

Case Study “Read and Understood”  
Mantos I.T. Consulting, Inc.  
06 DEC 2013

A small business owner had about 20 people working in headquarters, but had hundreds of employees across the nation who would work weekends for the company. The nature of the business was healthcare related as the company would recruit doctors, nurses, dentists, etc. to help ensure that military service members were ready for duty. The company was all about continually improving the level of service to the clients, but was unsure that its hundreds of non-corporate employees were keeping up with company improvements in practice and policies, which were regularly published on the company’s password- protected web site.

As you might guess, the password protected web site used a small database and a few programs to maintain user accounts and to publish the weekly updates in a blog-like manner. The company also maintained a lot of information on employees such as certifications, location, availability, etc. in one or more corporate databases, including Microsoft Outlook. Although the two databases (the one used by the website and the one used by corporate) contained some common information, such as the employee names and email addresses, corporate personnel had to add, delete, and update records in both databases separately.

The company recognized that it made sense to integrate the web database and the corporate database for at least two reasons. One reason was to be able to add (and delete) usernames and passwords for web site access to weekly updates as employees were hired and terminated in the corporate database. Secondly, corporate people wanted to know whether an employee was up-to-date on recent changes as indicated by whether they had logged in and read the “weekly updates.” Ideally, the company would not send a potential employee on assignment UNLESS that employee was current with respect to the weekly updates.

Although the company was maintaining all this information in one form or another, it had no way to match the two databases to make decisions about who should have logons to access confidential weekly updates, and who should be assigned to upcoming jobs. It was a classic case of the right hand not knowing what the left hand was doing. The sad part was that the company as a whole DID know, but individual employees had to duplicate work already done by another; for example, affecting an email address change. In the case of checking whether a person had read a particular update, that information was available through database logs; but the person scheduling employees had neither the access to those logs, nor time allocated to wade through them. Worse, there was no mechanism to verify that a person assigned to a job was current and knew, for example, that the company was now shredding confidential waste material by placing it in the red plastic tubs maintained on site.

The C.E.O. was adamant that no person would be placed on a job that was not current on every single weekly update since that person was hired. Although the already company generated all the data to do so, simply in the normal course of doing the day’s work, the company had no system to enforce the rule. The edict had no teeth. The staffing people continued to place people who were not “current” on jobs

as they had no way to verify currency, and the boss had no way to check up on the staffers. The answer was simple, share the information. Let the web database know who needed accounts, which was easily determined by the “active” flag maintained by the H/R department in the shared Outlook “Master Contacts” file; and let the corporate database know who had read all the required updates (or not) by setting the “current” flag in the same database.

This is exactly what was done and was done so relatively easily and inexpensively. The reason it was done inexpensively is that the company already had the databases and where already maintaining the information needed. The issue was one of writing a set of small programs that exchanged the data. Technically, linking the two databases was relatively simple as the underlying databases (MySQL, Exchange, and Microsoft SQL) were made accessible on the corporate side using Microsoft Access, a product already used in house. On the web side, the website already used a programming tools (PHP) and a database (MySQL) to maintain the website; so augmenting the program that displayed a particular weekly update (actually, a blog entry) to record in the database that a particular person read a particular update at a certain time was relatively easy.

The website was also updated so that the page that displayed the list of recent weekly updates to a logged-in user could also display each title in green letters if it had been read and red letters if it had not. In this way, the user got to see exactly the same data as the staffing people and was aware whether or not they were eligible for assignment and which weekly updates needed to be read to become so. The page that displayed an update was augmented with a button called “READ AND UNDERSTOOD” which the user needed to press to record that the update had been read, along with instructions on who to call if the update was NOT understood.

Additionally, the ACCESS database was augmented to include a tool to show, for any employee, which updates had and had not been acknowledged, whether that person was “current”, and a tool to set the “current” flag for all employees which was run once a week by corporate staff. Later, the tool was updated to identify anyone scheduled who was not current. That report gave the staffing manager and the C.E.O., the tool needed to enforce the policy and to ensure that the highest levels of quality and care were provided to their clients.

The point is that MANY people use “smart” websites that require a database and which support programming. For example, WordPress is a very popular tool used to construct websites that relies on a MySQL database and the PHP programming language. Why not create simple web-based pages that display customer or user specific information in a secure manner using authenticated logons? In many cases, the information that the user seeks is already available in some corporate databases, or spreadsheet, or 3<sup>rd</sup> party program such as CONTACT, Outlook, Quickbooks, etc. Why take a phone call from the customer and tell them that someone will call them back with the answer? Why not invest in a simple program that offers the user the information they seek right then and there?

If you integrate the website and the corporate database, your people will have less work to do and your clients will value you more. It’s powerful. It’s valuable. It takes less time and money to do so than you might think. Let’s talk about the possibilities. Give Mantos Consulting, Inc. a call at (505) 291-1047.