INVENTORY TRACKING SOFTWARE IMPROVEMENT PROJECT

A Senior Project submitted
In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Industrial Engineering

the Faculty of California Polytechnic State University,
San Luis Obispo

by
Jacky Leung
June 2016

Graded by:_______________ Date of Submission___________________________
Checked by:_____________ Approved by:______________________________

ABSTRACT

INVENTORY TRACKING SOFTWARE IMPROVEMENT PROJECT

Jacky Leung

Abstract

FanimeCon's Convention Operations department required standardized processes and controls for their inventory tracking software. On top of the current web application, new controls were put into place for checkout limitations. To allow for a continuous improvement effort, processes became standardized with a new graphical user interface where user input is recorded. These will provide processing times for further study in proposing redesigned layouts and more efficient processes.

ACKNOWLEDGMENTS

Thank you Kevin Wang for your support and time. Thank you Dr. Tao Yang for IME 312. Thank you Dr. Jianbiao Pan for IME 430. Thank you mom and dad for supporting my academic endeavour. Thank you for all of the Cal Poly faculty for continuing the learn by doing philosophy.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

I. Introduction

The subject of this report is to clearly show the methodologies used in solving the radio tracking problem provided by the Head of Infrastructure for FanimeCon, Kevin Wang. The problem statement provided by Kevin was the need for an accurate radio tracking system while ensuring a great experience for all staff and clients involved. This senior project will become the fourth iteration of the inventory tracking software.

The methodology that will be used in producing a satisfactory result will include properly defining the problem statement provided by Kevin, gathering as much recorded data as possible from the previous database implementations, gathering survey results, create the software with controls, and generating proposals that will improve the experience for all parties involved.

Since this software is now moving to another department, there will no longer be consistent users of the software. The primary objective is to create an accurate radio inventory tracking software that must be intuitive enough for any user without training to efficiently utilize.

The final deliverable of this project will be to provide an inventory web application using PHP and MySQL which will track radios, track headsets, have access controls, and basic reporting. In this program, access controls is defined as checkout limitations for each department.

The course that is most relevant to the deliverable of the web application is IME 312, Data Management and System Design. The content learned in this course taught data collection and retrieval in a methodical way. IME 430, Quality Engineering, brought exposure to control charts, which is an extremely useful tool for analyzing datasets. Although unused in this senior project, IME 443 taught students how to creatively develop facility improvement efforts while IME 420 taught simulations for the facility designs. IME 443 and 420 gave insights as to what data should be collected so that these courses may be used in future improvement efforts.

This report is organized in such a way that the background portion covers all of the necessary literature review that has influenced the decisions. The implementation design refers to the previous designs and how this new version of the web application will be implemented. Code design follows immediately after and is completely technical. This portion makes up the bulk of the report and contains snippets of both the front and back end parts of the web application. Therefore, the code design is split up into their own segments which include: checkouts, returns, history, advanced, and web analytics. This is followed by the results and conclusions portion of the project which includes ideas for improvement.

From here on out, a “user” is an individual who interacts with the checkout web application; a “staff” is an individual who is on the receiving end of the radio/accessories.

II. Background (includes Literature Review)

FanimeCon is a convention in San Jose that will occur from May 27th to May 30th this year. It is a convention that promotes Japanese media and is primarily ran by volunteers. When I reached out to the Convention Operations department to volunteer, I had the opportunity to use their new Python-based radio tracking system. While I was using the system, I had noticed that the radios that were being checked out were never returned, which means the system was unreliable. Later that year, IME 312 with Dr. Yang was taken and database solutions were learned. Since the course offered a similar project, Kevin Wang was approached and a MS Access version of the radio tracking system was created.

Originally, the inventory tracking process used pen and paper to track radios being checked out and returned. This was digitalized by Kevin through an implementation of a Python-based program. Since entries were inputted through command line, it was quite common to see the staff members become paralyzed when met with a blinking white bar asking for a text input. The second iteration of the program was created using MS Access to give staff a graphical user interface to avoid errors during entry. A third iteration was created in PHP and MySQL, providing a server-based database to eliminate potential data loss. Although the program was only used in the Convention Operations department, the program can be expanded for implementation in the Registration and Rovers departments. Currently, the system is expected to move into another department, which means that it is necessary to create an intuitive web application that does not require much oversight or training.

For the development and layout of the web application, touch screen was decided over keyboard and mouse inputs based on the research done by Karat, McDonald, and Anderson (1986). Figure 3 (Karat, McDonald, Anderson, 1986, p. 80) and Figure 7 (ibid, p. 86) shows that the menu selection speed was the quickest through touch. Given the choice, “The current study supports the statement that when comparing these devices in a menu-selection task, the touch panel is superior” (Karat, McDonald, Anderson, 1986, p. 88). Therefore, a touch screen device is assumed to be used for the development and implementation of this web application. The device being used to test the prototype of the web application is a Microsoft Surface Pro 4.

For the button design, the decision was to have “bold for label text, use positive polarity (i.e., dark text on a lighter background), have rounded corners...” (Bangor, Miller, 2005, p. 1450-1451). At a 95% confidence interval, the study concluded that these were the traits of a button that “participants overwhelmingly preferred” (ibid, p. 1451).

Suto, Sakamoto, and Okita (2009) conducted a study on the effects of button coloration on electronic devices. The study had concluded that “Blue panels are better than red ones” and “Formal colorations are better than other ones at encouraging an operator to answer calmly” (Suto, Sakamoto, Okita, 2009, p. 203).

It was pointed out by Kevin that due to a variable passing feature, the third iteration of the program was subject to SQL injection. This feature will be removed in the fourth iteration. The remaining SQL statements have been subject to “prepared statements” per the manual on php.net. The document states: “If an application exclusively uses prepared statements, the developer can be sure that no SQL injection will occur (however, if other portions of the query are being built up with unescaped input, SQL injection is still possible)."

Due to the lack of readily available data records, a new component known as *Web Analytics* will be implemented for the 2016 version of the software. Kumar, Singh, and Kuar (2012) defines that web analytics "deals with the methods for measurement, data collection, data analysis and providing the related feedback on internet for the motive of understanding behavior of the customer using website. The benefit of studying behavior of the customer leads to optimize the usage of web site" (Kumar, Singh, Kuar, 2012, p. 966). The "three fundamental pillars" are "data collection, data storage and data evaluation" with the needs being "measuring the performance of the website against targets...gaining clarity on user behaviors and needs, and how the site is meeting those needs…" and the "...ability to take proper action to improve the website based on the results" (ibid, p. 966-967). The data collection component will provide processing times and idle times that can be analyzed to further improve the software and processes.

Part of the process includes entry of staff information into the database of the web application. Previously, this was manually added when a staff checks out a radio for the first time that year. The solution to this introduces phpmyadmin, which is a tool used to manage the MySQL database. Per "http://docs.phpmyadmin.net/en/latest/import_export.html", a CSV file is able to be imported directly into the database without manually adding each staff.

The usage of Web Analytics in association with the storage of staff's personal information means that data privacy must be dealt with. Under section 4.2 of *Data Privacy in Web Analytics*, a "proposed certification" (Zumstein, Drobnjak, Meier, 2011, p. 469) provides seven criteria that gives the user detailed information about the data collection and control over their collected data. This program will be achieving a "Silver Certification," (ibid, p. 470) which does not allow opting out or deletion of user data.

III. Implementation Design

The new layout of the 2016 web application will assume that the platform being used is a touchscreen tablet with a USB port. This allows for full functionality of the web application which has been developed for a touch interface and a barcode scanner.

New requirements from Kevin include an inventory control system with access control and an additional component for headset checkouts. An inventory control system is defined as a special return page that ensures, on the last day, all of the radios in the inventory are accounted for. Access control is defined as the limited amount of radios a single department can have checked out at a time.

The previous version of the web application required new radio users to be added to the database upon the first time they check out a radio. A feature was added to allow the checkout to process either adding the staff or checking out the radio first; this feature created the opportunity for a potential SQL injection attack. Therefore, this feature is now removed. In addition, the database will be prepopulated with the department staff (rather than adding the staff upon their first checkout) using a data sheet from the registration department.

Using the recorded data from last year, timestamps from very specific events with inaccurate assumptions was graphed. See Figure A. Because of this unreliable data collection method, it was decided to include a *Web Analytics* data collection component to the web application.

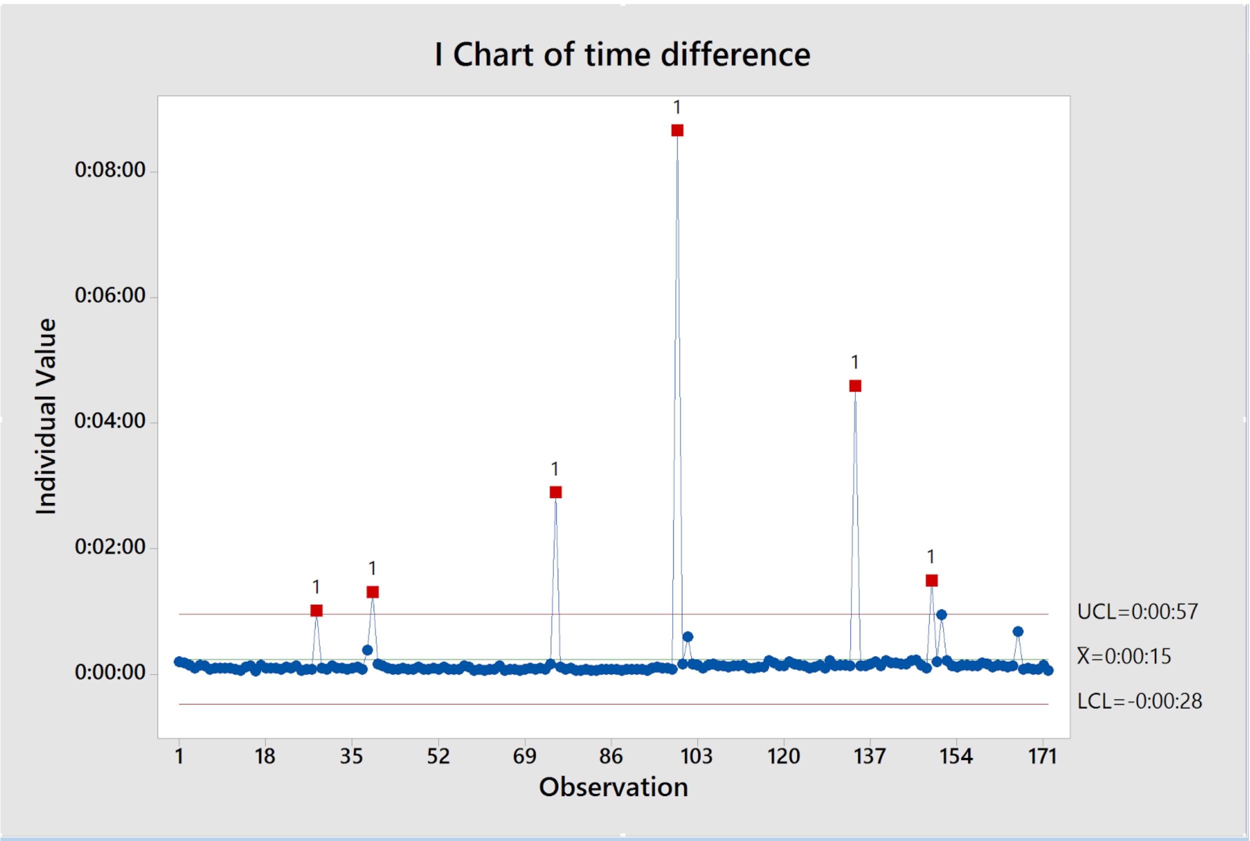


Fig. A - Adding Radios to Database

Using a few of the articles found on website design, the web application is designed to have a more user-friendly interface. The newly created buttons also have a new component to them, known as *Web Analytics* (Kumar, Singh, Kuar, 2012, p. 966). Once pressed, the event identifier and the timestamp is recorded in a table. This was included because there was a lack of readily available data for facility redesign and process control. By subtracting timestamps of initiation from completion of the process, process times are obtained which can be graphed in an I-chart. Errors in user entry can also be recorded by analyzing the sequence for which the buttons are pressed.

This web application has a lifecycle of at least one hundred and four hours, and changes are not recommended once implemented. Only code fixes under dire circumstances are reasonable to prevent loss of data integrity. Otherwise, all code or process additions and improvements will have to be made after this event for the next year.

Since the previous version of the web application did not record much historical data, a survey was sent out which had inquired for improvements from last year’s staff who had used the inventory tracking software.

IV. Code Design

The following sections will highlight the different components of the software (web pages) that perform their own individual functions. This will include: Checkout, Return, Checked Out, Return History, and Web Analytics.

When the user reaches the front page of the website, they will be met with the image shown in Figure 1. The intention is to provide the user with a dichotomy that the radio and headset is either going to be checked out or returned.

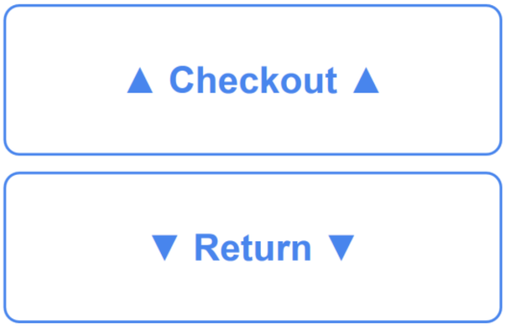


Fig. 1 - Index Page

Once the user touches the “Checkout” button for example, the colors invert, which provides the user with the feedback that the button has been registered. This is shown in Figure 2.

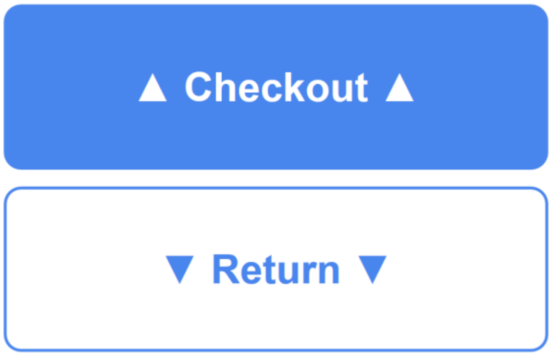
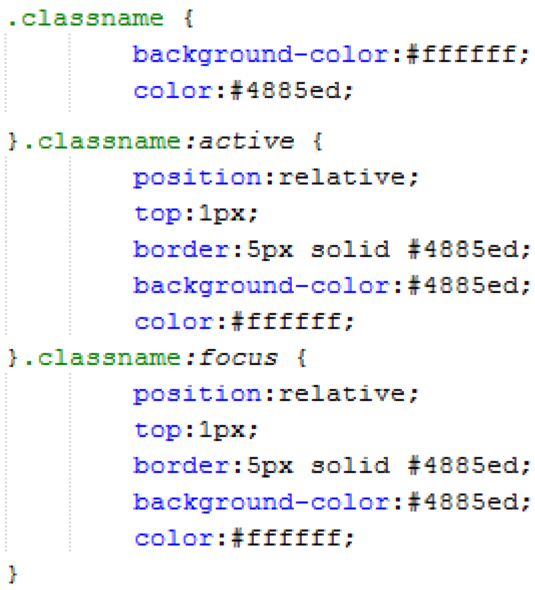


Fig. 2 - Highlighted Button

Figure 3 shows the relevant CSS (Cascading Style Sheets) code for both the “Checkout” and “Return” buttons that express this logic.



```
.classname {
        background-color:#ffffff;
        color:#4885ed;
}.classname:active {
        position:relative;
        top:1px;
        border:5px solid #4885ed;
        background-color:#4885ed;
        color:#ffffff;
}.classname:focus {
        position:relative;
        top:1px;
        border:5px solid #4885ed;
        background-color:#4885ed;
        color:#ffffff;
}
```

Fig. 3 - Checkout and Return Button CSS

A. Checking Out

After “Checkout” is pressed, we are met with Figure 4. This web form allows the user to scan in the Badge ID, Radio ID, and optionally the Headset ID. Figure 5 shows the relevant HTML and Javascript code for this page.

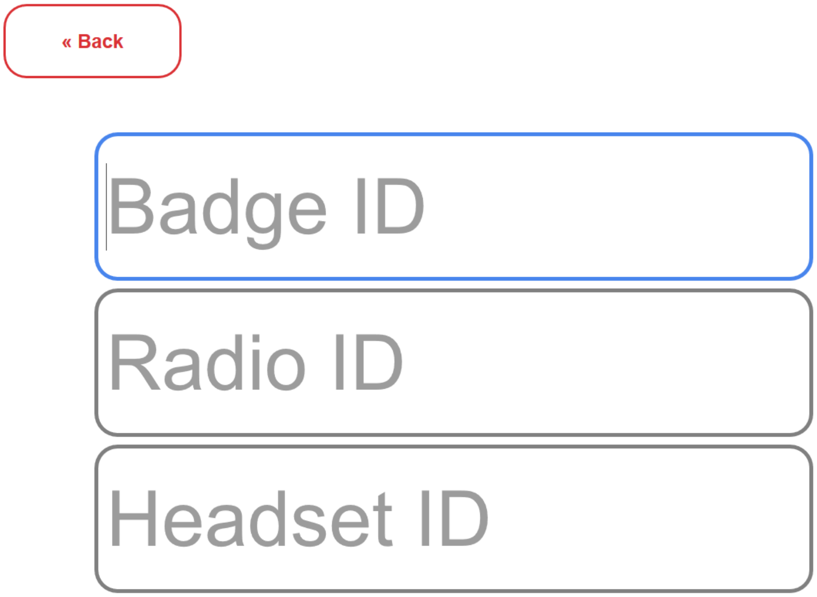
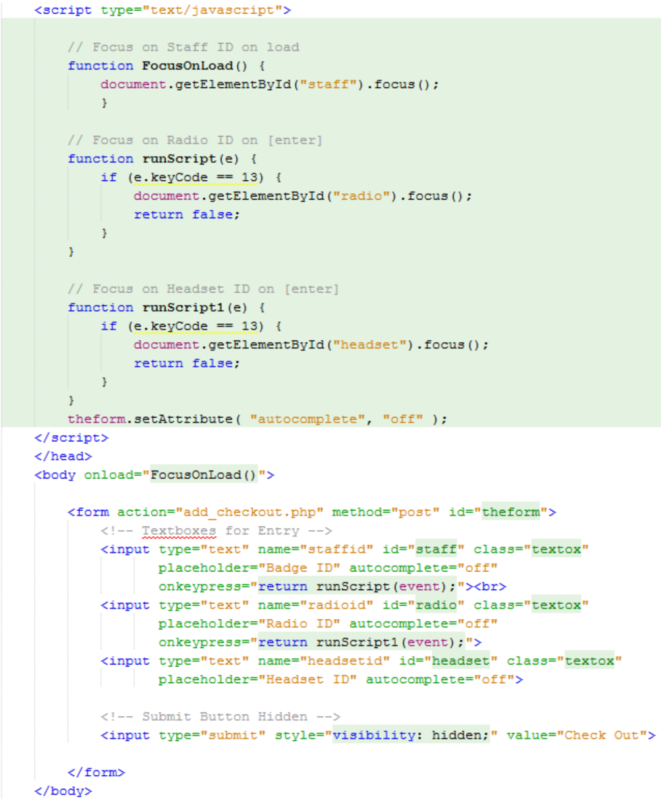


Fig. 4 - Checkout Textboxes

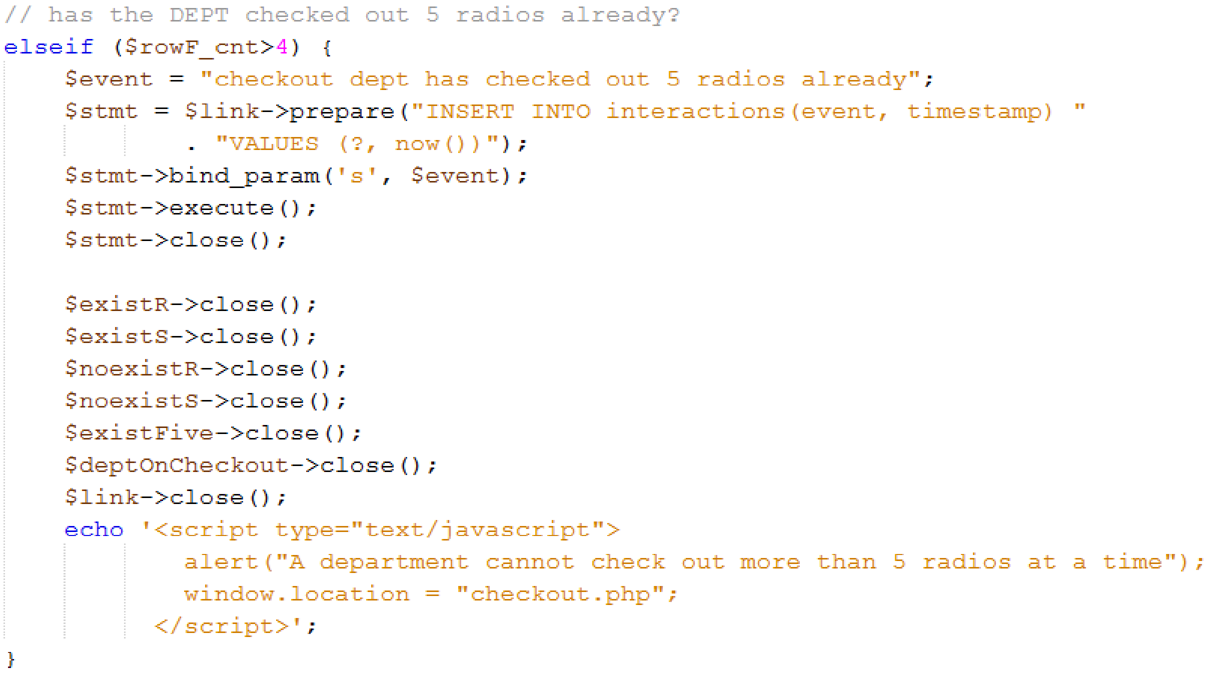


```
<script type="text/javascript">

    // Focus on Staff ID on load
    function FocusOnLoad() {
        document.getElementById("staff").focus();
        }

    // Focus on Radio ID on [enter]
    function runScript(e) {
        if (e.keyCode == 13) {
            document.getElementById("radio").focus();
            return false;
        }
    }

    // Focus on Headset ID on [enter]
    function runScript1(e) {
        if (e.keyCode == 13) {
            document.getElementById("headset").focus();
            return false;
        }
    }
    theform.setAttribute( "autocomplete", "off" );
</script>
</head>
<body onload="FocusOnLoad()">

    <form action="add_checkout.php" method="post" id="theform">
        <!-- Textboxes for Entry -->
        <input type="text" name="staffid" id="staff" class="textox"
               placeholder="Badge ID" autocomplete="off"
               onkeypress="return runScript(event);"><br>
        <input type="text" name="radioid" id="radio" class="textox"
               placeholder="Radio ID" autocomplete="off"
               onkeypress="return runScript1(event);">
        <input type="text" name="headsetid" id="headset" class="textox"
               placeholder="Headset ID" autocomplete="off">

        <!-- Submit Button Hidden -->
        <input type="submit" style="visibility: hidden;" value="Check Out">

    </form>
</body>
```

Fig. 5 - Checkout Page Code

This checkout screen already highlights the textbox for Badge ID and is ready for the barcode to be scanned. The barcode scanner is also programmed to press “Enter” after the barcode is scanned. Using this functionality, the program reads the “Enter” key as the signal to highlight the next textbox, Radio ID. Once the barcode scanner reads in the Radio ID, the program once again jumps to the next textbox and highlights the Headset ID. Not every user will be checking out a headset, so we will have a barcode with value “0” that can be scanned if the user does not check out a headset. Otherwise, the Headset ID will be scanned in and “Enter” is registered again by the barcode scanner. Upon pressing “Enter” while the Headset ID field is highlighted, the program will submit all of this data into the “Add_Checkout.php” page.

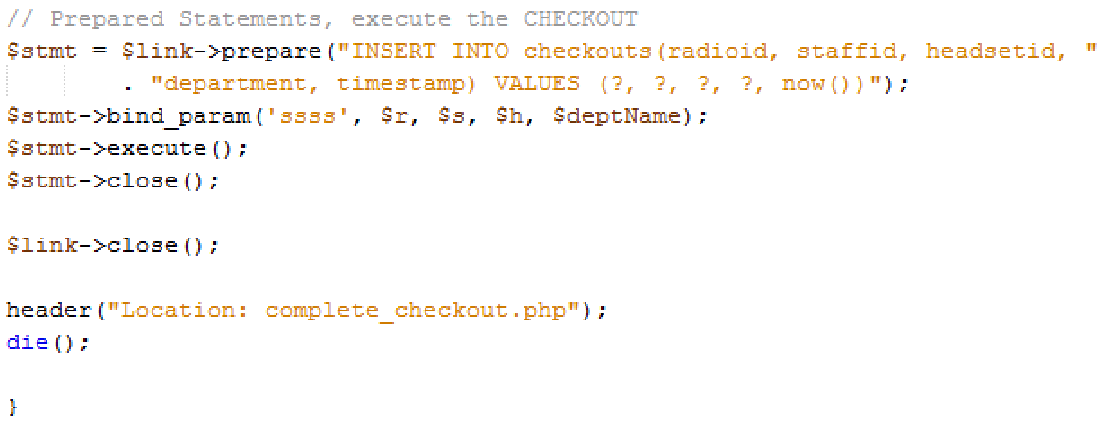
While we would like to assume that mistakes will not be made, it is good practice to check for errors when the user inputs data. In addition, a new requirement for 2016 is to only allow for a maximum of 5 radios to be checked out for each department. Figure 6 shows the an example from the backend PHP in “Add_Checkout.php” that looks for the following errors: Radio ID textbox is blank, Badge ID textbox is blank, radio is already checked out, a department has checked out 5 radios already, radio does not exist in the database, and staff does not exist in the database. If one of these conditions are met, a Javascript popup will show the error message and redirect the user to the Checkout or Return page for data re-entry or radio return without processing the current checkout or return. Figure 6 shows the relevant code for the radio checkout maximum for each department.



```
// has the DEPT checked out 5 radios already?
elseif ($rowF_cnt>4) {
    $event = "checkout dept has checked out 5 radios already";
    $stmt = $link->prepare("INSERT INTO interactions(event, timestamp) "
            . "VALUES (?, now())");
    $stmt->bind_param('s', $event);
    $stmt->execute();
    $stmt->close();

    $existR->close();
    $existS->close();
    $noexistR->close();
    $noexistS->close();
    $existFive->close();
    $deptOnCheckout->close();
    $link->close();
    echo '<script type="text/javascript">
            alert("A department cannot check out more than 5 radios at a time");
            window.location = "checkout.php";
          </script>';
}
```

Fig. 6 - Error Checking on Checkout

If there are no errors, then the checkout will execute and the page will be redirected to the record keeping script, followed by a redirect to the front page; the code is shown in Figure 7.



```
// Prepared Statements, execute the CHECKOUT
$stmt = $link->prepare("INSERT INTO checkouts(radioid, staffid, headsetid, "
        . "department, timestamp) VALUES (?, ?, ?, ?, now())");
$stmt->bind_param('ssss', $r, $s, $h, $deptName);
$stmt->execute();
$stmt->close();

$link->close();

header("Location: complete_checkout.php");
die();

}
```

Fig. 7 - Insert Checkout to Table

B. Returning

A similar procedural structure exists for returning a radio. The staff's Badge ID is no longer necessary since we can simply look for the Radio ID in the database. As a record, we have another table called "history," which tracks all the returns. Figure 8 shows the user page and Figure 9 shows the error checking piece of the code after submitting the input.

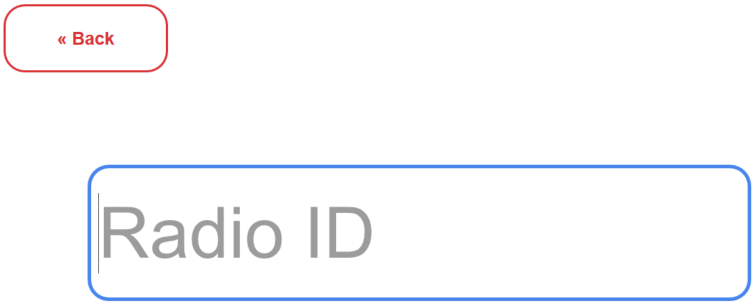
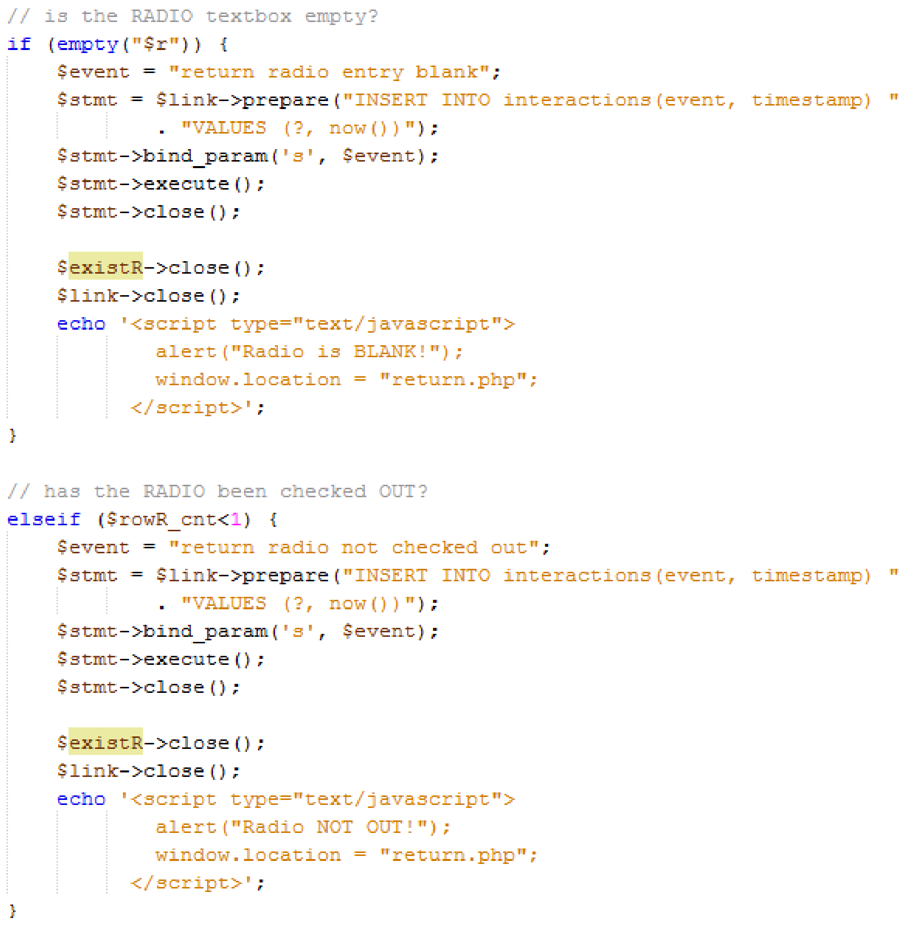


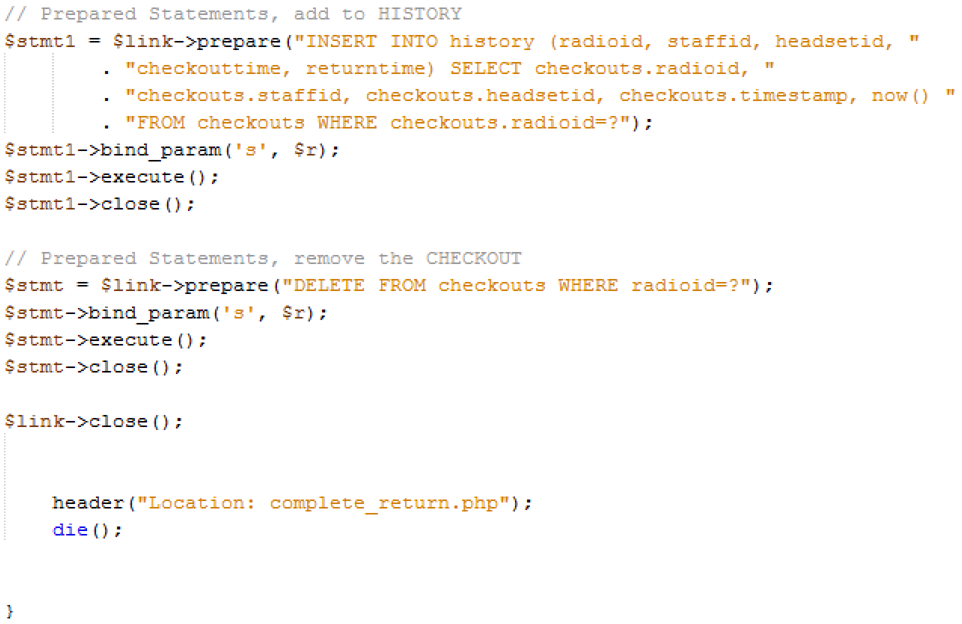
Fig. 8 - Return Textbox



```
// is the RADIO textbox empty?
if (empty("$r")) {
    $event = "return radio entry blank";
    $stmt = $link->prepare("INSERT INTO interactions(event, timestamp) "
            . "VALUES (?, now())");
    $stmt->bind_param('s', $event);
    $stmt->execute();
    $stmt->close();

    $existR->close();
    $link->close();
    echo '<script type="text/javascript">
            alert("Radio is BLANK!");
            window.location = "return.php";
          </script>';
}

// has the RADIO been checked OUT?
elseif ($rowR_cnt<1) {
    $event = "return radio not checked out";
    $stmt = $link->prepare("INSERT INTO interactions(event, timestamp) "
            . "VALUES (?, now())");
    $stmt->bind_param('s', $event);
    $stmt->execute();
    $stmt->close();

    $existR->close();
    $link->close();
    echo '<script type="text/javascript">
            alert("Radio NOT OUT!");
            window.location = "return.php";
          </script>';
}
```

Fig. 9 - Error Checking on Return

Again, we are checking for a blank entry made by the user followed by checking whether the radio had been checked out in the first place. We then copy over the data from the checkout table into the history table and include an additional timestamp for the return time. The checkout is then removed from the checkouts table. See Figure 10.



```
// Prepared Statements, add to HISTORY
$stmt1 = $link->prepare("INSERT INTO history (radioid, staffid, headsetid, "
        . "checkouttime, returntime) SELECT checkouts.radioid, "
        . "checkouts.staffid, checkouts.headsetid, checkouts.timestamp, now() "
        . "FROM checkouts WHERE checkouts.radioid=?");
$stmt1->bind_param('s', $r);
$stmt1->execute();
$stmt1->close();

// Prepared Statements, remove the CHECKOUT
$stmt = $link->prepare("DELETE FROM checkouts WHERE radioid=?");
$stmt->bind_param('s', $r);
$stmt->execute();
$stmt->close();

$link->close();


    header("Location: complete_return.php");
    die();


}
```

Fig. 10 - Insert Return to Table

C. History

To keep the user of the database informed of the current status of all radios, two history buttons exist immediately below the “Checkout” and “Return” buttons. See Figure 11.

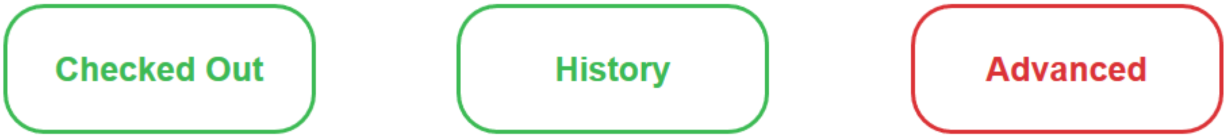
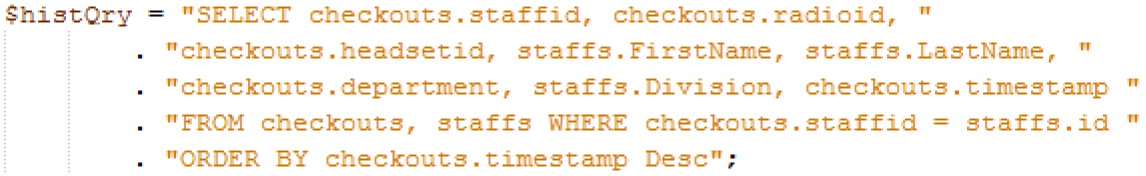


Fig. 11 - Miscellaneous Buttons

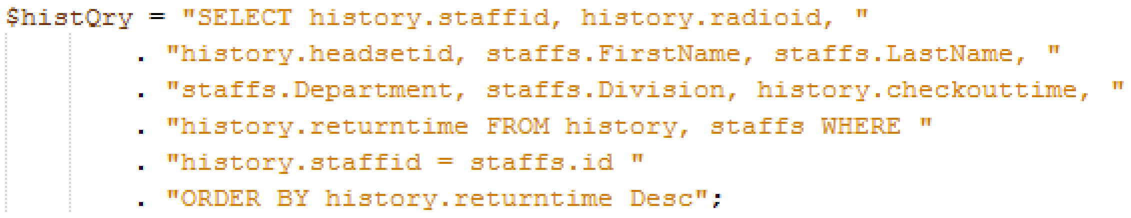
The “Checked Out” button on the far left of Figure 11 gives the user a list of radios (along with an optional headset) currently checked out from the inventory using the query shown in Figure 12.



```
$histQry = "SELECT checkouts.staffid, checkouts.radioid, "
        . "checkouts.headsetid, staffs.FirstName, staffs.LastName, "
        . "checkouts.department, staffs.Division, checkouts.timestamp "
        . "FROM checkouts, staffs WHERE checkouts.staffid = staffs.id "
        . "ORDER BY checkouts.timestamp Desc";
```

Fig. 12 - Checked Out History Query

The center button titled “History” in Figure 11 lists all radios that have been returned and is used to tracked which staff had checked out which radio and/or headset. The query is shown in Figure 13.



```
$histQry = "SELECT history.staffid, history.radioid, "
        . "history.headsetid, staffs.FirstName, staffs.LastName, "
        . "staffs.Department, staffs.Division, history.checkouttime, "
        . "history.returntime FROM history, staffs WHERE "
        . "history.staffid = staffs.id "
        . "ORDER BY history.returntime Desc";
```

Fig. 13 - Returned History Query

D. Advanced

On the far right of Figure 11 is the “Advanced” button. This button is extremely critical to the integrity of the database and should only be accessed when special circumstances arise. In actual implementation, the user did not know of its existence and was not used unless either the author or Kevin Wang was present to show the functionalities. This button was not password protected since there wasn’t consistent users. Figure 14 shows the buttons within the Advanced menu. In the future, these buttons will be removed.

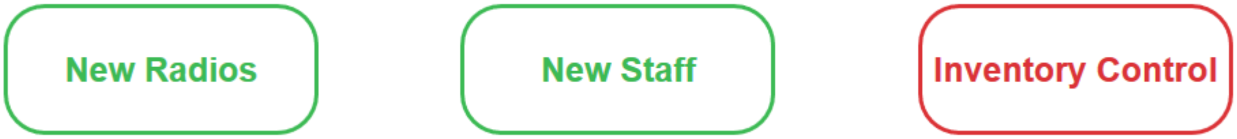


Fig. 14 - Advanced Menu

“New Radios” provides the user with a textbox to scan in all of the new radios prior to having the web application go live. This populates the database with radio identifiers that a staff member can check out.

“New Staff” provides the user with a list of textboxes to add a staff that does not exist in the database. This is typically used for hired security guards or San Jose police officers. The form is shown in Figure 15.

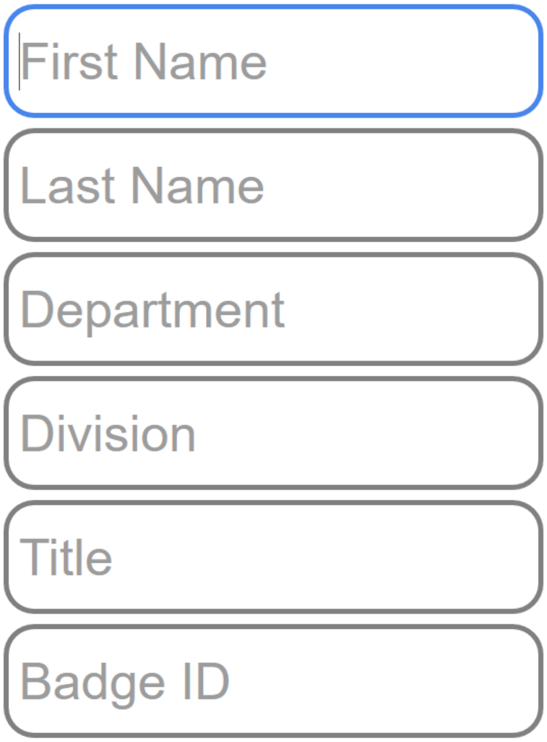
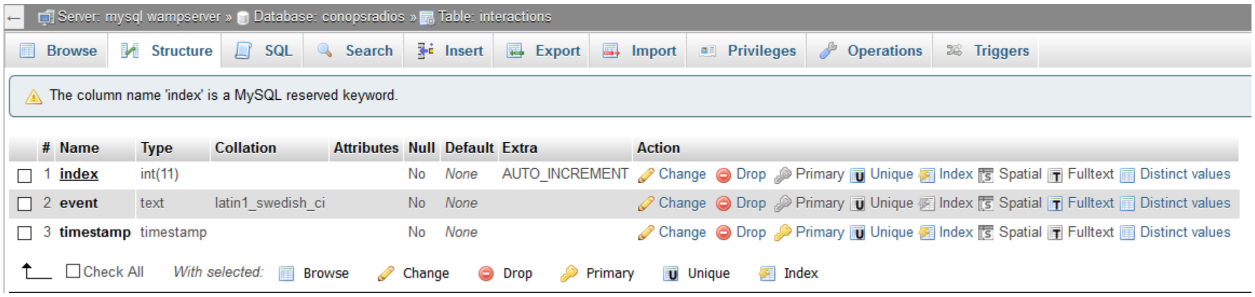


Fig. 15 - New Staff Textboxes

“Inventory Control” is built quite similar to the radio return form. The primary usage for this radio return form is to double check all of the radios in inventory at the end of the day. There are two main differences with the Inventory Control form and the Return form. Inventory Control will not redirect the user back to the “index.php” page; it will process the radio identifier and prompt for more radios to be scanned in. In addition, the Inventory Control form will not check if the radio has already been returned; if a radio has not been returned, it will be returned through this form.

E. Web Analytics

The solution to this problem statement will also allow for further improvements by collecting valuable data relevant to the usage of the software; aka Web Analytics. The table schema for data collection is shown in Figure 16.

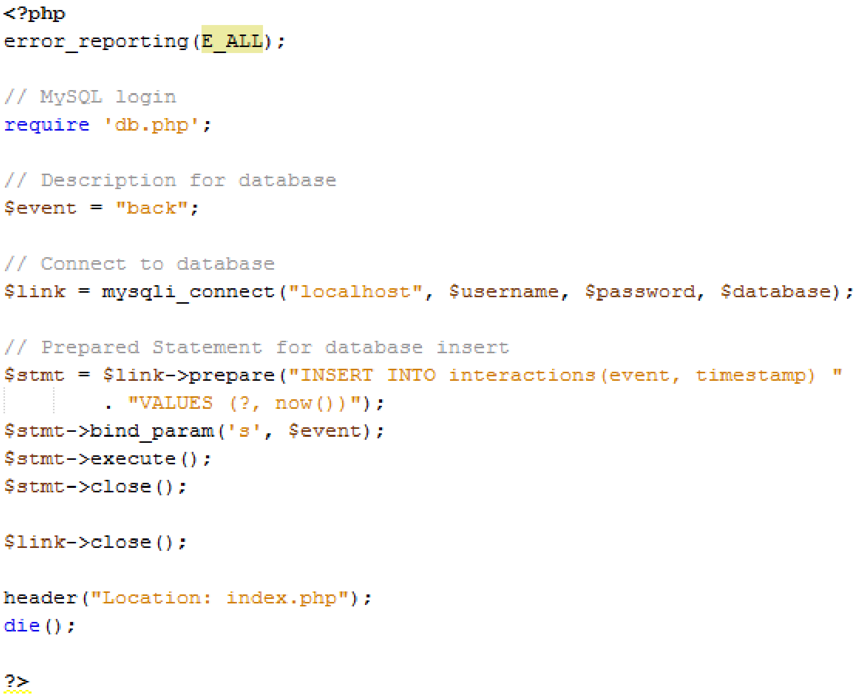

Server: mysql wampserver » Database: conopsradios » Table: interactions
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Triggers

The column name 'index' is a MySQL reserved keyword.

	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
□	1	index	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
□	2	event	text	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
□	3	timestamp	timestamp			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig. 16 - Interactions Table Schema

The relevant code to this table shown are scattered across the web pages using redirections. An example of this is shown in Fig. 17; when the user presses the “Back” button in any of the web pages, they will be redirected to this page which stores the $event string (varies depending on which button is pressed) and the current timestamp into the database’s Interactions table. The page then redirects itself back to the originally intended “index.php” page.



```php
<?php
error_reporting(E_ALL);

// MySQL login
require 'db.php';

// Description for database
$event = "back";

// Connect to database
$link = mysqli_connect("localhost", $username, $password, $database);

// Prepared Statement for database insert
$stmt = $link->prepare("INSERT INTO interactions(event, timestamp) "
        . "VALUES (?, now())");
$stmt->bind_param('s', $event);
$stmt->execute();
$stmt->close();

$link->close();

header("Location: index.php");
die();

?>
```

Fig. 17 - Interactions Sample Code

V. Experimentations

While developing the software, each individual function was tested along with the database schema. To ensure that the entire piece of software works as a whole, all values in the database were nulled and the database was populated using the New Radios and New Staff pages. Furthermore, by checking the Interactions table, it was ensured that each action taken while performing these actions were recorded and timestamped.

This process began by using a staff id to check out radios. Intentionally leaving either a staff id or radio id blank led to a popup with the relevant error message. If a department was already associated with five radios, a sixth radio would not be checked out and an error message would pop up. All of these events were recorded in the Interactions table.

Next, radios that have been checked out were returned, including radios that have not been checked out or does not exist. These errors returned an error popup and was recorded in the Interactions table. All returned radios were present in the Return History web page and did not show up in the Checkout History web page.

Next, a couple of radios were checked out to test the Inventory Control page. Any radios not checked out and returned by the Inventory Control page did not throw a popup error message. All radios that were checked out returned exactly like the return page without redirect to the "index.php" page. Instead, after an inventory controlled return, the page reloaded waiting for another radio to be scanned in.

VI. Implementation Observations

A couple of terms need to be defined before understanding the following paragraphs. A Rover is a volunteer staff member who are rotated around the convention to uphold the safety of attendees. Peacebonding is a subdepartment of the Rover department who mark cosplay (costume play) props that binds the liability of proper prop use safety to the attendees if approved. A Peacebonding Rover is therefore a dedicated Rover who is stationed at the Peacebonding room. Info Desk is a separate department who has a booth in the same room as Peacebonding; they provide answers to questions that attendees have. Heads and CoHeads are the managers and assistant managers of their respective departments.

The intention of this software was built for Heads/Co-Heads radio checkouts and returns. At the last minute, dual instances of the inventory tracking software was put into place for two separate entities: Heads/Co-Heads and Rovers. The Heads/Co-Heads instance of the software is managed by the Peacebonding subdepartment of Rovers, while the Rovers instance of the software is managed by the Rovers department. In order to show the difference between the Heads/Co-Heads and Rovers using the same system on separate computers, Kevin Wang added a drop down menu to query the results based on the department. This way, by selecting "Rovers" and using a submit button, the Checked Out records will only show the records of Rover checkouts. In addition, Rovers has the capability to check out more than five radios at a time, which requires the removal of the department limitation of five radios. At nights, Info Desk no longer has staff and a Peacebonding Rover takes their position.

ConOps (Convention Operations), which had dealt with the Heads' and CoHeads' radio checkouts in the previous years, staffed two to three staff to support their information desk and radio checkout process. Peacebonding Rovers, who has now taken over the radio checkouts, has two staff members, and are sometimes fully occupied by a long line of attendees wanting to peacebond their prop weapons. This means that they must interface with attendees while having the additional responsibility of checking out radios. This causes the staff to be highly stressed. A nearby staff from Info Desk will have to perform the checkouts or returns if both peacebonding and radio checkouts occurred simultaneously. While ConOps had stable and consistent staff who could be trained and their performance would improve over consistent usage of the web application, Info Desk constantly had new staff use the software. Training (knowledge transfer) was performed during shift changes and were quite quick since the software did not have many steps. While the web application's Interactions feature tracks the process times using the client computer, the actual processes extends slightly before and after usage of the software. For example, a staff member assembles the radio with a headset at some point during the process. Some staff will perform assembly prior to entering the Checkout process, while others will perform assembly in-between scanning of either the staff id, radio id, or headset id. Therefore, the time studies performed for the Checkout was split into four distinct events and the assembly process may have occurred in any of this events.

The four distinct categories for the time studies processes are: Arrival of Staff to Scanning the Staff ID, Scanning the Staff ID to Scanning the Radio ID, Scanning the Radio ID to Scanning the Headset ID, and Begin Radio Test to Complete Radio Test. Some staff will test their radios while still in the system, and others will test the radio after they have left the system. During the event, it was observed that there were errors in the physical processes surrounding the software. Some of these include: radio batteries not being charged properly, user did not recognize that a barcode scanner is present, user did not recognize that the client computer is touch screen, some headsets did not have barcodes to be scanned, and swapping headsets without returning and rechecking out a radio did not record the accurate headset in the database. There was a piece of paper immediately below the battery charger that reminds the user to re-insert the battery into the charger if there is an orange light. This note was often times ignored. When a senior user/staff who has familiarized themselves with the battery chargers pass by, they are able to fix those batteries that are not charging by re-inserting the batteries properly. Recognizing whether the batteries are charging is also based on their experience. Figure 18 shows the setup for the battery charging station and Figure 19 shows the note written to suggest to staff that an orange light requires reinsertion of the battery.

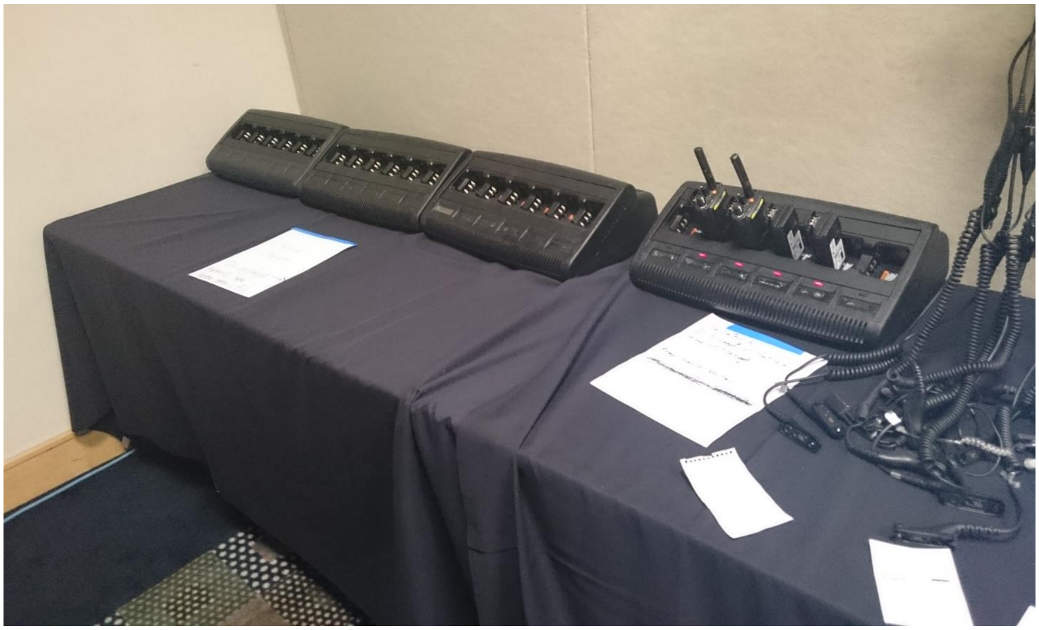
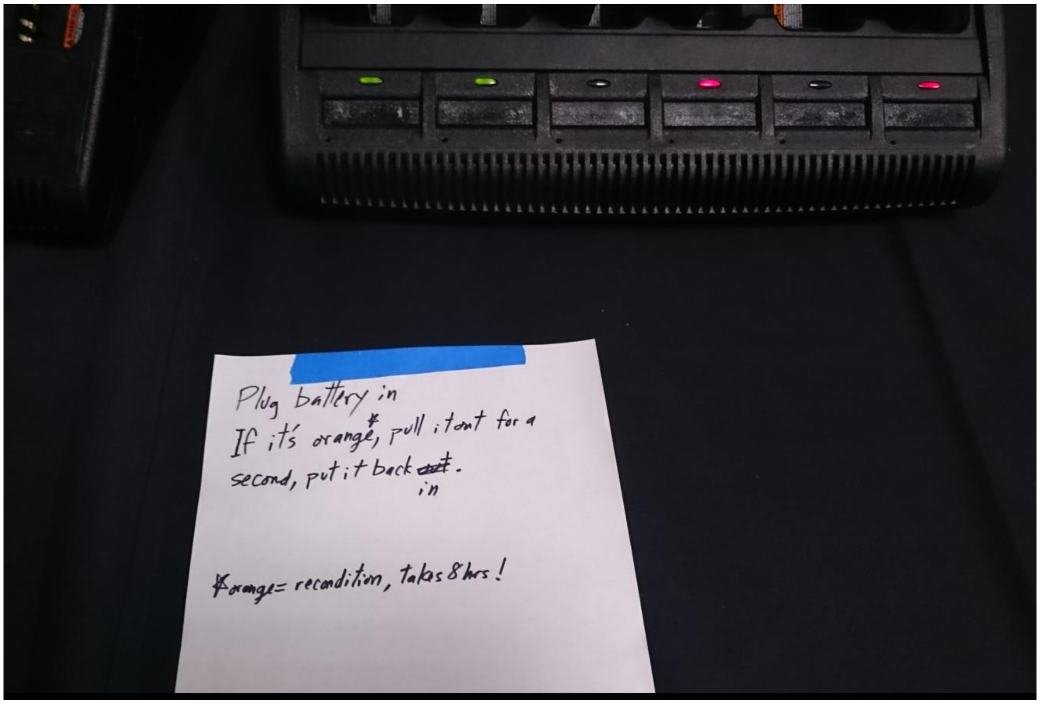


Fig. 18 - Battery Chargers



Plug battery in
If it's orange*, pull it out for a
second, put it back ~~out~~ in.

*orange= recondition, takes 8hrs!

Fig. 19 - Orange Light Note

For those staff who had not recognized that a barcode scanner was present, we had repositioned the barcode scanner next to the computer on top of a piece of paper with an outline of the barcode scanner and with a note saying to “Store Scanner Here”; shown in Figure 20.

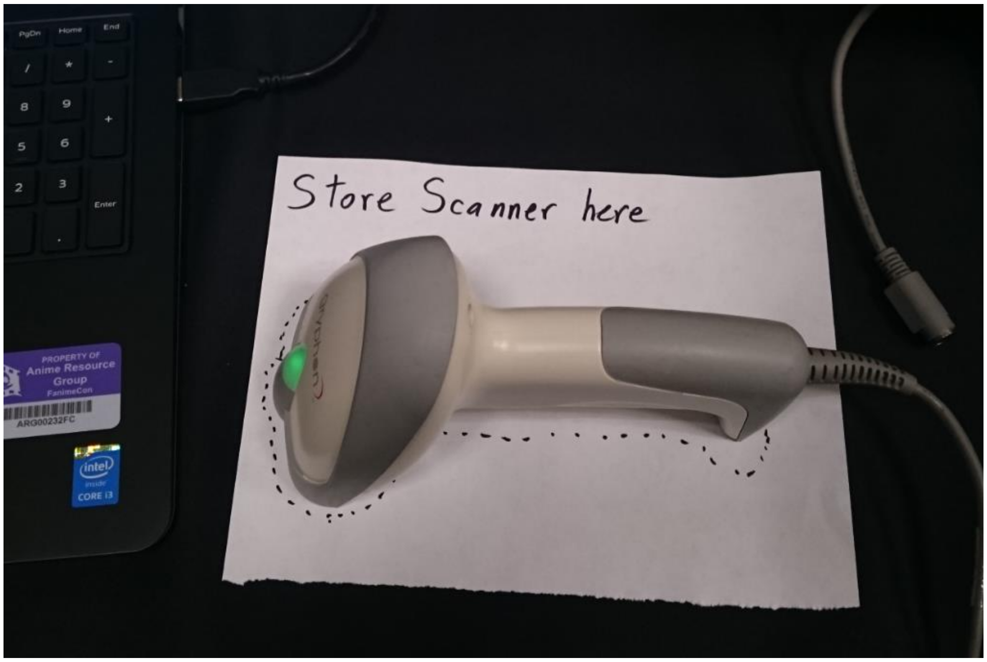


Fig. 20 - Scanner Note

The design of the web application has hard coded HTML that does not scale perfectly when the screen resolution of the monitor changes. Since there isn't a standard computer screen size that this application runs on, the design of the UI should be re-coded to scale properly across all platforms. The misalignment can be seen in Figure 21.

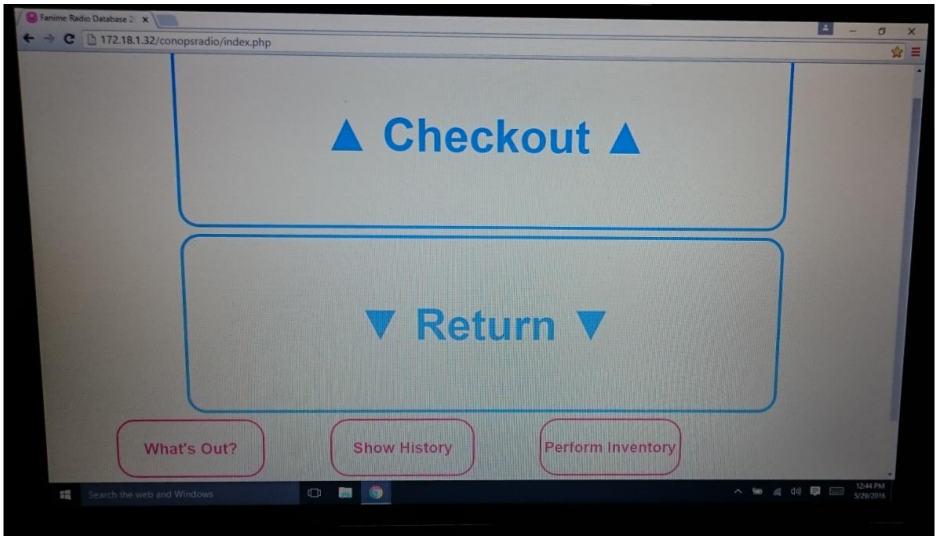


Fig. 21 - HTML Live Formatting

During the return process, after the radio return and component storage, the staff will move any fully charged batteries from the battery charger to the “Charged Batteries” location.

There isn't a standardized staff id for police officers or other individuals who are not in the system. The solution created by the untrained staff is to write down the name of the officer and the radio id that the officer has checked out. This loses the software timestamping function. To compensate for the missing police officer identifier in the radio database, the new staff web form was created. This required the creation of an arbitrary staff id from the staff, which was not performed. Therefore, the officer’s name and the radio id was written down; it was often overlooked and ignored.

In addition to the staff members of the Rovers department, the volunteers that request to be in Rovers through the Volunteers department became a part of the Rovers team for a limited amount of time. An untrained volunteer may be assigned to the Peacebonding Rover position, who essentially becomes a one-off radio checkout user. Figure 22 provides a picture of the facility layout.

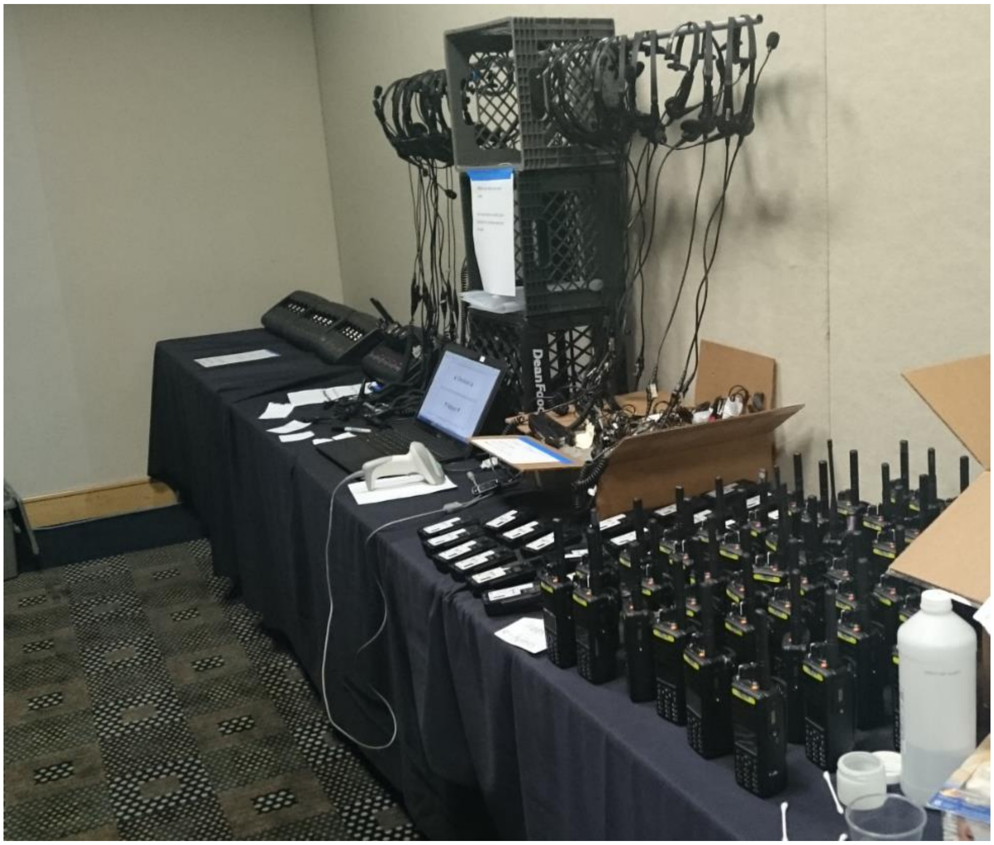


Figure 22 - Facility Layout

VII. Post Implementation Analysis

During the convention, time studies were taken based on Checkouts and Returns, and focused only on the Heads/Co-Heads instance of the web application since that was the original intention. The checkout process consisted of four separate time frames: upon arrival of the staff asking for a checkout until the scanning of the staff id, upon scanning of the staff id until the scanning of the radio id, upon scanning of the radio id until the scanning of the headset id, and finally, the beginning of radio testing until the completion of the radio test. Returns consisted of two separate time frames: upon obtaining the radio from the staff until scanning in the radio id, and beginning to store the radio components until completion of storing the radio components. Storage of the radio components included placing the radio into the battery charger and putting the headset back onto the hanger (or the earwig back into their slot). Although the headset id identifies as only one type of accessory barcode, there exists the choice of using either an earwig or a headset. The headsets are hung on a hanger, while the earwigs are disinfected every time upon checkout and return from a staff. There is an option for the staff to personally "own" the earwigs for the duration of the convention by storing them in a slotted box. This means that the the component storage times may vary depending on the type of communication tool (headset, earwig, or none) that was used by the staff.

Appendix E shows the I-chart for the time it takes for a staff member to enter the system until their staff id badge is scanned. The process seems to be out of control initially and stabilizes over time. This may be due to increased familiarity with the system and improvement has occurred. Appendix F shows the I-chart for the time between scanning the initial staff id badge and the radio id. This time is typically the time it takes for the staff member to get the radio and scan it. This value varies because some users will ask the staff to choose their own radio, while other users will pick the radio for the staff. Appendix G shows out of control data points towards the end of the data collection. Recall that both headsets and earwigs exist and can be selected upon checkout. The earwig is cleaned with alcohol by the staff and lengthens the time before the earwig gets scanned into the system. Appendix H shows the time between beginning a radio test and ending the radio test. A radio test consist of turning on the radio and speaking to ConOps to check whether their radio is functional. Some staff will perform the radio check within the system, while others will perform the radio check after leaving the system. It is also common for a user and a staff to perform small talk when interacting inside the system. These is the reason for the out of control data points. Appendix I is the summation of all the processes aforementioned. Appendix J shows the control chart for a user to obtain the radio from the staff and scan the radio back into the system. The initial out of control data points can be attributed to the initial learning process of the system. In addition, when a staff requests a “check-in” rather than a “return,” some users become confused and try to perform a checkout. This increases the time spent on this step as well in specific instances. Appendix K shows the time it takes for a user to store the returned components. During this time, there may be batteries and/or radios that are fully charged in the battery charger which is brought to the “charged battery” or “charged radio” location. Since these conditions are not easily measureable due to the length of time varying between different staff use of radios, the out of control data points are associated with these extra steps required from the user. Appendix L is the summation of all the return processes aforementioned.

VIII. Further Study and Alternative Solutions

To further improve this software, alternative solutions to the new requirements should be thoroughly reconciled. Since there will be two separate instances of the same database software, we can consider keeping the drop down menu installed by Kevin, but have the dichotomy of either Rovers or Heads/Co-Heads. This is opposed to allowing a staff to search across any department. Increasing the staff available at Info Desk or Peacebonding will alleviate the undue stress from a burst of attendees or staff coming in to either peacebond, ask questions, or perform checkouts. In order to further reduce training or knowledge transfer from staff to staff, using specific keywords next to the textbox entries will help guide the untrained staff to use the software. An example of this might be to have the keyword “Scan:” next to the “Badge ID” textbox. This will ensure that the staff is able to realized that there is an option to scan in the barcodes. To provide a whole solution to maintaining accurate radio inventory tracking, considerations in including radio assembly (mounting a headset onto the radio) should be taken into account. Since Info Desk is helping and the staff are new and untrained, stepwise checkin/chekouts can be considered. Examples of this would be to signal the staff to “Check headset and radio work together” when scanning in a headset, or “Check staff has reported to ConOps” before leaving the system. This solution will split the checkout process into multiple, separate, PHP pages.

Outside of the software, issues such as not charging radios properly or swapping headsets without returns will need to be revisited. In order to ensure that radios are charging (the light on the charger is green or red, not orange), a large orange banner with the word “BAD” in large font may be a potential solution to catch the attention of the staff, with additional information included in smaller font. This will reduce the chance that batteries will end up without charge for a long time. The temporary fixture of the piece of paper that had outlined the barcode scanner kept the barcode scanner visible and on to the table for the staff. This piece of paper may be replaced by a sturdier solution.

Bootstrap, an open source CSS package has a special class of objects called "containers". These are used to scale the CSS properly across different platforms. By implementing this, monitors with different resolutions will be able to maintain a similar look across the different instances. Radios handed out but was not recorded need to be accounted for. When performing the inventory control, an additional step can be taken to compare the null checked out radios set with the set of radios returned through inventory control. If they are not the same, then a log list of unchecked out radios should be shown and action should be taken to maintain data integrity. Since the radio inventory web application has two concurrent instances, the next step would be to control both environments to search for differences between the two instances. For example, while the Heads/Co-Heads do not require radio training, Rovers require radio training. This adds time to the process and require separate improvement efforts compared to the Heads/Co-Heads instance.

A recommendation for checking out radios to non-staff, is to designate a specific staff id for that entity. For example, a staff id "P100" barcode will be permanently hardcoded and used in the staff id field to check out a radio.

Since the Peacebonding Rovers are a subset of the Rover department, it is suggested to brief the Rovers or volunteers assigned to Peacebonding that the radio inventory software responsibility falls to Peacebonding. On shift change, relieved (experienced) Peacebonding Rovers can internally train new Peacebonding Rovers to use the inventory web application. A suggested table schema change would be to link the radio id and headset id separately to the staff id. The radio id and headset id would therefore not be linked. With this schema change for headset tracking, a new button known as "headset swap" can be included which will ask for the old headset and new headset identifiers. The swap will then replace the headset id of the referenced staff with another headset id.

Although took time studies of individual processes were taken, and problems were recognized during the processes evaluations, the next iteration of the software can automate this. Specifically, by using Javascript to send bits of information, it should be possible to record timestamps upon scanning in an item without submitting the form. Timestamps using this method may use up additional resources and cause lag when interacting with the database; timestamps therefore may affect the performance of the software.

IX. Conclusion

The intended goal of providing an accurate inventory tracking software is partially met with the new improvements using touch screen interfacing, large buttons, large font, large textboxes, and little menu choices. Since this iteration of the web application is the first to collect live data, these data points will become the baseline when implementing new alternatives. Primarily, I-charts are used to look for processes that are out of control. Incremental changes based on new requirements in the process requires re-evaluation of any analysis and alternative solutions created prior to the requirement changes. Overall, this project was a success due to the simple interface being easily used and easily picked up by untrained staff. The new staff form was not used as intended; when either security or SJPD, who did not have staff ids, checked out radios, many web application users did not know how to deal with it. This will therefore be replaced, and staff members will not be required to add additional staff. To check for whether the checkout process itself was accurate, we are able to rely on the results from the inventory control process. Again, this process returns any radios that was not returned through the regular process; if there are any radios returned through this process, the timestamp and history recording will provide a lead to find the causation of this error.

Finally, to ensure that the fifth iteration of this software meets the new demands discovered during implementation, it will be important to account for the additional instance of the web application, the untrained staff and eligible (but non-staff) radio staff. Also, since the staff had flexibility in choosing which procedures should occur first, the data points were unable to capture the actuality of the controlled environment. Therefore, a new facility redesign will take place by guiding the staff through each step of the process. The data captured by this method will ensure that the collected data is consistent and useable.

Bibliography

Karat, J., McDonald, J. E., & Anderson, M. (1986). A comparison of menu selection techniques: touch panel, mouse and keyboard. *Int. J. Man-Machine Studies, 25, 73-88*.

Suto, H., Sakamoto, M., & Okita, M. (2009). Affect of color of interface on accuracy and speed of operations. *2009 International Conference on Biometrics and Kansei Engineering, 201-204*.

Punter, T., Ciolkowski, M., Freimut, B., & John, I. (2003). Conducting On-line Surveys in Software Engineering. *Accepted for ACM-IEEE Int. Symposium on Empirical Software Engineering (ISESE 2003) - v1*.

Singh, A., Taneja, A., & Mangalaraj, G. (2009). Creating Online Surveys: Some wisdom from the

Trenches Tutorial. *IEEE Transactions on Professional Communication, 52(2), 197-212*.

Zumstein, D., Drobnjak, A., Meier, A. (2011). Data Privacy in Web Analytics - An Empirical Study

and Declaration Model of Data Collection on Websites. *Proceedings of the 7th International Conference on Web Information Systems and Technologies (WEBIST-2011), 466-472*.

Akkus, I. E., Chen, R., Hardt, M., Francis, P., & Gehrke, J. (2012). Non-tracking Web Analytics. *ACM 978-1-4503-1651-4-12-10, 687-698*.

Bei, L., & Yuan, S. (2013). Software Intellectual Property Management through Self-claiming of the Certificate of Origin of the Source Code. *2013 International Conference on Computational and Information Sciences, 613-615*.

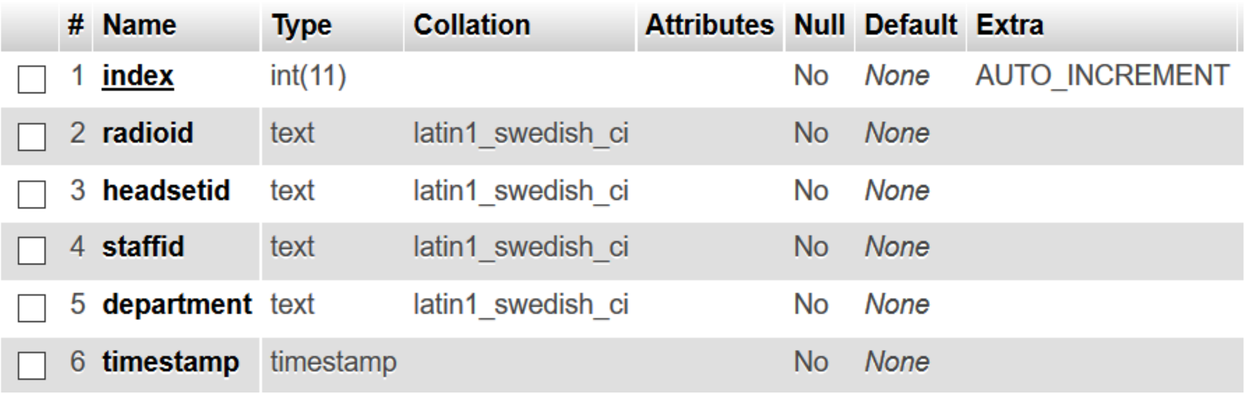
Bangor, A. W., & Miller, J. T. (2005). The Design and Presentation Order of Web Page Buttons. *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting, 1449-1453*.

Kumar, L., Singh, H., & Kaur, R. (2012). Web Analytics and Metrics: A Survey. *International Conference on Advanced in Computer, Communications and Informatics, 966-971.*

Prepared statements and stored procedures. (2001-2016). http://docs.phpmyadmin.net/en/latest/import_export.html

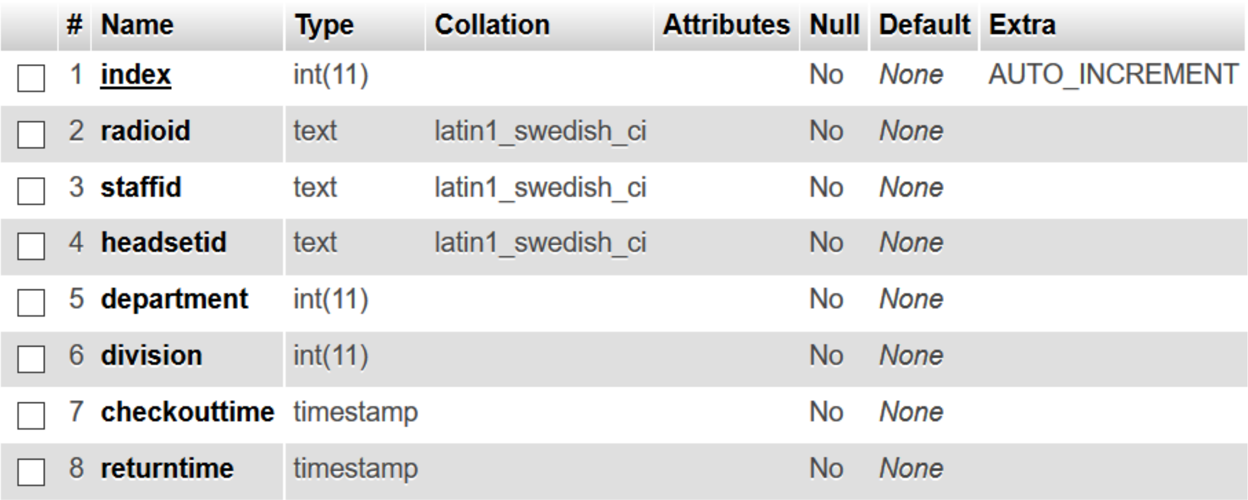
Import and Export. (2012-2014). http://docs.phpmyadmin.net/en/latest/import_export.html

Appendices



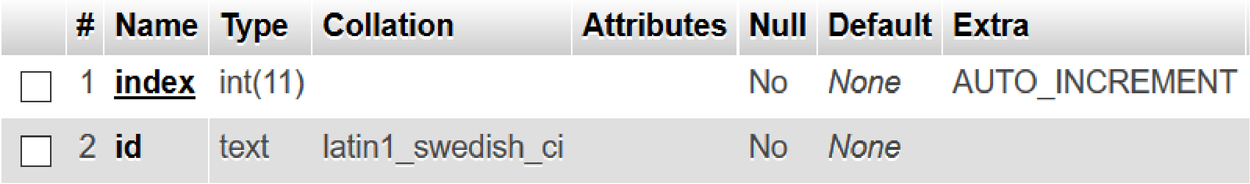
	#	Name	Type	Collation	Attributes	Null	Default	Extra
□	1	**index**	int(11)			No	*None*	AUTO_INCREMENT
□	2	**radioid**	text	latin1_swedish_ci		No	*None*	
□	3	**headsetid**	text	latin1_swedish_ci		No	*None*	
□	4	**staffid**	text	latin1_swedish_ci		No	*None*	
□	5	**department**	text	latin1_swedish_ci		No	*None*	
□	6	**timestamp**	timestamp			No	*None*	

A - Checkout Table Schema



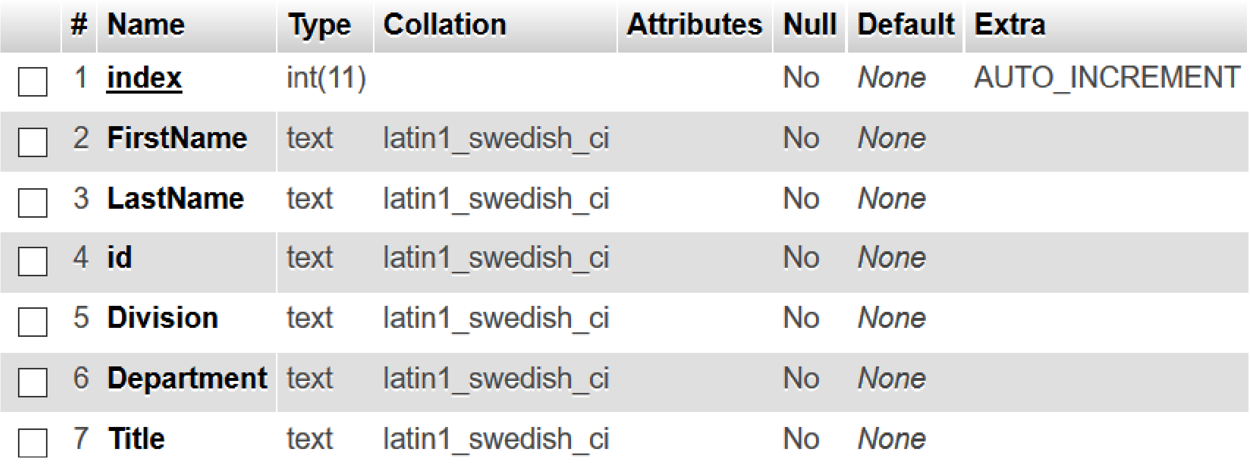
	#	Name	Type	Collation	Attributes	Null	Default	Extra
□	1	**index**	int(11)			No	*None*	AUTO_INCREMENT
□	2	**radioid**	text	latin1_swedish_ci		No	*None*	
□	3	**staffid**	text	latin1_swedish_ci		No	*None*	
□	4	**headsetid**	text	latin1_swedish_ci		No	*None*	
□	5	**department**	int(11)			No	*None*	
□	6	**division**	int(11)			No	*None*	
□	7	**checkouttime**	timestamp			No	*None*	
□	8	**returntime**	timestamp			No	*None*	

B - History Table Schema



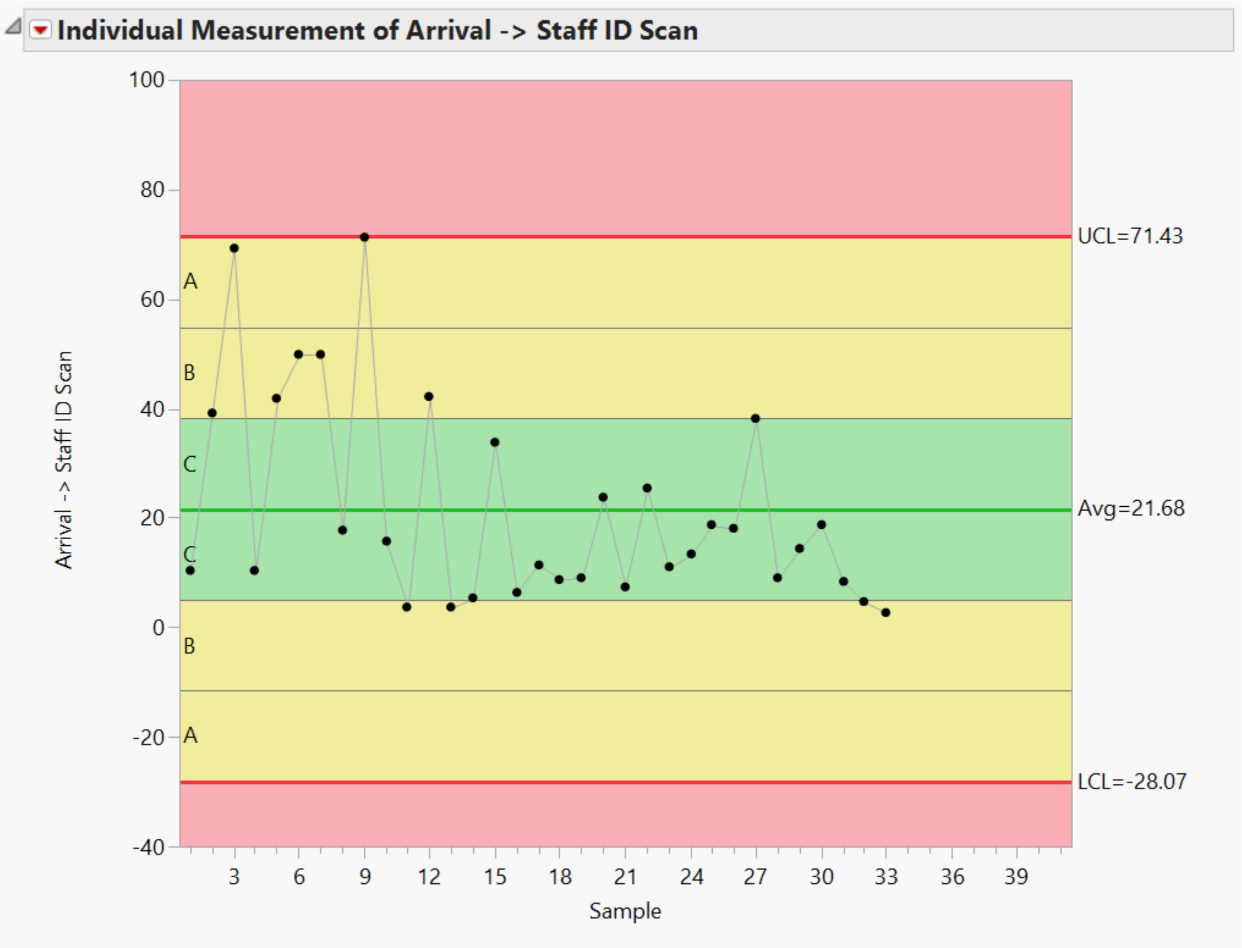
	#	Name	Type	Collation	Attributes	Null	Default	Extra
□	1	**index**	int(11)			No	*None*	AUTO_INCREMENT
□	2	**id**	text	latin1_swedish_ci		No	*None*	

C - Radios Table Schema

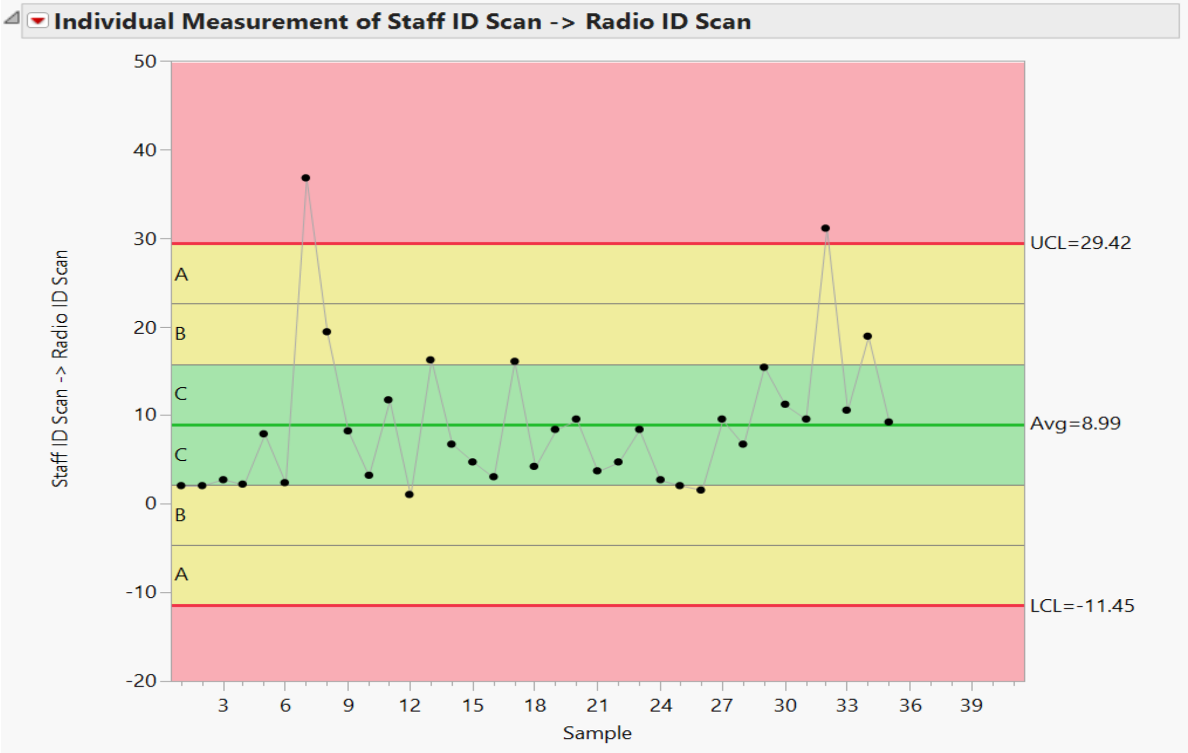


	#	Name	Type	Collation	Attributes	Null	Default	Extra
□	1	**index**	int(11)			No	*None*	AUTO_INCREMENT
□	2	**FirstName**	text	latin1_swedish_ci		No	*None*	
□	3	**LastName**	text	latin1_swedish_ci		No	*None*	
□	4	**id**	text	latin1_swedish_ci		No	*None*	
□	5	**Division**	text	latin1_swedish_ci		No	*None*	
□	6	**Department**	text	latin1_swedish_ci		No	*None*	
□	7	**Title**	text	latin1_swedish_ci		No	*None*	

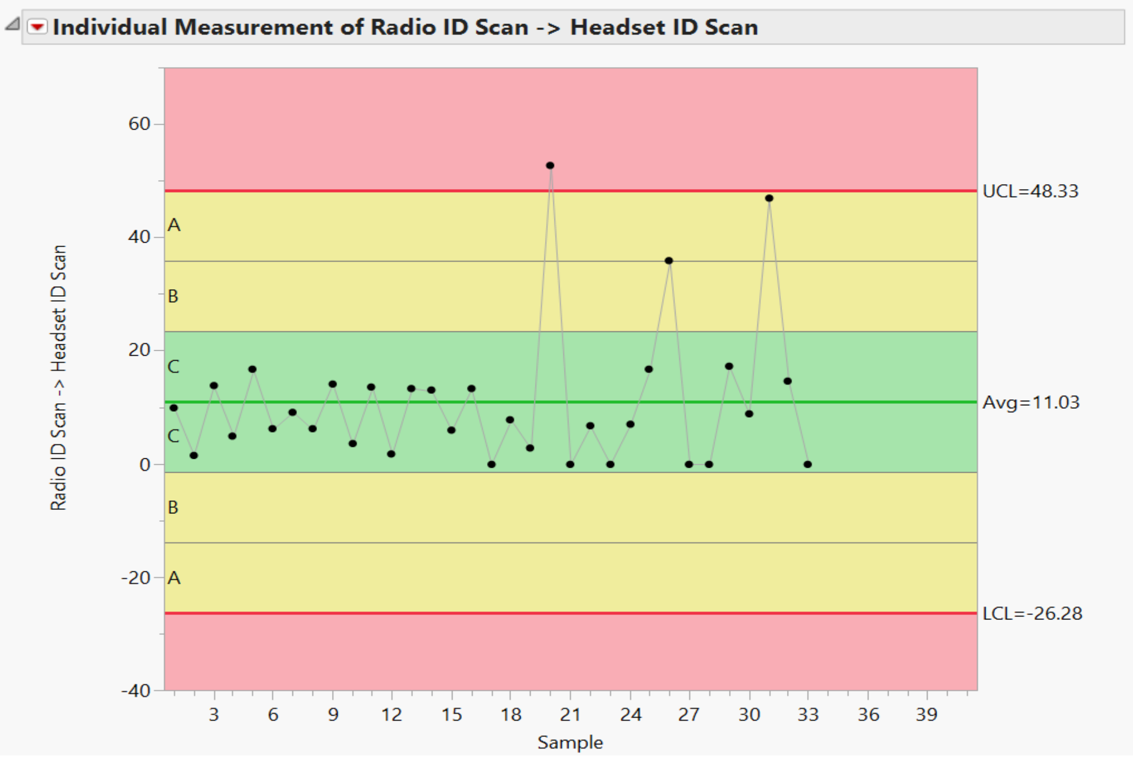
D - Staffs Table Schema



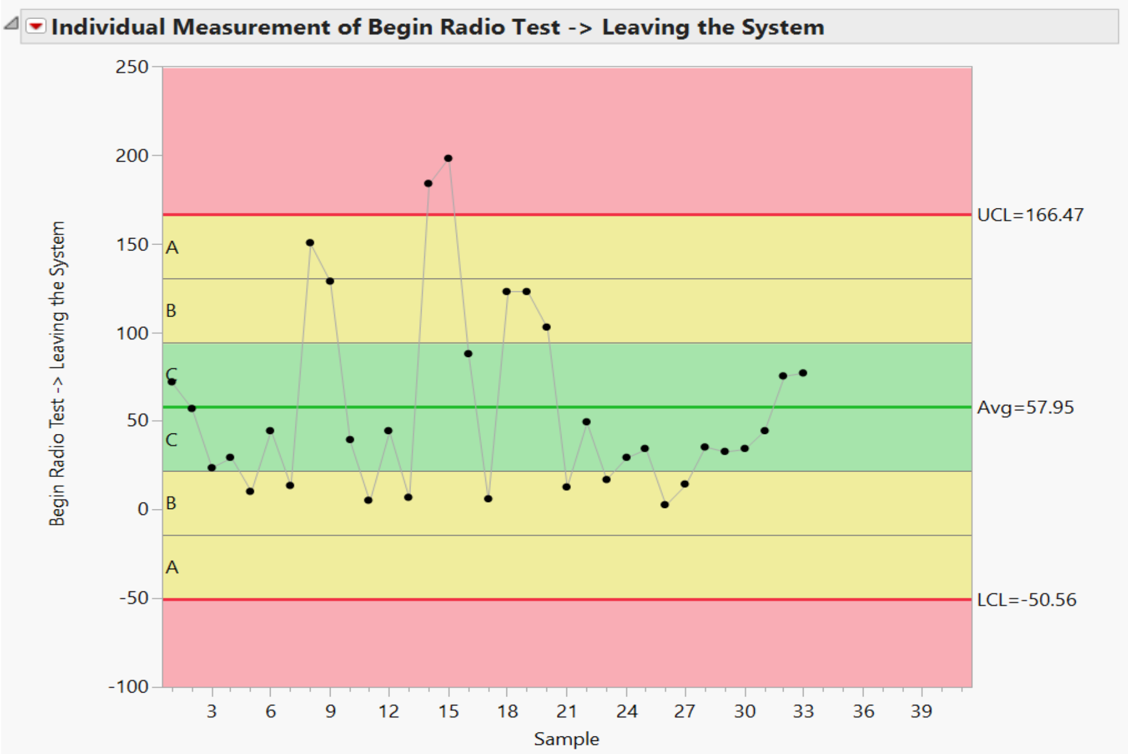
E - Checkout Process 1 Control Chart



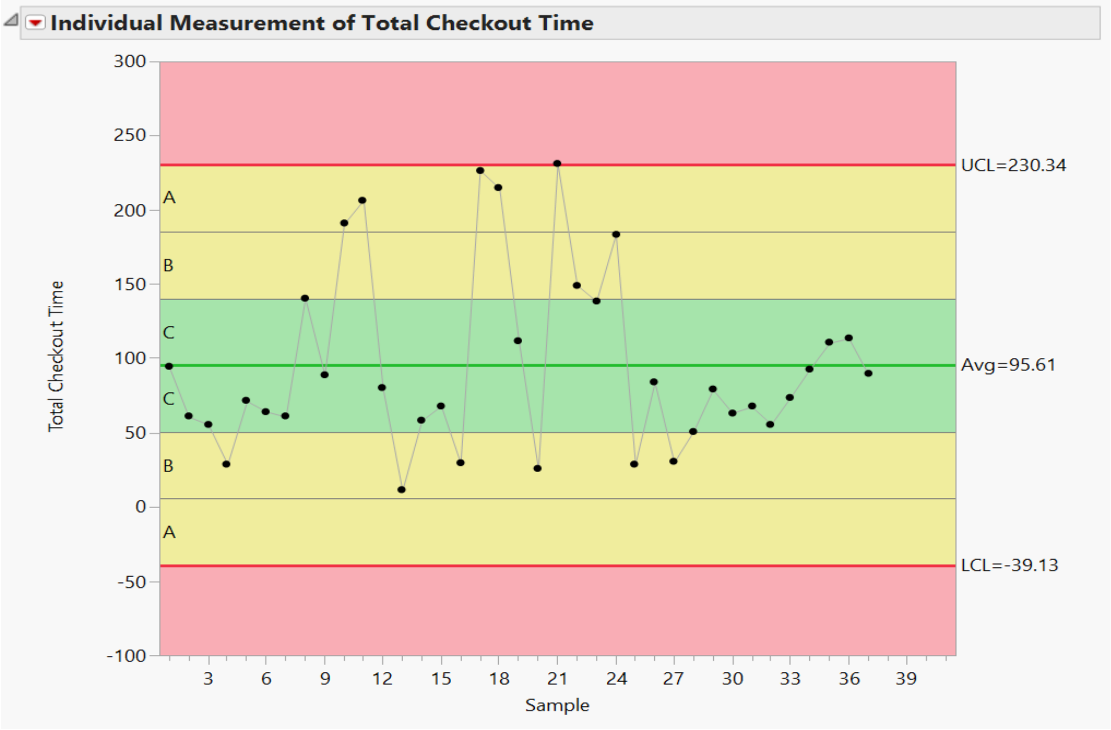
F - Checkout Process 2 Control Chart



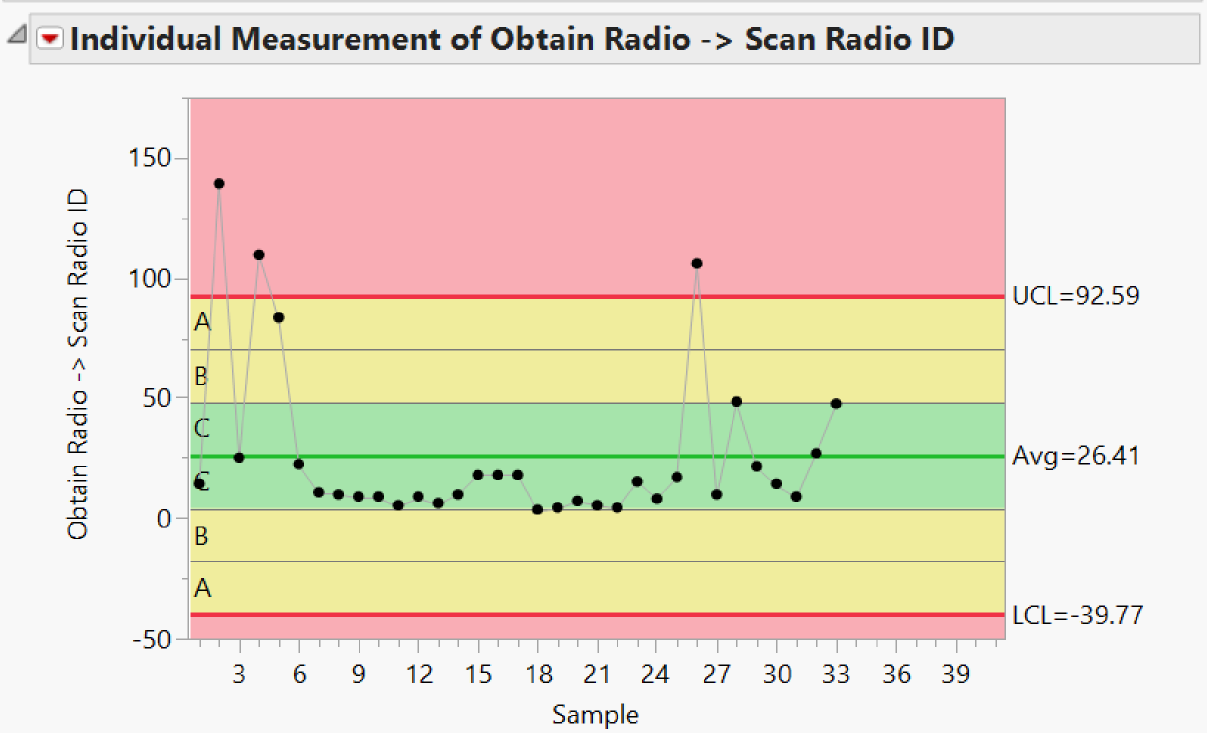
G - Checkout Process 3 Control Chart



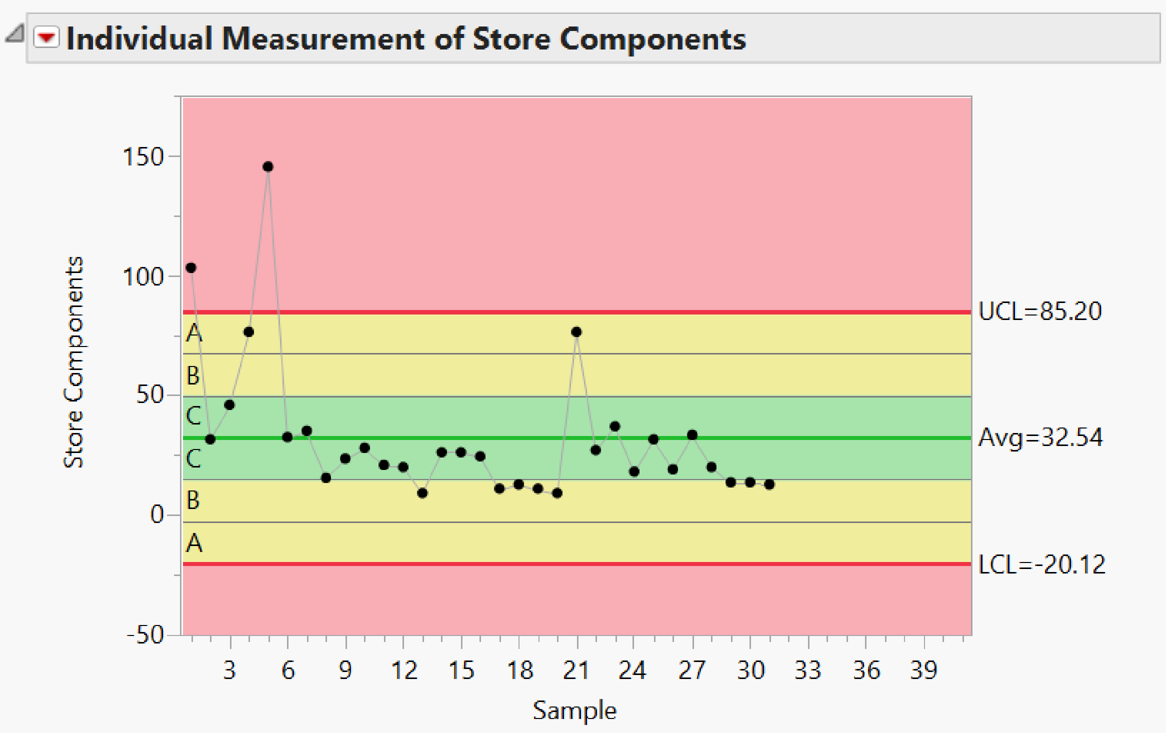
H - Checkout Process 4 Control Chart



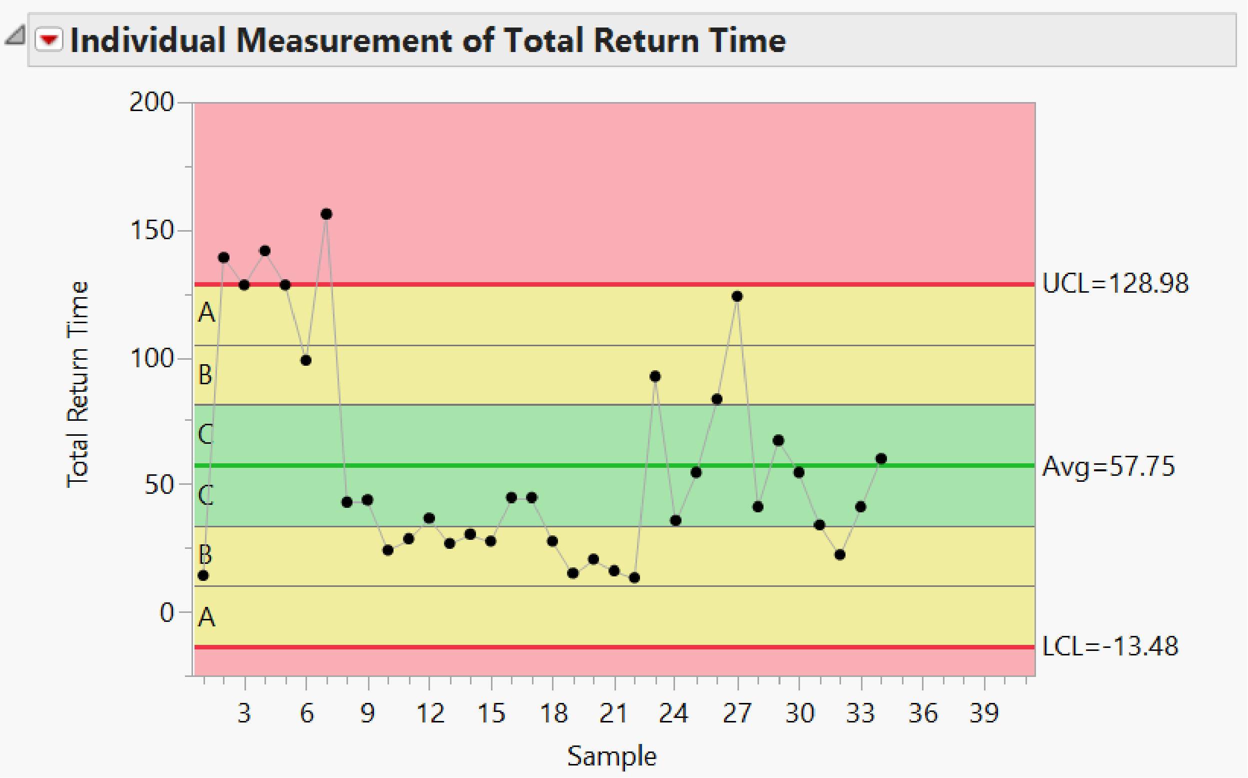
I - Checkout Process Total Control Chart



J - Return Process 1 Control Chart



K - Return Process 2 Control Chart



L - Return Process Total Control Chart