CEN4010 Principles of Software Engineering

Spring 2018

Milestone 4: Beta Launch and Final Project Reviews

Group 9: Hardware Nanny

Alicia Mitchell (amitch41@fau.edu)– Scrum Master

Sergio Ramirez – Product Owner

Bradley Rubenstein – Back End Lead

Guimel Gonzalez Hernandez – Front End Lead

Ryan Juraidini – GitHub Master

Date: Monday April. 16th, 2018

|  |
| --- |
| Document Revision History Table |
| Requirements Specification Document: Revision 1 - 2/19/2018 |
| Requirements Specification Document: Revision 2 - 3/26/2018 |
| Beta Launch and Final Project Reviews 1 - 4/16/2018 |
|  |

**Table of Contents**

1. Product Summary………………………………………………………………………………………………….Page 3
2. Usability Test Plan………………………………………………………………………………………………….Page 5
3. QA Test Plan…….……………………………………………………….………………………….….………….Page 12
4. Code Review………………..……………….………….…………….……………………………………………Page 19
5. Best Practices for Security………………....………………………………………………………………..Page 22
6. Non-functional Requirements.………………………………...………………………………………....Page 24
7. **Product Summary**

**Name of the product**: Hardware Nanny

**Marketing Strategy:**

**Product Objective**

The Hardware Nanny inventory management software solution was created in mind to provide reprieve to the tedious manual inventory management strategies that are typically employed and exhaust time and energy. The ambition of the system is to give its users a convenient, easy to use, effectual space to manage inventory and acquire items. The software itself offers a place to request various hardware items for both students and faculty to purpose for their respective academic, work, and research objectives. The system was built with goal to service its university clientele and make hardware inventory management and acquisition that much easier and more efficient.

**Position in the Market**

When considering the method in which to situate the Hardware Nanny product in the market we would start by pondering the target audience. In our case the university students, personnel and Hardware equipment management staff. From there the next goal would be to consider what software we could offer this audience. For this motivation we reflect on the school project’s humble beginnings that inspired our team to produce a versatile software tool to manage and sell various hardware equipment and parts transaction free. We the creator’s computer science and engineering students can understand the purpose and utility of such a product and sought to bring this insight combination and consideration to life.

The services that the Hardware Nanny’s software provides allows it to occupy in a niche market. The software has been customized and designed to accommodate the needs of the hardware parts manager. The position of the product in the market encourages use of more confined marketing strategies such as low budget marketing tactics and visual marketing strategies via the university website which brings exposure to the software and celebrates it on a local level within the university. We can also apply guerrilla marketing strategies to promote our product by advertising hardware nanny events when hardware parts from the store our used as well as making & offering t-shits to support the brand.

**Applicability of Software**

Hardware Nanny is still a mailable software system in its own right and can be adapted and built to suit other university and educational institutions, to give them meaningful inventory management assistance with their own inventory system infrastructures. Hardware Nanny itself can also be offered as an inventory management solution template for similar businesses. The adaptability and simplicity of its design and construction make the software well suited for similar businesses looking to both offer and manage inventory. Hardware Nanny gives its users and potential customers a flexible solution with its inventory management and retailing capabilities.

**All major committed priority 1 functions:**

1. Adding an Item to inventory

* Users must be able to see the entire list of parts and navigate the entire parts list and view individual parts
* Parts information will include: and be part number, part name, description of part, quantity in stock, price, vendor of part, & availability and can be viewed individually when viewing items via the Inventory page
* Staff and Admins must be able to add new inventory items at their discretion and changes be reflected in the Inventory table
* Uploading inventory items as a CSV file and view the changes in the inventory section

1. Shopping from the parts inventory catalog

* Users must be able to add items to a cart, which can then be ordered.
* Cart will include the number of each part ordered, the quantity can be changed and updated
* Users can edit their cart such as remove or add items from the cart or continue shopping.

1. Ordering Inventory Items & Viewing Orders:

* Customers must be able to place orders for inventory items
* Admin must be able to view all approved, rejected, and pending orders and filter the results based on status.
* Admin must be able to mark pending orders that come through the system as approved orders at their discretion
* Admin must be able to mark pending orders that come through the system as rejected orders at their discretion

1. Viewing All Profiles and User Profile Information:

* Users must be able to login with their credentials using email and password and have a specific view and button restrictions based on user type
* Users will have one of three levels of access (customer, staff, admin)
* Users must be able to see the information in their account, which includes: Name, account#, email, college, department, class information
* Admins must be able to view all Staff and User accounts

**Unique features:**

This software uniqueness comes from a simple place the fact that it is tailored to the university and to its intended end user Perry. It applies a simplistic easy to navigate interface to allow our end user to fluidly achieve his daily tasks in a more efficient manner. The software also provides students and university faculty an easy tool to shop for needed parts for class and independent study projects and labs.

**URL of the system to be tested:**

[**http://lamp.cse.fau.edu/~CEN4010\_S2018g09/M4/signin.php**](http://lamp.cse.fau.edu/~CEN4010_S2018g09/M4/signin.php)

**SUPPORTING VIDEOS**

**YouTube Shopping & Ordering Feature**

[**https://youtu.be/blMgcPQJlwQ**](https://youtu.be/blMgcPQJlwQ)

**YouTube Admin Features**

[**https://youtu.be/Y9Y29brZH\_0**](https://youtu.be/Y9Y29brZH_0)

1. **Usability test plan – maximum 2 pages**

**Feature to be tested:** shopping for inventory and ordering inventory items

**1) Test objectives:**

1. Validate that a user can shop successfully when on the hardware nanny site and can gauge how user friendly it is and its amount of eye appeal.
2. They are able to easily navigate the inventory page to see the totality of various parts offered.
3. Demonstrate the natural and simplistic way of navigating the site pages to view individual inventory items for the purpose of ordering and/or evaluating.
4. Show that via the site that there is a space to manage and review all of the shopping cart items.
5. Spotlight the ability to add/remove items in the cart or alter the quantity of inventory items in the shopping cart.
6. Showcase the ability to checkout and order items in the shopping cart.
7. Demonstrate order history after placing an order for both the user placing the order and the admin approving or rejecting the order.

**2) Test plan**

**System Setup**

Provide a space and setup for the software to be consistently and effectively tested by selected testers to evaluate the experience with performing all the test objectives. Establish a setup for the respective users that can offer a black-box testing style.

**Staring Point**

Create instructions for the user to guide them as they use the software for the first time and undergo the shopping experience.

Gather various users to individually use software based on the specified instructions and test them consecutively and provide them a place to record their experience

**Tasks to be accomplished**

1. Demonstrate that a user can peruse successfully when on the hardware nanny site and can gauge its user friendliness and eye appeal.
2. They can navigate the inventory page to see the totality of various parts offered.
3. Convey the fluid navigation and simplicity to view individual inventory items for the purpose of ordering and evaluating.
4. Present the ability to add/remove or alter the quantity of inventory items in the shopping cart.
5. Present the ability to add/remove or alter the quantity of inventory items in the shopping cart.
6. Demonstrate the checkout process of items in the shopping cart.
7. Demonstrate order history after placing an order for both the user placing the order and the admin approving or rejecting the order.

**Intended Users**

Allow 5 independent testers to use the Hardware Nanny Software beta with the specified feature. A black box testing approach was applied so that user could offer a more objective unbiased view of their experience using the software.

Each user that tested the software was requested to perform all of the operations listed above and comment and report to the Likert scale with their experience.

**Completion criteria**

Inform all testers to engage in all software operation instructions and allow them to give feedback on each task until all tasks have been completed.

**URL of the system to be tested:**

[**http://lamp.cse.fau.edu/~CEN4010\_S2018g09/M4/signin.php**](http://lamp.cse.fau.edu/~CEN4010_S2018g09/M4/signin.php)

**Usability Testers**

**Tester 1:**

**Name: Britt Forbes**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree | Strongly Disagree | Comments |
| Question 1:  Website has a user friendly and eye appealing Interface? |  |  |  |  |  |  | I was able to find everything easily and the design of the site is cool |
| Question 2:  Website is easy to navigate and view and select inventory |  |  |  |  |  |  | It functioned as intended but some of buttons were a little small |
| Question 3:  You can shop fluidly and add/remove cart items and update quantity |  |  |  |  |  |  | It was simple to add/remove/edit inventory in the car. I wish it was easier to tell when things new items were added to the cart. |
| Question 4:  You can upload new items and inventory easily and view orders placed and change their status |  |  |  |  |  |  | It was able to upload and view orders with no issues |

**Tester 2:**

**Name: Andrea Gerroni**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree | Strongly Disagree | Comments |
| Question 1:  Website has a user friendly and eye appealing Interface? |  |  |  |  |  |  | I think the slideshow is weird, but everything else looks good |
| Question 2:  Website is easy to navigate and view and select inventory |  |  |  |  |  |  | I thought it was easy enough to find what I was supposed to |
| Question 3:  You can shop fluidly and add/remove cart items and update quantity |  |  |  |  |  |  | Everything was accessible and easy to modify. |
| Question 4:  You can upload new items and inventory easily and view orders placed and change their status |  |  |  |  |  |  | It worked |

**Tester 3:**

**Name: Richard Leflur**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree | Strongly Disagree | Comments |
| Question 1:  Website has a user friendly and eye appealing Interface? |  |  |  |  |  |  | Good design, I like the background |
| Question 2:  Website is easy to navigate and view and select inventory |  |  |  |  |  |  | The prices don’t make sense, but everything works |
| Question 3:  You can shop fluidly and add/remove cart items and update quantity |  |  |  |  |  |  | I was able to manage the cart with no problem |
| Question 4:  You can upload new items and inventory easily and view orders placed and change their status |  |  |  |  |  |  | The upload button was a little out of the way, but it worked fine |

**Tester 4:**

**Name: Diane Surdel**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree | Strongly Disagree | Comments |
| Question 1:  Website has a user friendly and eye appealing Interface? |  |  |  |  |  |  | It’s stylish and user friendly |
| Question 2:  Website is easy to navigate and view and select inventory |  |  |  |  |  |  | I was able to operate the website without issue |
| Question 3:  You can shop fluidly and add/remove cart items and update quantity |  |  |  |  |  |  | Worked perfectly |
| Question 4:  You can upload new items and inventory easily and view orders placed and change their status |  |  |  |  |  |  | The order and upload files functionality worked nice |

**Tester 5**

**Name: Joshua Myers**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree | Strongly Disagree | Comments |
| Question 1:  Website has a user friendly and eye appealing Interface? |  |  |  |  |  |  | Very nice visuals and use of color on the site |
| Question 2:  Website is easy to navigate and view and select inventory |  |  |  |  |  |  | The inventory information was formatted strangely, but was easy to use |
| Question 3:  You can shop fluidly and add/remove cart items and update quantity |  |  |  |  |  |  | Very easy to add to your cart but would like a search functionality |
| Question 4:  You can upload new items and inventory easily and view orders placed and change their status |  |  |  |  |  |  | Uploading and status changing seemed straightforward |

1. **OA Testing**
2. **Test objectives:**
3. Validate that a user can shop successfully when on the hardware nanny site meaning that they can navigate the inventory page to see the totality of various parts offered.
4. Demonstrate the natural and simplistic way of navigating the site pages to view individual inventory items for the purpose of ordering and/or evaluating
5. Show that via the site that there is a space to manage and review all of the shopping cart items
6. Spotlight the ability to add/remove items in the cart or alter the quantity of inventory items in the shopping cart
7. Showcase the ability to checkout and order items in the shopping cart
8. Demonstrate order history after placing an order for both the user placing the order and the admin approving or rejecting the order
9. **Hardware & Software Setup**

**Hardware used:** Desktop Computer using windows 10 operating System

**Software Browsers used:** Edge & Chrome

A selected group member using their personal computer hardware ran all of the software QA tests on the feature described above. The software was tested for the quality assurance test cases with two different browsers on the same computer each was done using the same test objectives. Each test was performed three times on each browser to ensure consecutive performance the average results are specified on the QA Test Case table. In the process the software was simply accessed via navigation after uploading the beta version to the lamp server.

The tester began the software test setup by going to the Hardware Nanny login page using the lamp server URL The computer used to test the software experienced no operational issues upon setup with both browsers. Additionally, during the setup of one of the testing cycles a video recording program open broadcaster was used simultaneously while testing the software through each of the test case. The video is link has been added to the document.

1. **Feature to be tested**

The feature tested in the quality assurance testing phase was shopping for inventory and ordering inventory items. This feature itself exposes many of traits and positive elements of the software that we are proud of. We aimed to showcase the whole process of the item/s that start in the inventory section and then go on to become an order in the system viewable by admin. We wanted to test how the user goes onto the website and browses the inventory page for items and views the item information. From there can select items add them to the cart and adjust the quantities of the items and remove or add items as needed. Following this they can checkout their order and the order becomes part of the order system where it can be viewed or rejected at admin or staff discretion.

Moreover, we wanted used the quality assurance testing as an opportunity to examine the software flexibility and ease of use and showcase aspects of its intuitive nature. The wanted to test if it was informing the user consistently and properly about the quantities of their cart items and the restrictions and how clear it was to modify their cart and review item information. Furthermore, we wanted to inspect its ability to change order statuses and keeps records of orders placed by account number. The QA testing of the shopping and ordering feature conveyed the strengths and weakness of our software system as well as its potential growth and possibility.

1. **Actual 4 Test Cases**

**Test Case 1**

**Title**: View Inventory & Inventory Items- Access Inventory page and View items

**Description:**A user should be able to login to the site and have their credentials give them access to the landing page. At the landing page they can navigate to the Inventory Items. Here they can see all inventory items and view any item that they choose from that inventory list

**Test Steps:**

1. Navigate to Hardware Nanny Login Page
2. Enter their user credentials and click the ‘Login’ button
3. From there they see their user landing home page select ‘Inventory’ on the top left options bar.
4. Once navigated you should see all the items on the inventory page which you can scroll to see all the options
5. While on the inventory page select the view button on any item on the page to further examine that individual item

**Expected Results:**After login with proper credentials the landing page navigation bar and upper menu bar must be visible along a specific view tailored to the profile type. Additionally a image slider and user information is visible in the main page body. After selecting the inventory option on the inventory page the user should see all of the inventory items displayed. Clicking the view button on any item on the inventory page should show the individual inventory item selected with its respective details about the item.

**Test Case 2**

**Title**: Shopping for Items & Filling your cart– View all the items that you can add to cart and do so

**Description:**A user can while having site access peruse the inventory page and select items to add to their cart. They can observe the changes in the cart bubble number as they add more cart items. By clicking on the cart they can see their shopping cart. Here at the shopping cart page they can add, remove or update item qualities before checkout

**Test Steps:**

1. Navigate to the Inventory Page to view orderable items
2. Click the ‘Add To Cart’ button and observe the upper right green bubble near the shopping cart change in value repeat as necessary to demonstrate this ability with this and other inventory items
3. Click on the ‘Shopping Cart’ on the upper right part of the page
4. The shopping cart page should now be viewable
5. Go to the inventory items quantity box on an individual item box with another number and update the inventory item. Observe the change in the upper right green bubble near the shopping cart
6. While on the shopping cart page remove an item from the shopping cart by selecting the red x on an individual item box on shopping cart page observe the change in the upper right green bubble near the shopping cart

**Expected Result:** The user should notice that all inventory items can be added to the cart after clicking the add to cart button and logically updating the green bubble near the shopping car. Once the shopping cart itself has been clicked the cart items selected on the inventory page and added to cart are viewable. At the shopping if the user inputs a new number value in the quantity box for a specific item this should reflect in the cart automatically and be shown by the green cart bubble. When the remove X button is clicked the item should be removed as specified and the change reflected in the green shopping cart bubble.

**Test Case 3**

**Title**: Checkout & Order Records – Checkout with shopping cart items order is reflected in the orders page, Orders can be approved or rejected

**Description:**Admin, Staff & Students can place an order and checkout and are brought to the confirmation page with order information. After completing the checkout process then their order becomes viewable in the orders section on their profile. For admin & Staff all orders for all users can be seen. At the orders page for admin or staff the orders can be evaluated and approved or rejected this new status is reflected in the orders table

**Test Steps:**

1. Assuming you are ready to place an order for inventory parts and are at the shopping cart page
2. The admin or staff or student user in this example clicks on the ‘Checkout’ button and redirected to the confirmation page
3. Then and admin can login and go to the orders page or go from the confirmation page to the orders page and view all placed orders as well as the order that was just placed.
4. Here with all orders available they can see the orders with two options reject and approve on all pending orders
5. If approve is selected the order updates with an approved status
6. If rejected is selected the order updates with a rejected status
7. All orders are still viewable on the orders page as can be observed
8. Orders can be filtered by status pending, rejected and approved

**Expected Result:**Once the user checkouts with their shopping cart with inventory then the are brought to the confirmation afterwards if given an admin or staff user that can navigate to the orders page and see all orders placed including the previously placed order. Here the order can be rejected or approved depending on which button is clicked. This shows that the item was updated to a new status on the orders table.

**Test Case 4**

**Title**: Uploading multiple Inventory items using a CSV File

**Description:**Admin, Staff & can go to the Inventory page where they can see as the last option the Upload Inventory Batch on the left side menu and click on it. Afterwards at the upload batch page they can browse for the CSV file with multiple inventory items and upload the file. From there they can go to the inventory page again to see the new inventory items added to the page

**Test Steps:**

1. The admin user logs in with the specific admin credentials
2. They can then go to the Inventory page where they can see an Upload Inventory Batch option on the left side menu
3. After selecting the Upload Inventory Batch, they can see the section to browse for CSV file to upload
4. They can click browse find the CSV file they wish to upload with inventory items as well and receive a message about their upload once submitted
5. They can navigate back to the inventory page
6. While they can account for all the items they just uploaded with the CSV file and validate that the inventory can through with all included items

**Expected Result:**When the admin navigates to the Inventory page after logging in through the proper admin credentials they can see the ‘Upload Inventory Batch’ on the left side selection menu as the last option. From there they can click on the ‘Upload Inventory Batch page and see the option to browse for CSV file on their computer. Once the proper CSV file is selected then the user receives a popup to alert them after they completed the upload. The can validate the new addition of CSV file inventory items by going to the inventory page and seeing the top entries.

1. **QA Test Case Table on two browsers**

|  |
| --- |
| QA Testing Table - Browser Edge |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test# | Test Title | Test Description | Test  Input | Expected  Corrected Output | Test Results  PASS/FAIL |
| 1 | View Inventory & Inventory Items | Login, go to the inventory page and view all of the inventory items or view an individual item | 1. Username 2. Password 3. The Inventory button 4. The view button | See landing/ home page, navigate and access the inventory page see all inventory and view specific inventory items | PASS |
| 2 | Shopping for Items & Filling your cart | At the inventory page  Select items and add them to the cart  Update item quantities  Remove items from the cart | 1. Shopping cart button 2. The quantity box 3. The X remove button | See and use the inventory page to add to cart, see green shopping cart bubble changes add to cart remove from cart and update cart quantities | PASS |
| 3 | Checkout & Order Records | At the checkout page place an order by clicking the checkout button. Admin can view the order existence on the orders page  Admin can reject or approve an order | 1. Checkout Button 2. Orders Button 3. Approve Button 4. Reject Button | Place order, see the confirmation page, admin can view orders, reject or approve orders and change their status and filter the results | PASS |
| 3 | Uploading Files | At the Inventory page an admin can see the Upload Inventory Batch and click on it. At the upload batch page they can upload a CSV file and go to the inventory page again to see the new items added | 1. The Inventory Button 2. The Upload Inventory Batch Button 3. Browse Button 4. Ok Button 5. Inventory Button to validate changes | See the Upload Inventory Batch option on the Inventory page and upload a csv file that shows the upload inventory items in the Inventory table | PASS |

|  |
| --- |
| QA Testing Table - Browser Chrome |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test# | Test Title | Test Description | Test  Input | Expected  Corrected Output | Test Results  PASS/FAIL |
| 1 | View Inventory & Inventory Items | Login, go to the inventory page and view all of the inventory items or view an individual item | 1. Username 2. Password 3. The Inventory button 4. The view button | See landing/ home page, navigate and access the inventory page see all inventory and view specific inventory items | PASS |
| 2 | Shopping for Items & Filling your cart | At the inventory page  Select items and add them to the cart  Update item quantities  Remove items from the cart | 1. Shopping cart button 2. The quantity box 3. The X remove button | See and use the inventory page to add to cart, see green shopping cart bubble changes add to cart remove from cart and update cart quantities | PASS |
| 3 | Checkout & Order Records | At the checkout page place an order by clicking the checkout button. Admin can view the order existence on the orders page  Admin can reject or approve an order | 1. Checkout Button 2. Orders Button 3. Approve Button 4. Reject Button | Place order, see the confirmation page, admin can view orders, reject or approve orders and change their status and filter the results | PASS |
| 3 | Uploading Files | At the Inventory page an admin can see the Upload Inventory Batch and click on it. At the upload batch page they can upload a CSV file and go to the inventory page again to see the new items added | 1. The Inventory Button 2. The Upload Inventory Batch Button 3. Browse Button 4. Ok Button 5. Inventory Button to validate changes | See the Upload Inventory Batch option on the Inventory page and upload a csv file that shows the upload inventory items in the Inventory table | PASS |

1. **Coding Review**

**Our Coding Style**

The coding style employed in the Milestone 4 project is based on many conventional practices and approaches in accordance with the syntax of the various used programming languages: PHP, JavaScript, Html, and CSS. In general terms we organize the code itself by combining both the PHP, Html and reference calls and code to our separate CSS and JavaScript documents in each of PHP code document. This setup allows aspects of our code to be traced back and easily manageable on the whole. The PHP in particular used in the given code segment for the placing orders and shopping feature discussed in the QA and usability testing sections conveys those practices.

We use PHP with the conventional syntax strategies such as indenting code and using white space for readability, tiering our code, using <?php ?>, commenting and using meaningful names, using single and double quotes appropriately as well as safety and error handling practices and much more to characterize and functionalize the code to render meaningful results. We use JavaScript references within the PHP in a responsible way as well by using JavaScript objects properly, use apposite statement coding conventions, proper code indentation with functions and naming conventions as well as good use of accessing html elements. Additionally, we use html to create interface visuals by apply conveying the correct document types for the html sections, properly closing and usefully indenting Html elements along with using lower case attribute names and good equal sign and spacing strategies as well as placing in document roots properly. Lastly, we additionally utilized CSS styling to harmonize the code by properly decomposing and sectioning the stylesheet grouping styles on their own line, combining it appropriately with the html, minimizing redundancies, and keeping selector specificity low comprised of the style approach we used with the example given. This code and the other types used provide the detail the behind the scenes programming for the inventory, shopping Cart, addToCart, RemoveItem PHP and other supporting documentation that enabled the given example. Overall, the coding style is efficient, clear organized and achieves its intended goal.

**Correspondence & Code Review**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Bradley Rubenstein** <bradley@rubenstein.net> | | Attachments3:44 PM (14 minutes ago)  https://mail.google.com/mail/u/0/images/cleardot.gif |  | **https://mail.google.com/mail/u/0/images/cleardot.gif**  **https://mail.google.com/mail/u/0/images/cleardot.gif** |
| |  | | --- | | to me  https://mail.google.com/mail/u/0/images/cleardot.gif | | | |

Hey Alicia!

Overall the code looks great, I didn't find any mistakes. The style is fairly consistent and easy to read, however I think we can edit a few things to increase clarity. There are also a few things I wanted to add to the code, that I think we included in some of the other functions. You can find all my suggestions as comments on the code, which I attached. If you have any questions, let me know!

Brad

https://ci6.googleusercontent.com/proxy/RnNZfQn2o2xpggJQqefCOervMbPIci5mujDPJnvl43kv6Rtxjyh5gHN_JKVzeU-aaGz3pePFgxfoAAtZJZNx8mveVTc-11j98EfuAJVcumUenA=s0-d-e1-ft#https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif

On Sat, Apr 14, 2018 at 12:08 PM, Alicia Mitchell <[amitch41@fau.edu](mailto:amitch41@fau.edu)> wrote:

Dear Bradley,

Below please find the Code Review M4 Instructions Please be so kind to provide commentary about code relating to the tested feature and review the style and approach employed.

Thank you so much for your help with this matter.

Sincerely,

Alicia

2.5 Code Review By now you should have chosen a coding style.

In the report state what coding style you chose. Chose the code (substantial portion of it) related to the feature you used for QA and usability test. You need to submit an example of the code (or part of it -2 pages or so MAX ) for its function to be peer reviewed, and document this as follows:

1) One team member should submit code to another team member(s) for peer review.

2) Peer review should be performed by other group member(s) (1 review is OK).

3) Peer review is to be done by e-mail and comments are to be included in the code

4) Submit listing containing the peer review and commented code and communication related to this in your Milestone 4 document Important: It is critical that code reviews are friendly and helpful, intended to help and education, and not to criticize. It is strongly suggested that you use peer review in the development of the whole system.

<?php

session\_start();

$servername = "localhost";

if(isset($\_SESSION["accNumber"])){

$userID = $\_SESSION["accNumber"];

try{

$username= "CEN4010\_S2018g09";

$password= "cen4010\_s2018";

$conn = new

PDO("mysql:host=$servername;dbname=CEN4010\_S2018g09",trim($username),trim($password));

$conn->setAttribute(PDO::ATTR\_ERRMODE,PDO::ERRMODE\_EXCEPTION);

$stmt = $conn->prepare("SELECT \* FROM ShoppingCart ");

$stmt->execute();

$flag=$stmt->setFetchMode(PDO::FETCH\_ASSOC);

$result = $stmt->fetchALL();

}catch(PDOException $e){

echo "Connection failed: " . $e->getMessage();}

**//Excellent use of fail error messages to verify when the connection is not established correctly**

}else

{

$link = "signin.php";

echo "<script>window.location = '$link'; </script>";

**//Would it be possible to include something that also checks if the shopping cart is empty? I know we have something similar that checks if the inventory is empty.**

}

?>

**//When we connect to the database, we should also be checking the account type.**

$items = []; $orders = [];

$first = true; $temp = "";

**//We might be able to improve our naming conventions. What does temp represent? What does number represent?**

for($i=0;$i<count($result);$i++)

{

$row = $result[$i];

$number =$row['Number'];

if ($temp != $number)

{

if ($i+1==count($result)){

array\_push($orders,$items);

$items = [];

array\_push($items,$row);

}

if ($first != true )

{

array\_push($orders,$items);

}

$items = [];

$first = false;

}elseif($i+1==count($result)){

array\_push($items,$row);

array\_push($orders,$items);

}

$temp = $number;

array\_push($items,$row);}

**//Everything is correct, however you should be more consistent with the placement of your brackets**

1. **Best Practices for Security**

**a.) List major assets you are protecting**

**User sensitive data**

**Username:**

This information asset is essential to protect to ensure that accounts no matter of which type cannot accessed by any other user the credentials of the user are tested upon login and must match in order to allow access to the site. This must be combined with the correct password to achieve successful site access. We intend to further protect the user credentials and amend the asset protection after single sing in is been implemented with the professor and universities assistance.

**Password**

This asset was a prioritize asset to protect as well and measures were taken to validate the user’s credentials before giving them site access and must combine properly with the username credential. This is as it is done conventionally hidden at the login page itself and cannot be seen as its typed in. We intend to further protect the user credentials and amend the asset protection after single sing in is been implemented with the professor and universities assistance.

**All other account information**

The private information about the student’s class information, name, email should all be protected from any unauthorized user access this information is only visible upon the successful login with correct corresponding username and password credentials.

**Admin & staff special access permissions:**

The ability to access staff and admin exclusive information by only those intended users is paramount and is protected in such a manner to allow certain views of the website based on the user account. Only admin and staff can view profile or vendor and all order information and those with student accounts cannot see that information. The credentials they provide and profile type determine and provide them specific access to certain site pages and more sensitive information.

**b.) Confirm that you encrypt password in the DB**

Professor Haung stated that this could be done in the M5 iteration and was not required to be complete in this iteration.

**c) Input data validation**

**Code Excerpt**

<?php

session\_start();

$servername = "localhost";

if(isset($\_SESSION["accNumber"])){

$userID = $\_SESSION["accNumber"];

try{

$username= "CEN4010\_S2018g09";

$password= "cen4010\_s2018";

$conn = new PDO("mysql:host=$servername;dbname=CEN4010\_S2018g09",trim($username),trim($password));

$conn->setAttribute(PDO::ATTR\_ERRMODE,PDO::ERRMODE\_EXCEPTION);

$stmt = $conn->prepare("SELECT \* FROM Inventory");

$stmt->execute();

$flag=$stmt->setFetchMode(PDO::FETCH\_ASSOC);

$result = $stmt->fetchALL();

}catch(PDOException $e){

echo "Connection failed: " . $e->getMessage();}

if(count($result)==0){

$link = "index.php";

echo "<script>window.location = '$link'; </script>";

}

}else

{

$link = "index.php";

echo "<script>window.location = '$link'; </script>";

}

?>

**Our Input Validation Approach & Objectives**

We understand the importance of validating the input entered in by the user into a given input box. The above code excerpt highlights our effort to ensure that the user cannot access the site without having valid credentials entered into the username and password fields. Meaning if the user enters the password and username incorrectly then they can not gain access to the site. This input validation measure ensures the integrity and security of users who attempt to access the Hardware nanny site cannot put credentials with errors but only their unique exact login credentials.

Currently we are developing further credential validation measures with other aspects of our software including: the new registration input validation to ensure the new user enters in the required fields with the proper data. We will use red highlighted text to advise on the correct data needed for input when input is entered incorrectly and prevent the user from registering without proper input in the desired format. We also are striving and in development to create input validation red text and only allowing submission with correct data with our add new item and add new vendor features. Additionally, we will setup input validation for the quantity box for users shopping for parts and for users that edit or rather make changes to their account.

Input validation is essential strategy which has been utilized in our login page beta delivery. Doing so guides the user to provide correct input and gives our software additional security from users that enter bad data to gain access or use services that our software provides. All in all, we understand the merit of programming in this fashion and have done so to a degree and anticipate doing more in the future.

1. **Self-Check on Non-functional Requirements**

### List of high level non-functional requirements

**Usability**

* 95% of all users will be satisfied with the usability of the product. ON TRACK
* The system must be easy to use by both Perry, Students, Faculty and have minimal to no issues when used and is operable for a majority of users without a need for application support documentation. ON TRACK
* The system must be quickly accessible to Perry Students Faculty & other authorized personnel on authorized computer terminal workspaces. ON TRACK
* The system must be intuitive and simple in the way it displays all relevant data and relationships. DONE
* The menus of the system must be easily navigable by the users with buttons that are easy to understand. DONE
* The Hardware Nanny System should operate using all faculty, students and admin operating systems and function and install properly. DONE
* Students, staff and admin data is taken for all services used to monitor frequency enabling billing for outside departments and overall monitor services used over time. ISSUE \*Billing measures will not be implemented
* Recommender systems or recommendation systems are a subclass of information filtering system that seek to predict the 'rating' or 'preference' that a user would give to an item. ISSUE \*Decision to develop feature is not been determined may save for M5

**Reliability**

* The System must give accurate inventory status to the user continuously. Any inaccuracies are taken care by the regular confirming of the actual levels with the levels displayed in the system. DONE
* The system must provide a password enabled login to the user to avoid any foreign entity changing the data in the system. DONE
* The system will provide the user updates on completion of requested processes and if the requested processes fail, it should provide the user the reason for the failure. ON TRACK
* The system should not update the data in any database for any failed processes. ON TRACK
* In the event of the user cancelling or quitting the process “updating part information” any changes made by the user will be reversed. ISSUE \*Requires further testing and refinement
* The system is required to support multiple terminals simultaneously. The system should handle reasonable number of users without break or inconsistency. ON TRACK

**Performance**

* The system must not lag, because the students, faculty and the admin using it don’t have down-time to wait for it to complete an action. ON TRACK
* The system must complete updating the databases, when a user creates an account, when students orders or rents parts, when staff adds items to inventory or adds a new vendor when admin create/edit staff accounts, create/edit customer accounts successfully every time the user requests such a process. ISSUE \*Requires further testing and refinement
* All the functions of the system must be available to the user every time the system is turned on. ON TRACK
* The calculations performed by the system must comply according to the norms set by the user and should not vary unless explicitly changed by the user. ON TRACK

**Supportability**

* The software is designed such that it works even on systems having the minimum configuration. ON TRACK
* The system is adaptable even if additional plugins or modules are added at a later point. ISSUE \*requires further testing and refinement
* The data can be imported by the admin so as to make the system more portable. ON TRACK
* The system should be operational on all designated University computers terminals and accessible off campus with specific off campus restrictions including a different ip address that distinguishes the user. ISSUE \*not implemented completely yet waiting for single sign in setup

**Security**

* Login requirements: an account must be established with university email and credentials and verified during the account creation process. ISSUE \*could not implement feature without single sign and University/Professor assistance
* Password changes and forgotten password recovery are permissible to all users

ISSUE\*could not implement feature without single sign in and University/Professor assistance

* For security purposes all users must sign in with email and password after 4 attempts the system locks the user out and has the account owner reset their password and when doing so has them answer a security question. ISSUE\*could not implement feature without single sign and University/Professor assistance
* Password requirements: passwords must be at least 8 characters in length, cannot be based on dictionary words/common names, and must contain at least 3 of the following 4 types of characters: lowercase letters (i.e. a-z), uppercase letters (i.e. A-Z), numbers (i.e. 0-9), special characters (e.g. -=[]\;,./~!@#$%^&\*()\_+{}|:<>?) ISSUE



* Inactivity timeouts: a user of the system is automatically logged out of the software after a period of 15 minutes of inactivity and needs to log back in as they are notified of the software timeout. ISSUE university must provide details for this

\*could not implement feature without single sign and University/Professor assistance

* In order to distinguish on campus and off campus users 2 different designs will be used to differentiate users with ip addresses in campus and out of campus. Restricting certain functionality based the type of ip address design used when software system is accessed. ISSUE \*could not implement feature without single sign and University/Professor assistance
* DoS attack protection strategies will be employed such as DoS attack identification and detection techniques to differentiate between legitimate and malicious traffic. Using activity profiling procedures and mechanisms which measure average traffic rates and flags significant increases in traffic to identify when an attack is underway. The use of throttling and rate-limiting technologies will be used to reduce the effects of a DoS attack including such tactics as a response mode that stops all new inbound connections in the event of a DoS attack, allowing established connections and new outbound connections to continue. ISSUE\*could not implement feature without single sign and University/Professor assistance

**Storage**

* Throughput storage of the web store application should have approximately 5 megabytes. ISSUE\*advised to estimate and require University/Professor assistance
* The bandwidth to be determined must meet approximately 100Mbps per 1,000 users or 100Kbps and will require 1Gbps per 1,000 students or 1 Mbps per user ON TRACK
* The system should be able handle at least the total student population that have accesses to the system making requests special jobs and products per hour.

ISSUE\*advised to estimate and require University/Professor assistance

* A dedicated cloud server is used to store data with at least 8 GBs of storage space.

ISSUE\*advised to estimate and require University/Professor assistance

**Fault tolerance**

* Recovery process can be initiated by the admin personnel this process is prompted to executed during instances when the security has of the software has been compromised ISSUE \*Requires further testing
* Recovery time depends on the database size growth potential approximately a 1 hour to 2 hours to completely analyze the system for failures or security compromises ON TRACK

Backup frequencies: occur every week, or at the end of every day during peak times with new shipments or large request orders are placed. ISSUE \*Requires further testing and University/Professor assistance

**Constraints**

* The Hardware Nanny Software Systems is most influenced by the admin and faculty that regulates various system functions at their discretion and Its database is moderated by both admin and faculty the admin in particular manages all customer accounts and requests and overall inventory. DONE
* Student users are given limited access and retain the most restriction access when using the specialized inventory management software. ON TRACK
* Monetary transactions are not made through the systems only data about the records of requests are collected to reflect request and service use history. DONE

**Implementation**

* The System User Interface and Software is built on Adobe Brackets version 1.12 using Bootstrap PHP, JavaScript HTML, & CSS. DONE
* The Programing is done in on Adobe Brackets version 1.12 & Visual Studios Code. DONE
* The Database is implemented on the phpMy admin & MySQL DONE