allow. The plugin will be written 
in PHP since it is used by WordPress. The Web sites will be written mostly in PHP when it 
comes to communicating with the server to pull queries and reports for the user. Java Script will 
also play a big part with PHP to query the databases of the system. HTML will be used mainly 
for forms and theme layouts.
Hardware

The environment in which all of these applications will be installed and developed is going to be on one of UC’s virtual machine. This virtual server also called a LAMP or Linux, Apache, MySQL, and PHP will run an operating system called SUSE. The exact version of the operating system is SUSE Enterprise version 11 service pack 3 or 11.3 for short. It is a Linux based operating system and is different from other distributions like Red Hat, Knoppix, and Debian. This current environment is not on a production server meaning that all work that is done in this project needs to be able to transfer to a new environment when completed. Another thing to note about the development environment is that it is on UC’s network. This means that all work either needs to be done while on UC’s network or a VPN or virtual portal network needs to be established. Development of this project is done on 2 machines outside of UC’s network so both are set up with the VPN.

Budget

The actual budget for this project is $0 so far but a theoretical break down of the cost is shown in the table below.
<table>
<thead>
<tr>
<th>Billable Service/Item</th>
<th>QTY</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming Labor Hours</strong></td>
<td>30hrs/week + 16weeks = 480hrs</td>
<td>$43/Hr@480 = $20,640</td>
</tr>
<tr>
<td><strong>Software</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- CMS</td>
<td>1 WordPress</td>
<td>$ 0</td>
</tr>
<tr>
<td>- Database</td>
<td>1 mySQL</td>
<td>0</td>
</tr>
<tr>
<td>- IDE</td>
<td>1 notepad ++</td>
<td>0</td>
</tr>
<tr>
<td><strong>Hardware</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Server License</td>
<td>SUSE Enterprise 11.3</td>
<td>0</td>
</tr>
<tr>
<td>- Physical Server</td>
<td>PowerEdge T20</td>
<td>300</td>
</tr>
<tr>
<td><strong>TOTAL:</strong></td>
<td></td>
<td><strong>$20,940</strong></td>
</tr>
</tbody>
</table>

Figure 4 Budget Chart

The breakdown on the cost is as follows:

- Programming labor hours were calculated from the average yearly salary for a PHP programming developer. (5)

- Software is all open-source therefore free.

- The operating system is also open-source therefore free. The server is not the actual server that UC uses. This was a base server that was picked because it’s only for development and testing. It’s not going to be used for production. The pricing on the project will only include the development server since the production server will not be touched.

The development of all PHP, Java Script, and HTML will be done on a basic IDE called Notepad ++. Transferring code to the virtual server will be done through an FTP program called FileZilla.

The connection actually uses an SFTP connection but FileZilla supports that. Management of the SUSE server will be done through another open-source program called Putty. Putty is a basic command line console program that allows users to control the LAMP server whenever needed.

All of these extra tools are open-source and have no cost to them.
Proof of Concept

The equipment check-in and check-out system was structured where the users have access to different Web pages. The normal users would have access to add items, check in items, and check out items. Normal users have access to all pages that allow for those basic functions. The administrator users have access to all of the basic functions plus the administration console. This console is actually the backend of the entire system and allows the administrator complete control over all Webpages and features. Setting up the structure this way made the process of changing the system much easier. This is why WordPress was one of the best choices for content management. After setting up the basic features as well as the basic part of the Administrator features, the project was presented to the end-users for review before it went forward. After the review, clear goals were put in place for this project to continue. Integration with the existing lightweight direct access protocol was needed. Further modifications to the check in and check out pages were required. The backend system needed to be simplified for the users. This project would fail if the end-users who would play the role of administrators didn’t know how to use the back-end console. The main objective of the project in the second phase is to simplify the back-end system as much as possible. Overall the proof of concept was a good start and did what it was meant to do at that certain point in development. It allowed the development of the project to be shifted a bit to focus on a more important part. That part was user interface design.

Testing

Testing was done after each Sprint and focused on a major part of the system. Most tests started off with the behavior of the Web pages. The first part of testing was the home page behavior. The home page is supposed to accept serial numbers from a user and direct them to the
correct page. Other functionalities of the home page include, resetting session variables, showing the logged in user information about what he/she has checked out, and giving user feedback to an action.

The second part of testing included the add equipment page. This feature does a basic function of adding equipment to the database and nothing more. Testing this page included making sure all affected tables reflected the desired changes when the equipment was added.

The check-out page needed some additional testing after its initial development. This page should add and update specific tables in the database. Along with that, it should allow the user to add associated items to the checkout so that those items stay together throughout the entire process. Additional database testing was also done with the design of this page because of the associated items feature. All items needed an optional group id field which could be updated on-demand by the system. Different scenarios needed to be tested for this part of the project mainly because there were so many things that could go wrong.

The check-in page was tested for its ability to verify the right input was being made and that the right tables were being affected. Based on what was being checked in, the page needed to give user feedback if something was not allowed. This page along with the check-out page is by far the two most important pages of this system, and therefore had a considerable amount of testing time behind it.

The administrator plugin was by far the most complicated part of the project. The parts being tested included the reports being generated from the database, the ability to update and delete equipment from the database, the ability to customize the security and pages being seen through the system, and the management of users.
The security functionality was one of the last things to implement and test. The WordPress CMS allows for the easy implementation of security and that’s why it was last on the list. A plugin to connect to the LDAP security server in CECH IT was implemented along with another plugin to log a user in using Asynchronous JavaScript and XML (AJAX). Both plugins were lightly modified to fit the needs of the system and heavily tested in their ability to identify a valid user, create an account from that user, and tracking the actions for that user.

Conclusion

Throughout the design process of the Equipment Management Check In/Out System, I’ve learned how powerful and convenient WordPress can be. Along with the WordPress CMS simplicity of setting everything up, there is a community of people committed to building plugins and solving issues related to the WordPress platform. This provided a great reference for code and help with some of the walls that stood in the way of progress. The only challenging part when development first started was actually designing the special functioning Web pages. Each page needed to be set up as a template. From there, a new page had to be created on the WordPress admin site with the template. This seems tedious but allows each page to become more manageable when it comes to modifying and changing a feature. Major changes in a feature can be implemented across all pages with relative ease. This accomplishes the goal of allowing end users the ability of easily modifying the system to fit their future needs. The last thing to note is the difference between a Web page and WordPress plugin. Though both may be written with html and PHP, there are more available libraries for the plugins that give them greater capabilities compared to the average Web page. In conclusion, I believe that I’ve created
something that can be easily used, and upgraded for years to come. Most importantly of all, the system works like it should.
References

1. Chris Jesse, A Journey Through Oz (Kendall/Hunt Publishing Company; Dubuque, Iowa, 1997)


