**Code Inspection Report**

UMBC Bazaar

**Client**

Ericka Marable

**Team 3: UMBC Obsidian**

Gene Burchette

Jerson Guansing

Karl Justice

Darren Stevens

Victor Wu

Last Edited: 11/29/2016

*UMBC Bazaar*

Code Inspection Report

#### **Table of Contents**

1. Introduction 3

1.1 Purpose of This Document 3

1.2 References 3

1.3 Coding and Commenting Conventions 3

1.4 Defect Checklist 3-6

2. Code Inspection Process 6

2.1 Description 6

2.2 Impressions of the Process 6-7

2.3 Inspection Meetings 7

3. Modules Inspected 8

4. Defects 12

Appendix A – Peer Review Sign-off 14

Appendix B – Document Contributions 15

**1. Introduction**

1.1 Purpose of This Document

This document’s purpose is to record the procedure and results of the coding and review of the UMBC Bazaar application. The practices being documented include coding and commenting conventions used, software defects found, and system reviews made.

1.2 References

1. *UMBC Bazaar* System Requirements Specification
2. *UMBC Bazaar* Software Design Document
3. StackOverflow.com

1.3 Coding and Commenting Conventions

For the portions of the project in PHP, HTML, and CSS, common CamelCasing was used for variable names, and stacking tab-spacing for line delineation (most of the coding was done on notepad++ so tabs are equivalent here to 4 spaces).

1.4 Defect Checklist

Defects in code can cause bugs and make it difficult to maintain. Possible defects range from logic and programming errors to missing comments. We categorized our defects into the following categories: coding convention errors, logic errors, security oversights, and commenting errors. Please view the table below, table 1, for a detailed list of defects we look for when reviewing code. Our list of found defects is found in table 2. For information about specific tests completed please see the Testing Report Document.

Table 1. Defect Categories

|  |  |
| --- | --- |
| **Category** | **Comments** |
| Coding Convention Errors | Coding convention errors are when code deviates from the acceptable norms. This makes code difficult and is not good for application evolution. |
| Logic Errors | When code does not have the desired outcome. |
| Commenting Errors | Comments are not consistent, insufficient, missing, or incorrectly describe the code indicated. |
| Security Oversights | Security oversights are vulnerabilities in code. They provide entry points for hackers and others to take advantage of your application in ways not originally intended and may inadvertently provide access to the underlying computer system. |

Table 2. Defect Checklist

|  |  |
| --- | --- |
| **Category** | **Comments** |
| Logic Error | Parameters in the wrong order |
| Logic Error | Incorrect Method Called |
| Logic Error | Array index out of bounds |
| Logic Error | Objects compared incorrectly (case sensitivity) |
| Logic Error | Mismatched parentheses and precedence |
| Security Oversight | Exceptions not handled properly |
| Security Oversight | Error displayed to user |
| Security Oversight | SQL injection vulnerability |
| Security Oversight | Cross-site scripting vulnerability |
| Security Oversight | Format string vulnerability |
| Security Oversight | Remote code execution |
| Security Oversight | Incorrect URL access restriction |
| Commenting Error | Missing comments |
| Commenting Error | Vague comments |
| Commenting Error | Too many comments |
| Commenting Error | Incorrect comments |
| Coding Convention Error | Variable names do not follow the naming convention |
| Coding Convention Error | Method names do not follow the naming convention |
| Coding Convention Error | Class names do not follow the naming convention |
| Coding Convention Error | Hard-coded as string when should be passed in as a variable |
| Coding Convention Error | Code duplicated instead of placed in a method |

2. **Code Inspection Process**

2.1 Description

Primary Code Inspection for each segment was performed by the team member writing that segment while in the process of writing that piece of code. Secondary Code inspection was performed by fellow team members remotely using Github access and Skype/GoogleDocs for communication/discussion.

This inspection process does diverge from standard and best practices (in particular those which include sitting down together in a room for more formal meetings). This is because, as college students and not office-working programmers, our schedules differed widely, and a great many more demands on our time and attention prevented a more formal process from being feasible.

2.2 Impressions of the Process

Working from a cloud-based storage system was actually really nice, the convenience and availability of the code being up-to-date at all times eliminated messy version control, making sure that when inspecting code, we were never wasting our time looking at old code that had already been fixed. With GitHub’s comment feature, it’s really easy to ask the other coders what a line is supposed to do and how to it fits into the product. Notifications expedite the lag in conversation between coder and questioner, and clarifying code quickly is critical when trying to re-use someone else’s code.

The best modules or areas of code are the ones that have been reviewed extensively by more than one person. These are the PHP files related to the login and registration modules (login.php, register.php, confirm.php, etc.) that are least likely to have remaining flaws as they have been reviewed since before the midterm presentation. In contrast, checkout.php and orders.php will more likely contain coding flaws due to their late implementation and less time spent inspecting the code.

2.3 Inspection Meetings

Table 3. Inspection meetings

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Time | Location | Team Members |
| 11/08/2016 | 12:00am - 2:00am | Virtual | Jerson, Gene |
| 11/28/2016 | 8:30pm - 12:00am | Virtual | Jerson, Karl |
| 11/29/2016 | 12:00pm - 4:00pm  8:00pm -  12:00am | Virtual | Victor, Jerson, Karl, Gene, Darren |

3. **Modules Inspected**

UMBC Bazaar is primarily written in PHP with a few supplemental files in HTML and JavaScript, and cannot be adequately described using a standard structural decomposition diagram. However, the system is mostly procedural as described in our Systems Design Documentation.

Please refer to the table below to view files with associated brief description of their functionality.

Table 4: Code Description Table

|  |  |  |
| --- | --- | --- |
| # | File | Brief Description |
| 1 | .htaccess | Configuration file for the web server that contains rules for session URI and error pages. |
| 2 | CommonMethods.php | This is a helper file that contains the most commonly used PHP functions by all other files. |
| 3 | DBConnection.php | The class abstraction that establishes the database connection and queries to the MySQL database. This file is included by all other PHP files that require database access. |
| 4 | ./\*(png|jpg) | Images used throughout the website. |
| 5 | UMBCDirectory.php | This PHP file contains a PHP function that accesses and scrapes student information given a UMBC ID from the directory page of UMBC. |
| 6 | addproduct.php | This file acts as the controller for adding, updating and viewing products or services listed on UMBC Bazaar. |
| 7 | bottom.html | HTML template file for the bottom and footer section of the webpages. |
| 8 | captcha.php | This PHP file generates a PNG image with 10 randomly generated characters as an added form submission security. |
| 9 | checkout.php | This PHP file handles the checkout process and order confirmation functionalities. It displays appropriate payment or trade options, validates the submission, and sends confirmation emails to both the buyer and seller. |
| 10 | confirm.php | This PHP file activates the user’s account after registering and clicking the link provided in the confirmation email. |
| 11 | error.php | Generic error page for errors 400-417 |
| 12 | getCourses.js | JavaScript is used for client-side form validation and JSON parsing the output from students.php to dynamically update textboxes. |
| 13 | index.php | The main page of the web application. By default, it will display all published products and services. |
| 14 | listtrade.php | This templates displays all trade options by the user. |
| 15 | login.js | This JavaScript file contains client-side code used to validate the login form. |
| 16 | login.php | This PHP file is the authentication interface of the web application. It performs IP address blocking for repeated failed logins, password hashing (SHA512 with salt), and session validation. |
| 17 | logout.php | Invalidates or destroys the current session. |
| 18 | middle.html | HTML template file in between the top section and the main section of the web pages. |
| 19 | orders.php | Not completed yet. This PHP file displays all the transactions where the user is either the buyer or seller. It displays the details about each order. |
| 20 | productDetail.php | Template for viewing more details about a product or service. |
| 21 | profile.php | This PHP file allows user to edit/update their user information including changing their password. |
| 22 | register.php | This PHP file controls the user registration process. It displays the registration form, processes the form submission, and sends user confirmation email. |
| 23 | resetpass.php | This PHP file provides an interface for users to reset their password in case they forgot. The user’s UMBC ID and username are required. |
| 24 | search.php | This PHP file serves as a template to query the database for products or services that matches a given a search keyword. |
| 25 | sellproduct.php | This template displays all the published and unpublished products or services for a given user. Also, it serves as an interface to select specific products to update or create a new. |
| 26 | students.php | Produces an application/json output that are requested through XMLHttpRequest to dynamically update on the client-side the login page based on a given UMBC ID. It calls the curl function to retrieve and scrape data from the UMBC Directory page. |
| 27 | styles123.css | Contains all the stylesheet information |
| 28 | top.php | Template file that displays the website logo, banner, and navigation bar of the web pages. |
| 29 | tradeoptions.php | This PHP file serves as an interface for users to add or logically delete trade options. |

### 

### **4. Defects**

The following table reviews the defects found in our system.

Table 5. Defects

|  |  |  |  |
| --- | --- | --- | --- |
| **Category: Defect** | **Location** | **Comments** | **Fixed** |
| Logic Error: missing line in authentication process | confirm.php | Prevented users from confirming their emails via the provided links. | Yes |
| Commenting Error: minimalistic commenting | addproduct.php | Comments are present, but very brief and minimalistic. | No |
| Coding Convention Error: some css style names not in CamelCase | styles123.css | Some of the style names do not follow the CamelCase standard | No |
| Commenting Error: missing all comments | captcha.php | File has no comments at all. | No |
| Commenting Error: minimalistic commenting | checkout.php | Comments are present, but very brief and minimalistic. | No |
| Coding Convention Error: some variable names not in CamelCase | checkout.php | Some of the variable names do not follow the CamelCase standard | No |
| Commenting Error: minimalistic commenting | confirm.php | Comments are present, but very brief and minimalistic. | No |
| Commenting Error: missing all comments | index.php | File has no comments at all. | No |
| Commenting Error: missing all comments | listtrade.php | File has no comments at all. | No |
| Commenting Error: missing all comments | productDetail.php | File has no comments at all. | No |
| Commenting Error: minimalistic commenting | profile.php | Comments are present, but very brief and minimalistic. | No |
| Commenting Error:  Minimalistic and incorrect commenting | search.php | Comments are present, but very brief and minimalistic. Some comments are rejected code snippets | No |
| Commenting Error: minimalistic commenting | sellproduct.php | Comments are present, but very brief and minimalistic. | No |
| Coding Convention Error: class names do not follow the naming convention | productDetail.php | File should instead be named productdetail.php to be consistent | No |

**Appendix A – Team Review Sign-off**

This document has been collaboratively written by all members of the team. Additionally, all team members have reviewed this document and agree on both the content and the format. Any disagreement or concerns are addressed in team comments below.

Gene Burchette \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 11/29/16

Gene Burchette

\_\_\_\_Jerson Guansing\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 11/30/2016

Jerson Guansing

\_Karl Justice\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 11/29/16

Karl Justice

\_\_Darren Stevens II\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:

Darren Stevens II

Victor Wu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 11/29/16

Victor Wu

**Appendix B – Document Contributions**

Identify how each member contributed to the creation of this document. Include what sections each member worked on and an estimate of the percentage of work they contributed. Remember that each team member must contribute to the writing (includes diagrams) for each document produced.

Gene - 1 entry in Table 4, first paragraph in 2.2, updated references

Victor - contributed to the creation and population of section 1.4 and tables 2, 3, and 4

Jerson - contributed to the creation and population of table 4