# DEVELOPMENT OF A PROTOTYPE WEB BASED DECISION SUPPORT TOOL FOR INVENTORY CONTROL IN A COMPUTER REPAIR FACILITY

*Robert D. Spillman, Radford University, rspillma@radford.edu*

*Jeremy Hodges, Radford University, jhodges27@radford.edu*

## ABSTRACT

The focus of this paper is to describe the development of a web-based prototype decision support tool to assist managers and technicians in formalizing a procedure and priority protocol for scheduling computer repairs in a more efficient manner. It was designed for a local branch of a national retailer of personal computers (PC's) and other consumer electronics. The tool was developed in a manner to assist stakeholders determine work flow schedules reflective of the type of repair(s) and customer requests. Among other variables, formalization of an initial scheduling process took into account current PC repair inventory, technician's work schedules, the number of open workstations permitted by the company at a given time, and time estimates for specific repairs. The prototype application produced an increase in technician efficiency and level of throughput. It reduced repair expenditures and included back end code that would allow for its integration into a more comprehensive future application.

## INTRODUCTION

In the current dearth of business resources, many retail businesses and mid level managers are finding themselves increasingly accountable to upper management and the customers they serve. They are expected to produce results and deliver finished products in a timely manner while being held accountable for identifiable levels of quality and adherence to production schedules. In addition, their accountability is increasingly being defined as their ability to produce and maintain a specific level of profitability in proportion to resource utilization. Clearly, profitability can be directly affected by a manager's ability to optimize internal resources.

The validity and importance of process management has been identified extensively in the literature as a critical component in managing and maintaining a successful business (Anupindi, Chopra, Deshmukh, Van Mieghem & Zemel, 2006). Accordingly, development and implementation of the prototype in this project was designed to support the tenets of process management. The design took into account the needs of management and technicians to have access to current data, be able to update data, produce on-demand reports, and incorporate new data into a dynamic database. Creation of the application took into account the existing scheduling structure while identifying critical variables that seemed lacking. Ultimately, implementation of the application and allowing input of computer repair scheduling transactions should improve flow time process.

## PURPOSE

Therefore, the purpose of this project was to create an online scheduling prototype to assist managers and technicians at a local branch of a major electronics retailer to more efficiently manage PC repairs. The application should do the following: provide an end user friendly graphical interface, be intuitive, allow users to leave notations for colleagues, and allow for easy access to future platform maintenance. Later, access to back end code should allow expansion of the application to include prioritizing potential repair jobs reflective of each one's time requirements and profitability.

## NEED

The need for this project was evident to management and technicians who were feeling pressure from each other and from customers to meet projected repair schedules. Tension increased as management and technicians had to explain to customers why jobs were not completed as promised. It was a situation that needed to change.

Lack of communication among technicians as well as with management greatly reduced throughput. Technicians work two separate shifts, and their informal way of communicating the progress of repairs between shifts meant repair tasks were often delayed and sometimes repeated. Because of that, tasks such as virus scanning, reformatting, or even hard disk defragmentation could take hours to complete. Additionally, not properly identifying tasks, a common problem among this type of retailer, created an inability to prioritize the current workflow of inventory. A common online application allowing technicians to post information concerning work in progress was necessary, one that is accessible by management to respond to customer inquiries and apprise technicians of any work order changes.

Evidence of need became even clearer after review of a typical monthly repair schedule. Table 1 indicates representative monthly services, the number of times each service was performed (count), and the average number of days required for each service job to be completed. Two six hour shifts, each with two to three technicians employed, make up a twelve hour technical repair work day. As indicated in the table, nine installations (typically installing software purchased from the retailer, RAM, or hard drives) were performed with each taking an average of one half technical repair work day or six hours per installation. Fifteen diagnostic applications were run and averaged almost nine days each to complete. The remaining data in the table presents over extended time requirements for most typical jobs performed. Company policy states that there should be a maximum of a three day turnaround for repair and return of customers PC's.

| Service | Count | Avg (Days) |
|---|---|---|
| Installs | 9 | 0.56 |
| Diagnostic | 15 | 8.87 |
| Restore | 3 | 4.67 |
| Setup | 6 | 0.67 |
| Other | 3 | 5.33 |
| Virus Removal | 16 | 6.60 |
| Tuneups | 25 | 2.16 |
| Total AVG | | 4.121 |

**Table 1**

Of all repair jobs performed during the sample month, Diagnostics, Restoration, Other Repair, and Virus Removals accounted for almost half of the total repairs performed with the average repair time almost seven days. This is clearly below an acceptable level of minimum performance as prescribed by the parent company. Because of advertising and media exposure, customers have grown to expect a turnaround time of three days or less, so improving repair times will lead to improved customer relations. Figure 1 indicates the number of jobs performed in each category.

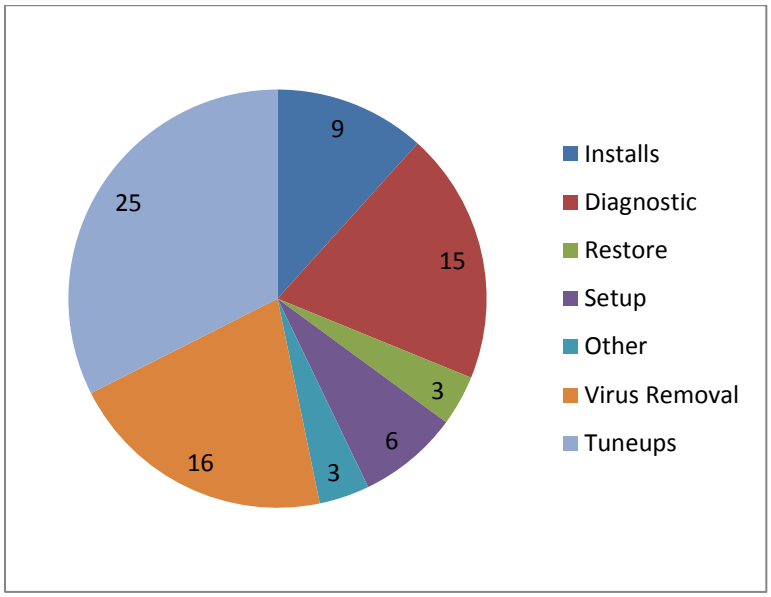


**Figure 1**

Further data review of technician's performance per job indicated job repair time exceeded company policy. There were a total of 77 repair jobs completed in the sample month. On the chart below, the x-axis represents individual work orders, the y-axis represents the amount of days in service, and the horizontal line at tick mark three represents the upper control limit of three days. Any individual repair job shown above the three day limit indicates a failure to adhere to the desired specifications. Approximately 42% of repair jobs extended beyond the acceptable limit.

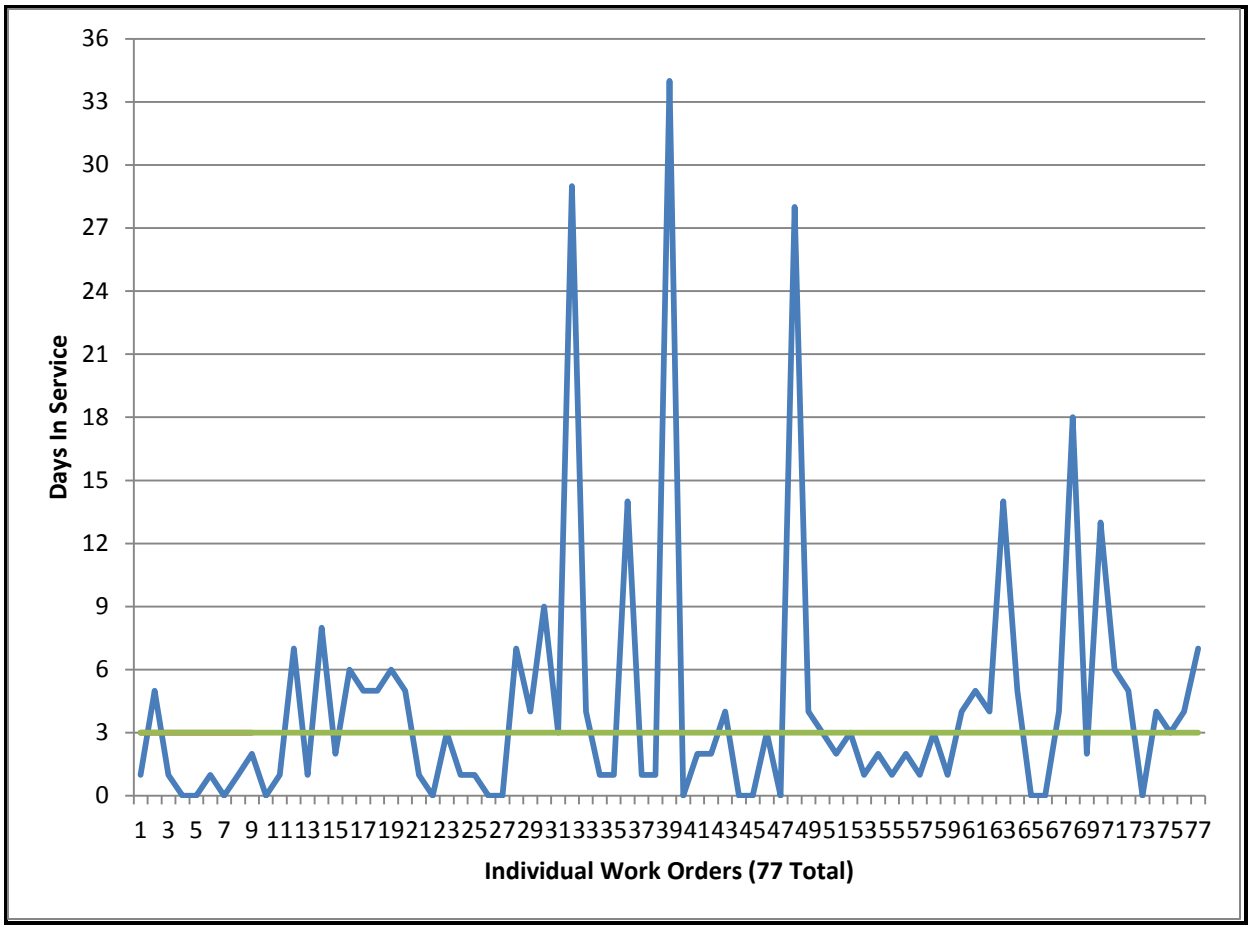


**Figure 2**

## THEORETICAL FRAMEWORK

An adaptation of the Software Engineering Research Methodology (SERM) framework (Gregg et al., 2001) was used in the development of this application. SERM combines traditional software development lifecycles and new technologies resulting in tangible, well-documented, software (Gregg, Kulkarni, & Vinse, 2001, pg. 169). For example, SERM adheres to three tenets of software engineering: a conceptual overview, formal specifications, and the development process itself. Overall, software engineering is a function of all three tenets, but understanding and documenting system requirements is the focal point of the SERM framework. The conceptual overview is followed by mathematical, logical, graphical, and structured natural language requirements and then by the development of the system. For more information on the SERM framework please review (Gregg, et al).

The waterfall software development lifecycle was utilized to provide operational boundaries within the SERM framework and seemed appropriate for three reasons: (1) it provides a structured approach; (2) it progresses linearly through discrete, easily understandable and explainable phases (e.g. planning, requirements and analysis, design, development, and testing); and, (3) it provides easily "visible" milestones in the development process (Sommerville, 2008).

## PROJECT DOCUMENTATION

The developers followed the waterfall software development lifecycle to document and develop the project. This approach provided a framework within which the developers could proceed in a systematic manner. To begin the project development process, an initial Project Planning Document was created, developed, and revised. In the document, project resources, key stakeholders, major risks, and solutions to minimizing risks were identified. This led to an initial prototype. Operational scenarios were examined to include "best", "normal", and "worst" case performances (Turban, 2009). These included scenarios for both managers and technicians. These were developed as pragmatically as possible given identified project constraints.

## PR0JECT DEVELOPMENT

When a customer brings in a PC for repair, the first step at computer "check in" by a technician is to place a bar code on the PC for proper identification. We chose to use this system because it produces a unique customer order number, allows easier identification of the PC in the database, and minimizes data error entry when compared to the current paper based system. An open source freeware program was used to produce typical barcode structures which are read by a USB LED barcode scanner. To create a work order, in conjunction with the barcode number, the technician then obtains additional relevant customer information including the following:

- Order Number (This is captured from the barcode number)
- Customer Information (Name, Address, Contact Information, etc.)
- Customer Comments
- Initial Cost Estimate
- Technicians' Comments – anticipated problem, expected completion

With this type of data entry, users can simply scan the barcode placed on a PC and instantly all notations that have been created by other technicians will be displayed. This will assist in identifying the next steps and should minimize repeating tasks that have been previously performed.

There was a need to develop a workable manager's section to provide options for them to access critical data such as: average technician work time, work in progress, current inventory repair levels, and technician's notations. Managers should be able to add, delete, and update information related to computer repair jobs as well as update human resource data.

### *Graphical User Interface Development*

An important dimension of this project was to create a system that would be intuitive to users. Training future users should be simple and straightforward. With that in mind, a basic layout was created to separate main on-screen user areas. Figure 3 illustrates the initial user login screen for mangers or technicians.

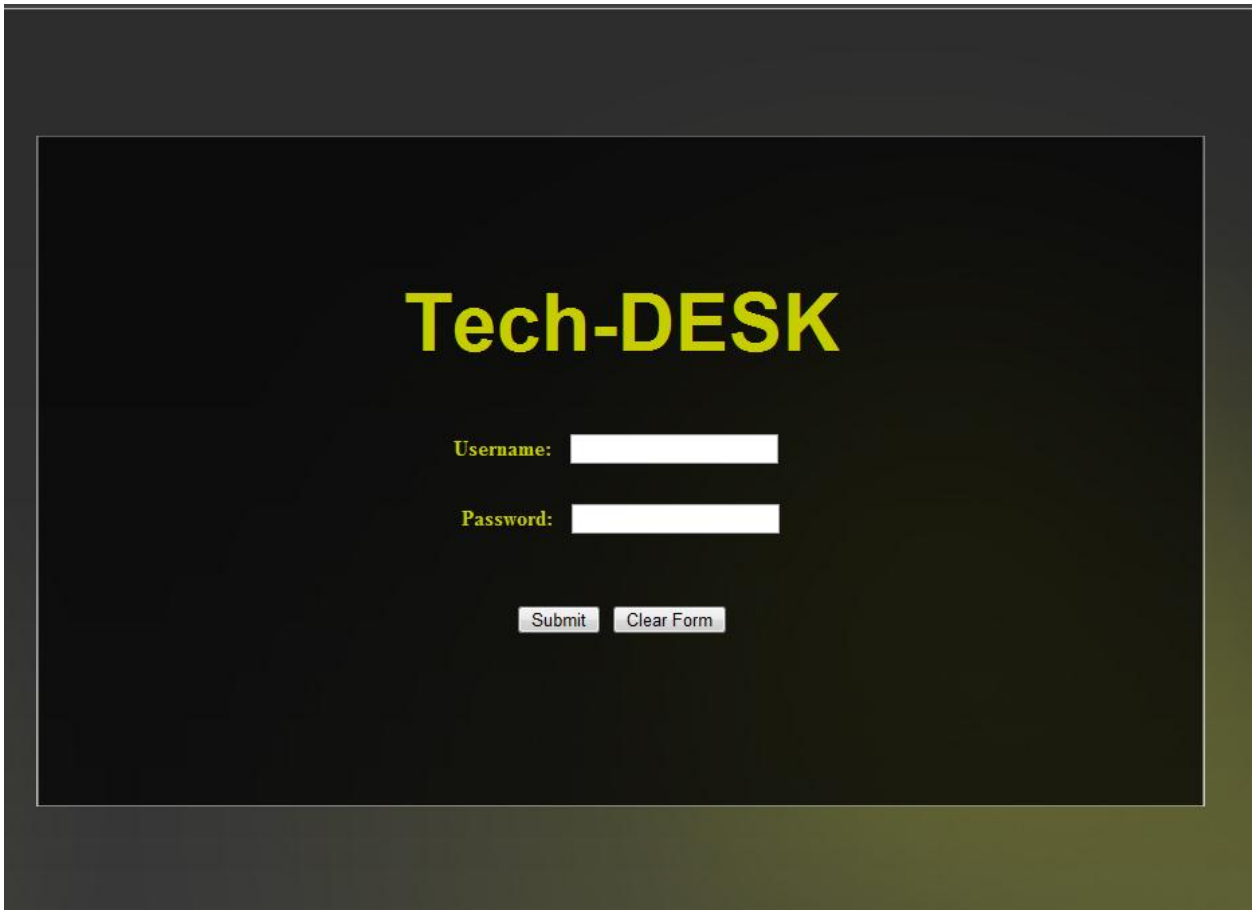


**Figure 3**

After successful login, technicians move to a second screen as shown in Figure 4 allowing them to input data onto a "Notes" page. This open window is separated into three main parts: barcode input, active computers, and the notation area. Users can search for work orders by entering a barcode number into the search window dialog box. In the notation area, they can post comments or instructions for other technicians. The "Active Computers" area provides users with constant visibility of PC repair priority.

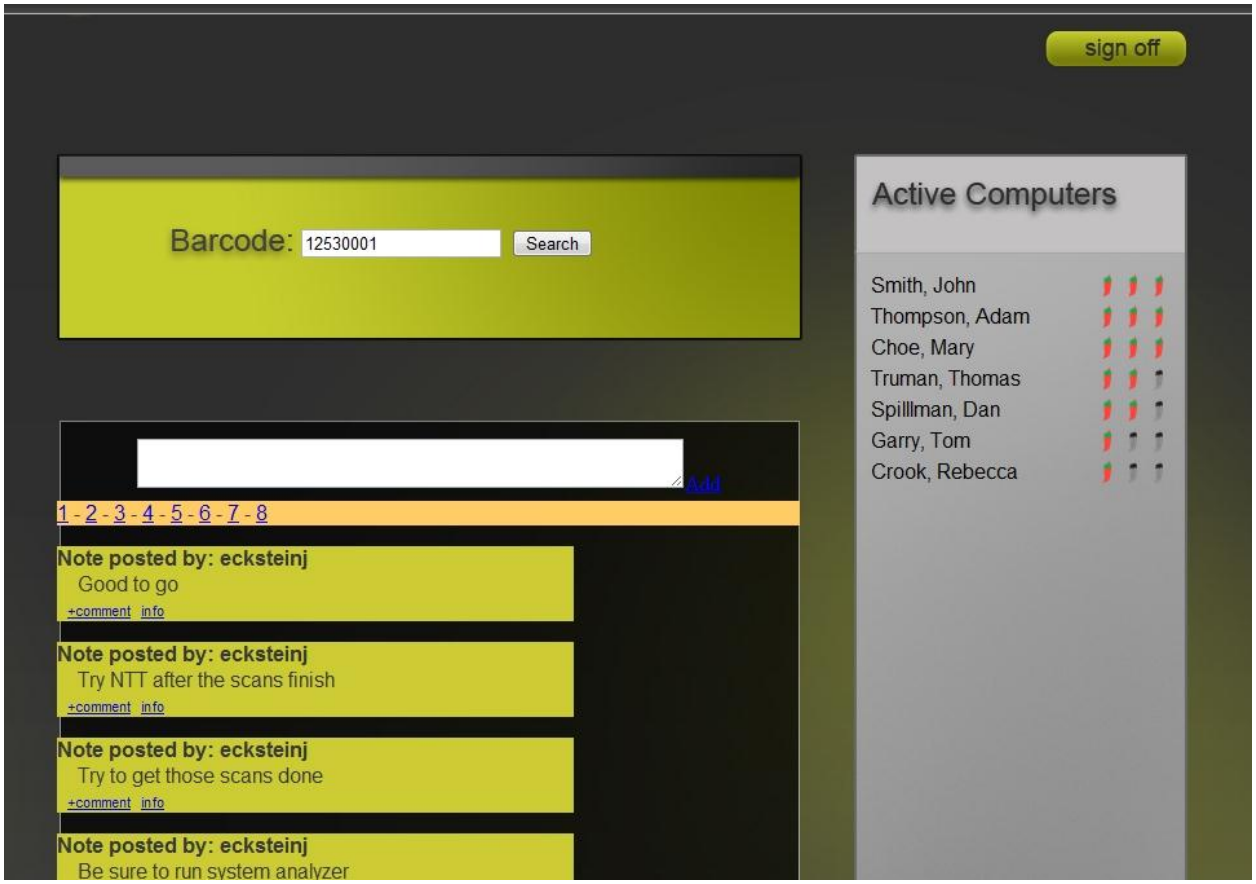


**Figure 4**

Currently, priority is a subjective assessment assigned by technicians. It is indicated by the number of "peppers" associated with each work order to indicate urgency of job completion. Coding in AJAX allows a technician to simply click on the desired priority level and the account will be automatically updated.

Administrator login produces a slightly different opening screen as shown in Figure 5. Their screen is subdivided into three main navigation tabs: Notes, Reports, and Manage Users.

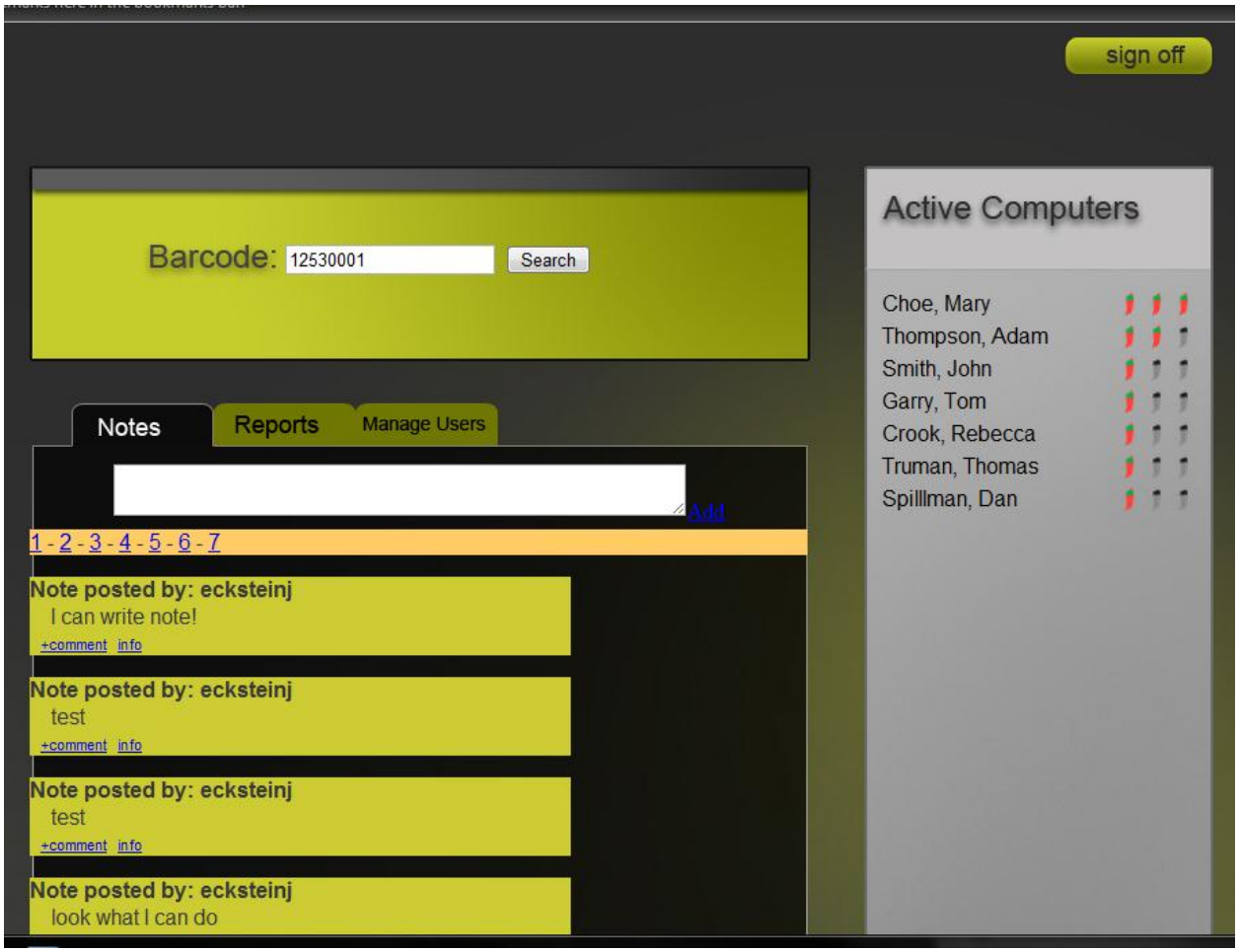


**Figure 5**

They have access to data linked to all three tabs. Selection of the "Reports" tab reveals the screen shown in Figure 6.

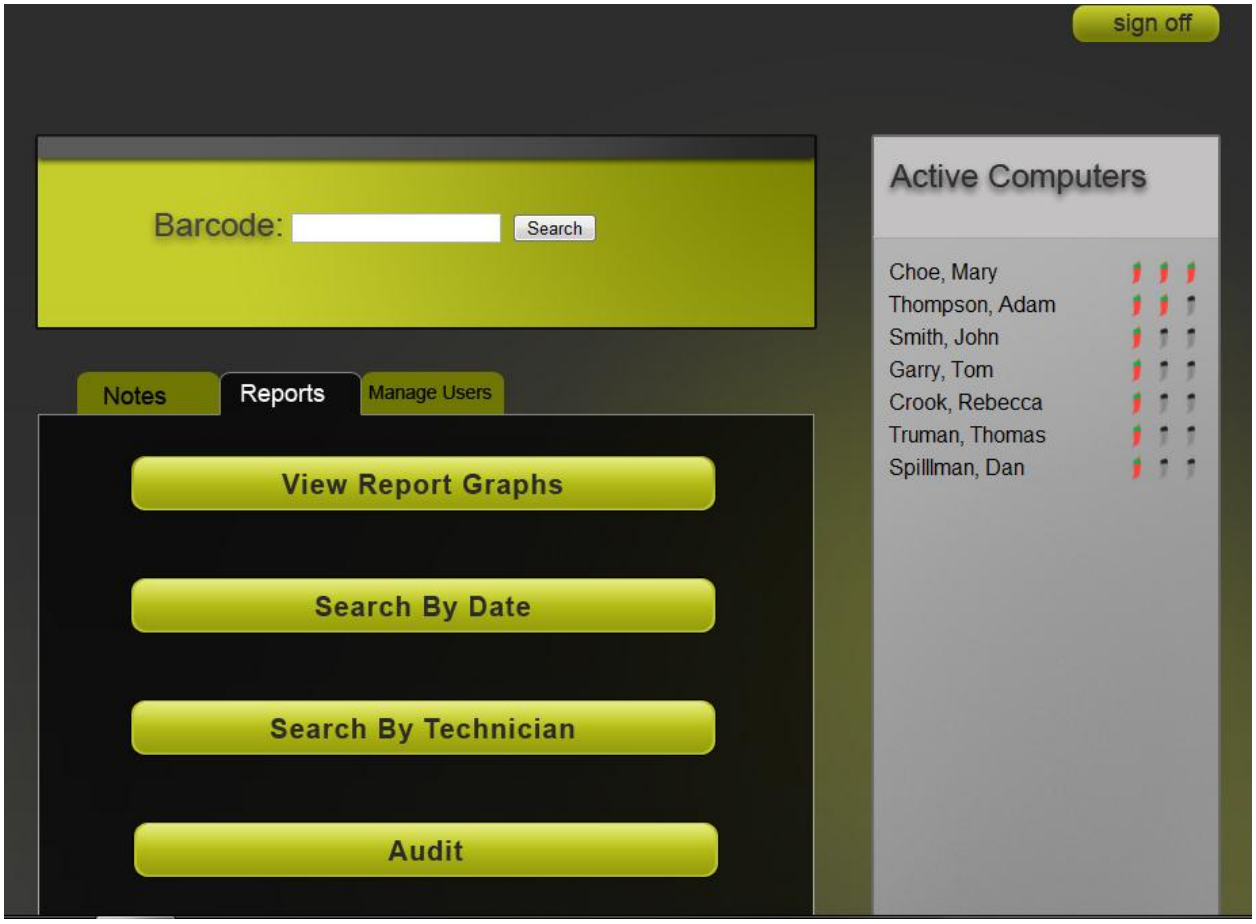


**Figure 6**

This selection allows access to various data views. The "View Report Graphs" option displays a dynamic chart of the current upper control limits per job in progress. "Search by Date" displays notations for a specific date. "Search by Technician" displays notations generated by individual technicians and "Audit" displays an overview of current job repair inventory and work in progress.

The "Manage Users" tab provides managers with access to edit data created by employees using the system. Links within that tab are shown in Figure 7.

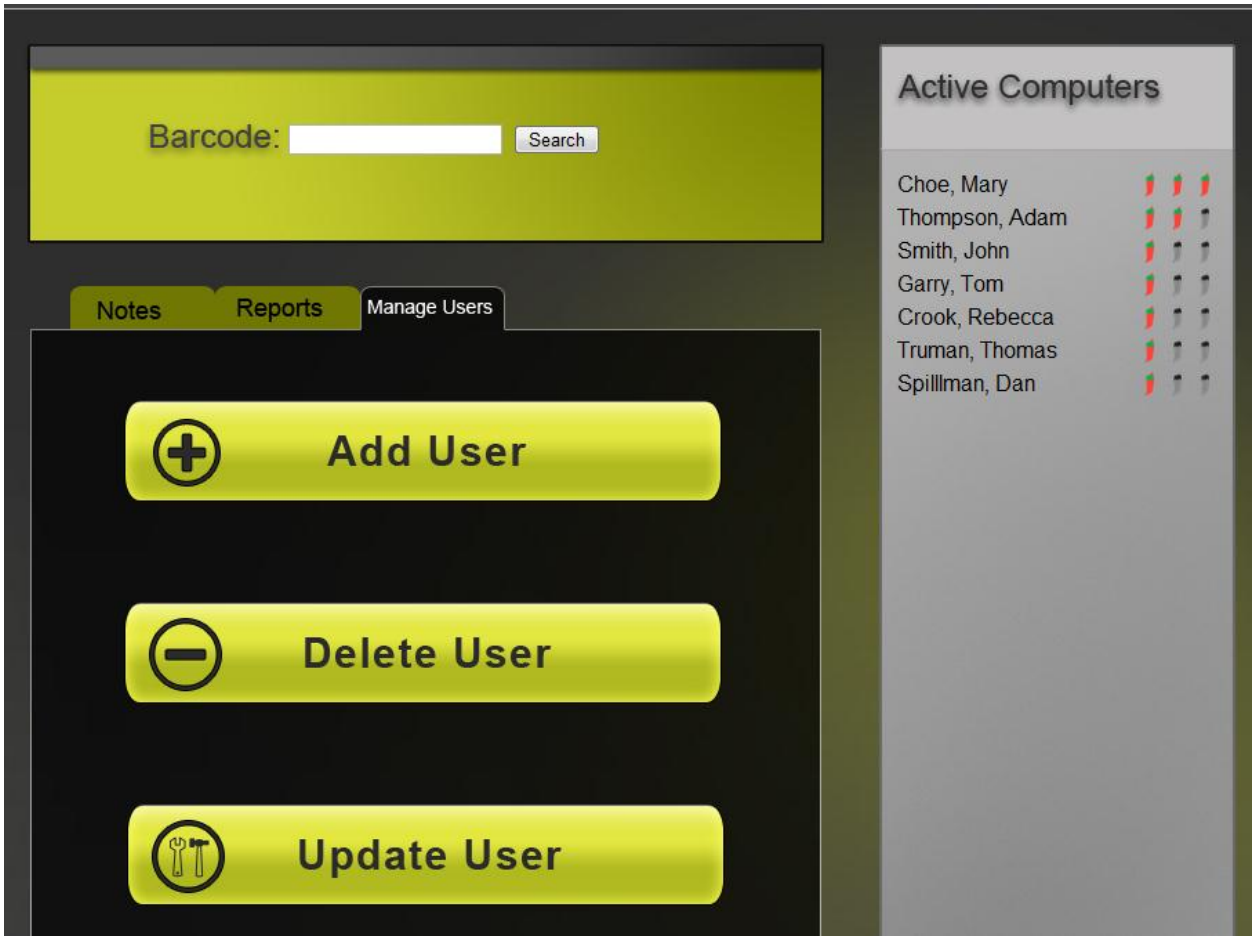


**Figure 7**

To make the GUI more natural to navigate, a "fancy box feature" was added for all buttons selected in the administrator option. This feature creates an open window that appears over top of the current page and displays an area for selected data input. An example follows in Figure 8:

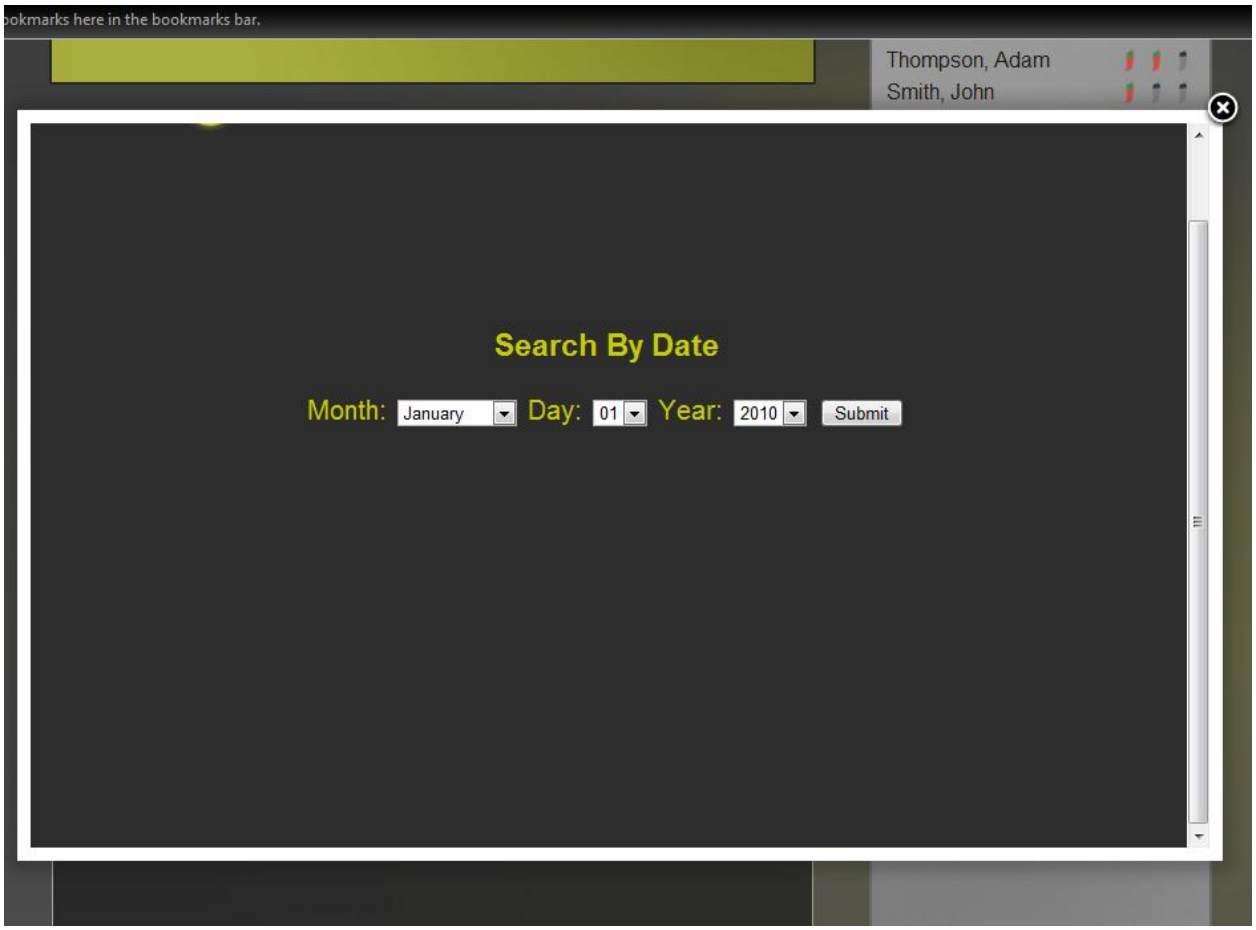


**Figure 8**

Use of the "fancy box" decreases end user confusion while navigating the site. Users are allowed to enter only specified data and can close the box by selecting the 'X' button which will return them to the underlying window.

## SUMMARY

The project was developed to be a prototype for a future system that will maintain a schedule for determining the necessary flow of inventory for a computer repair facility. Upon implementation of the prototype, several concerns have been identified. Some relevant concerns include:

- AJAX allows the website to be dynamic permitting users to click links or other items of interest without having their browser load a new page. This appeared feasible on paper, but when using the application within the administrator options, if the user clicked 'back' or 'refresh' everything would reset. This clearly decreased the functionality of this part of the application. This part of the prototype will need to be recoded to be more in line with other parts of the application which were coded in HTML, CSS, and PHP.

- Originally, the plan was to store completed work orders for 60 days and then back up the data off site. However, within the constraints of management, a feasible way has not been determined to accomplish this task. Currently, completed works are being deleted after 120 days or every quarter. The web files and SQL database is still on a daily backup schedule.

- A code cleanup process has been initiated making the code much more maintainable. Some of the code has been reworked allowing the application to flow better. Variables have been given more meaningful names to facilitate maintenance. Considerable nonessential comment code was also deleted.

As with any prototype, maintenance is essential in correcting deficiencies to create an application that provides a minimal level of acceptable performance. Addressing the concerns above and additional issues as they present themselves should allow for a final application that will allow users to see the long-term effects of shortening flow times and maintaining a more controlled environment.

## REFERENCES


[1] Anupindi, R., Chopra, S., Deshmukh, S., Van Mieghem, J., & Zemel, E. (2006). *Managing business process flows (2nd ed).* Upper Saddle River, NJ: Pearson Education, Inc.

[2] Gregg, D., Kulkarni, U., & Vinze, A. (2001). Understanding the philosophical underpinnings of software engineering research in information systems. Information systems frontier, 2nd special issue: Philosophical reasoning in information. 3(2), 169.

[3] Sommerville, I. (2008). Software engineering (8th ed.). Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

[4] Turban, E., Araonson, J., Liang, T., & Sharda, R. (2007). Decision support and business intelligence systems. Upper Saddle River, NJ: Prentice-Hall Publishing Co., Inc.

[5] Mrs. Ruth Derrick, (Instructor of English, Radford University, Radford, VA). We would like to express our thanks for her editorial review.