

Why Not Zoidberg?

Code Inspection Report

Version 1.0

11/29/2012

Prepared for
Ms. Lisa Matthews
math1@umbc.edu

Team Name: Planet Express
Team Facilitator - Chris Raborg
Requirements Leader - David Guldán
Design Leader - Kevin Yu
Implementation and Testing Leader - Zach Hisley
Delivery Leader - Matt Sperbeck

Why Not Zoidberg?
Code Inspection Report

Table of Contents

	<u>Page</u>
1. Introduction.....	4
1.1 Purpose of This Document.....	4
1.2 References.....	4
1.3 Coding and Commenting Conventions.....	4
1.4 Defect Checklist.....	5
2. Code Inspection Process.....	6-7
2.1 Description.....	6
2.2 Impressions of the Process.....	6
2.3 Inspection Meetings.....	7
3. Modules Inspected.....	8
4. Defects.....	9-11
Appendix B – Peer Review Sign-off.....	12
Appendix C – Document Contributions.....	13

Document Versioning Control

Version Number	Date	Changes from Previous Version
1.0	11/29/12	Introduction, Code Inspection Process, Modules Inspected, Defects, Appendices

1. Introduction

1.1 Purpose of This Document

The purpose of this document is to show the results of the code inspection process from the web application “Why Not Zoidberg?” It will highlight all of the bugs that were found, and note the process taken to find them. There might be bugs such as logical errors, some commenting issues, or security flaws. There will also be details on how the code is commented and what measures were taken to make sure that it was properly commented to the standards of the customer.

1.2 References

These are the references that we used:

- <http://www.csee.umbc.edu/courses/undergraduate/341/fall12/projects/coding-standards.shtml>

1.3 Coding and Commenting Conventions

The commenting conventions that were used are based upon the conventions stated on the UMBC CMSC 341 course website. These standards for coding and commenting assure that all code is written and commented to the highest standards and can be easily read and followed. This allows for long-term maintenance of the product that is simple and could be done by someone who did not originally take part in writing the code.

We created the code using a to allow enough security and data hiding between the different levels of users, and between the different users itself. The code, which was written in PHP, allows for fast access to different web pages, and a quick search process for a user.

1.4 Defect Checklist

#	Error Type	Description	Is Fixed?
1	Logical	Session Header Location	Yes
2	Logical	Incorporating Popular Genre php script	Yes
3	Logical	Incorporating Shopping Cart php script	Yes
4	CSS Layout	Popular Genre font issue	Yes
5	CSS Layout	Shopping cart font issue	Yes
6	Coding Convention	Incorrect variable names	Yes
7	Logical	Displaying the shopping cart only to users logged in	Yes
8	Validation	Checking if at least 1 book is added to shopping cart	Yes
9	Syntax	Setting the admin variable correctly when creating an account	Yes
10	Coding Convention	Incorrect redirection of pages	Yes
11	Logical	Incorrect randomization of books in popGenre.php	Yes
12	Logical	Forgot to include "cart.php"	Yes
13	Logical	Forgot to include "commonMethods.php"	Yes
14	Coding Convention	Lack of comments in the admin source code	No
15	Validation	Quotes inside of summary text field	No

2. Code Inspection Process

2.1 Description

The group followed a relatively simple code inspection process. The programmers for the group discussed either in person or via Skype the code that they were working on. There were several meetings held where the group would run the code and find an error, and then would inspect through all of the files that were related to the error found. When we found the error, we would write a short description down so that we would be able to document it here later. Many of these errors were found while each programmer was working on the code individually. That way when the files were syntactically correct, some output would be given, and the code could be uploaded to the repository.

Once our code was in a relatively useable state (no noticeable errors), we sent the code to the Design Lead who had all of the updated code. We started to identify and lingering bugs, caused by either logic errors or validation errors. In many cases, we were more interested in getting different programmers' code integrated together, that we overlooked some simple validation. When we got all of the code integrated, we could then carefully inspect the code for bugs and fix them accordingly.

2.2 Impressions of the Process

The code inspection process was mostly rough. A portion of the code was not yet completed therefore a full evaluation could not be fulfilled. But separate modules, such as the Popular Genre and Shopping cart module, could be tested without relying on other external components.

The popGenres.php, createAccount.php, and index-build.php were the best modules because they aren't reliant on other modules and produce no errors. The checkout.php, and addToCart.php are the modules that contain a few remaining bugs. checkout.php and addToCart.php are still missing validation for the quantity of books. The testing process could have been much more efficient if it was not mostly done by the programmers themselves.

2.3 Inspection Meetings

Meeting 1:

Date: 11/27/12

Location: (Conversation was over facebook)

Start Time: 8:00pm

End Time: 11:00pm

Participants: Dave, Chris

Roles: Dave and Chris were both acting as inspectors of their own code.

Code Units: item.php, cart.php, createAccount.php, viewAccount.php

Meeting 2:

Date: 11/28/12

Location: ITE Building

Start Time: 5:30pm

End Time: 9:30pm

Participants: Kevin, Dave, Chris

Roles: Dave was the Inspector, Kevin was the Moderator, and Chris was the Author/Scribe

Code Units: cart.php, shoppingCart.php, browse.php, store-style.css, admin.php, login.php, updateAccount.php, createAccount.php, index-build.php

Meeting 3:

Date: 11/30/12

Location: RLC

Start Time: 1:00 pm

End Time: 3:00 pm

Participants: Kevin, Dave, Chris, Matt, Zach

Roles: Kevin was the moderator, Matt and Zach were the inspectors, Chris was the scribe, and Dave was the author

Code Units: addToCart.php, checkout.php

3. Modules Inspected

The modules that we have not inspected (or finished) at this point are addToCart.php, shoppingCart.php, checkout.php. Most of these files are missing some small validation components, but the checkout.php page is still a work in progress.

Name of Module	Description	Where it fits into SDD
Admin.php	Displays the administrator page, and top sold items.	Edit Inventory
UpdateDelete.php, AdminSubmit.php	Takes care of adding, updating or removing an item from the database.	Update Item, Remove Item
CreateAccount.php, UpdateAccount.php	Displays the create account form.	Create Account
GetLogin.html, Login.php	Login form and validation. Starts the PHP session.	Login
item.php, cart.php	Classes that define items to be stored in a cart. Used to easily manipulate the shopping cart.	Shopping Cart
AddToCart.php	Adds a particular quantity of items to the cart.	Shopping Cart, Browse
AddFunds.php, confirmFunds.php	Adds/Confirms funds to a logged in user account.	Add Funds
popularGenres.php	Displays four books from genres that are popular.	Popular Genres
search.php	Searches the database for a book.	Search
browse.php	Displays a book given a ISBN.	Browse
viewAccount.php	Allows a logged in user to view their account.	View Account

4. Defects

Description:	Session Header could not be sent because header has been already set.
Module:	search.php, browse.php, viewAccount.php
Error Type:	Logical

Description:	include(popGenre.php) would not work
Module:	search.php, browse.php, viewAccount.php
Error Type:	Logical

Description:	include(shoppingCart.php) would not work
Module:	search.php, browse.php, viewAccount.php
Error Type:	Logical

Description:	Popular Genre font issue
Module:	search.php, browse.php, viewAccount.php
Error Type:	CSS Layout

Description:	Shopping Cart font issue
Module:	search.php, browse.php, viewAccount.php
Error Type:	CSS Layout

Description:	Incorrect variable names
Module:	admin.php, login.php
Error Type:	Coding Convention

Description:	Display the shopping cart to only users logged in
Module:	search.php, browse.php, viewAccount.php
Error Type:	Logical

Description:	Checking if at least 1 book is added to shopping cart
Module:	browse.php
Error Type:	Validation

Description:	Setting the admin variable correctly when creating an account
Module:	createAccount.php
Error Type:	Syntax

Description:	Incorrect redirection of pages
Module:	login.php, addFunds.php
Error Type:	Coding Convention

Description:	Incorrect randomization of books in popGenre.php
Module:	popGenre.php
Error Type:	Logical

Description:	Forgot to include "cart.php"
Module:	search.php, browse.php, viewAccount.php
Error Type:	Logical

Description:	Forgot to include "commonMethods.php"
--------------	---------------------------------------

Module:	popGenre.php, browse.php
Error Type:	Logical

Description:	Lack of comments in the admin source code
Module:	admin.php
Error Type:	Coding Convention

Description:	Quotes inside of summary text field
Module:	updateDelete.php
Error Type:	Validation

Appendix B – Team Review Sign-off

We agree as a team that we have documented the code inspection to the fullest possible extent and that we have all read the entire document and agree with the contents. If there is an issue within the team about the document, the team shall resolve the issue at the next team meeting.

Christopher Raborg, Team Facilitator

David Guldán, Requirements Leader

Kevin Yu, Design Leader

Zachary Hisley, Implementation Leader

Matt Sperbeck, Delivery Leader