CMSC 345

Software Design and Development

UMBC-CMSC 447 Section 2 Team 4
Team Awesome

Planes for Hire

Code Inspection Report

Client: John Winder

Member: Sundar Sekar <ssekar1@umbc.edu>, Tam Tran <tamtran1@umbc.edu>,
        William Cahill <wcahill1@umbc.edu>, Roberto Melgar <rmelgar1@umbc.edu>,
        Du Nguyen <du2@umbc.edu>

2/12/2015

Planes for Hire
Code Inspection Report

Table of Contents

1. Introduction

1.1 Purpose of This Document

The purpose of this document is to provide an overview of the coding practices that were adhered to in the the creation of the Planes for Hire application. Practices that will be included are coding and commenting conventions as well as any software defects, in addition to a review of all systems. Lastly, meetings will be outlined. Since no actual code has been written yet, we have no code to inspect. There will be more results and elaboration for Spiral 2.

1.2 References

1. Refsnes Data (1999-2015). w3schools.com. Retrieved from "http://www.w3schools.com/googleapi/"
2. The PH Group (2001-2015). PHP. Retrieved from "http://php.net/"
3. QuinStreet Inc (2015). SQLCourse.com. Retrieved from "http://www.sqlcourse.com/"
4. Planes for Hire System Requirements Document
5. Planes for Hire Design Document
6. Planes for Hire Testing Document

1.3 Coding and Commenting Conventions

We will be using camel case variable names and with c style coding conventions.This provides easy readability of code. We apply the formal format on php coding as well with all the comment and newlines. We also limit the code lines each page so that there are not many lines for just one simple page. There is also a short brief header comment on each page. We will use meaningful variable names that reflect their literal functionality. We use local SQL database and Apache web server on our computer for development. Database naming convention uses meaningful names that reflect their data elements and and their functionality.

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, commenting errors, database coding convention error. 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 norm. This makes maintaining code difficult and is not good for application evolution.
Logic Errors	When code does not have the outcome originally intended.
Commenting Errors	Comments are not consistent or missing.
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.
Database Convention Error	When the database isn't built to handle scalability or is unorganized.

Table 2. Defect Checklist

Category	Defect	Comments	Fixed
Logic Error	Search feature loads entire table.	In searching, all tables in database for airports are buffered in before the search begins.	No
Logic Error	Check out Duration	Once a user selects a date, they can't change it to "no days" if they change their mind.	No
Logic Error	Main page doesn't give message when no planes are available.	If a given airport doesn't have any planes, it should display an error message stating that there are not available planes.	Yes
Security Oversight	Same email for multiple users	Database should prevent user from making account in the event a user attempts to register an account with the same email as an existing user.	Yes
Logic Error	Cross platform defect	User cannot upload pictures when they are not on localhost	No
Security Oversight	Restore password for suspended accounts.	Admin cannot restore original user password when they are suspended.	No
Logic Error	Map issue	On the main page, without logging in, the map doesn't pan.	Yes
Database Convention Error	Scalability	New user tables are created for each user. This could lead to 100	Yes

		tables that consist of 100 users.	
Security Oversight	Encryption	User information not encrypted.	Yes
Commenting Error	Missing comments	Most of the files were missing comments which made it hard to read for other users.	Yes
Logic Error	Div shifting error.	The div storing forum data for checking in and out planes likes to shift to the bottom of the screen if the user resizes the screen.	No
Coding Convention Error	Profile Picture	User cannot add a picture from the Registration page	Yes
Logic Error	Paying issues	User can pay with negative dollars	Yes
Logic Error	Old dates when checking out	User is able to select a date before the current when checking out a plane.	Yes
Logic Error	Balance issues	User can have a negative balance	Yes

2. **Code Inspection Process**

2.1 **Description**

Code inspection was completed by the coder who wrote the code while simultaneously navigating through that part of the application. It was also inspected by a peer through Github. In future spirals code will be reviewed in pairs in person, with the coder who wrote the code paired with another coder.

Our team performs weekly meeting to discusses application design approach and use case implementation. We also work on project documentation, and test any previously

implemented use case design. Our meeting also involved discuss members job roles and responsibility.

2.2 Impressions of the Process

Our code inspection process has been effective but could been improved in meetings. We had a lot of issues with syncing on github when multiple users were working on the same files. During meetings, only few members showed up and contributed to the project.

We made a list of defects and things to do on github so that our team members were aware of it and were able to fix the issues. This also provided a method for Team members to easily comment on both good and bad code.

Coding process can be tedious, for instance, the avatar was difficult to implement. Source code found was used for a different platform. Had to recode file uploading process rather that using example code. Very time consuming and difficult.

2.3 Inspection Meetings

The following is the inspection meetings where codes were reviewed. There were no specify roles assigned to each team member. Everyone shared the roles and workload equally. However, Tam and Andrew lead the development process.

Table 3.

Date	Name	Location	Time
3/13/2015	Tam, Andrew, Sundar, Roberto	2nd Floor Library	10 am to 3 pm
3/23/15	Tam, Andrew, Sundar, Roberto	Math and Psychology Building Basement	6:45 pm to 7:30 pm
3/25/15	Tam, Roberto, Duke, Sundar	Math and Psychology Building Basement	6:45 pm to 7:30 pm
3/27/15	Tam, Sundar, Andrew	2nd Floor Library	9 am to 3 pm
3/30/15	Tam, Sundar, Andrew	2nd Floor Library	5:15 pm to 7: pm

4/1/15	Tam, Sundar, Andrew	2nd Floor Library	5:15 pm to 7: pm
4/3/15	Tam, Andrew, Sundar	RLC	8 am to 4 pm
4/10/15	Tam, Andrew, Roberto, Sundar	2nd Floor Library	11 am to 4 pm
4/15/15	Tam, Andrew, Roberto, Sundar	Gameroom	7 am to 8 pm
4/24/15	Andrew, Roberto, Sundar	2nd Floor Library	10 am to 4 pm

3. Modules Inspected

Planes for Hire is a javascript and php based application written with the help of GitHub and phpMyAdmin. Currently code exists to handle url paths as well as the controller logic for each view.
Please refer to the table below to view files with associated brief description of their functionality.

Table 4.

#	File	Brief Description
1	index.html	Loads Main.php
2	Main.php	Includes headHTML and tailHTML. Utilizes registration.php Utilizes login.php Utilizes logout.php Utilizes search.php Utilizes checkIn.php. Utilizes checkInResult.php Utilizes checkOut.php Utilizes userProfile.php Utilizes admin.php Utilizes planesForHire.js Utilizes planesForHire.css

		Utilizes link.php to establish link between server and database for querying.
3	registration.php	Utilizes link.php to establish link between server and database for querying. Utilizes confirm.php
4	logout.php	Destroys session cookie created by login.php
5	link.php	Class containing methods for establishing connection between server and database
6	confirm.php	Initializes session variable and appends a new user into the database.
7	headHTML	Contains planes for hire css and js.
8	tailHTML	Couple of lines of code to finish off html body of most pages.
9	picsUploads	Directory storing user profile pictures, as well as default image.
10	planesForHire.sql	Stores export of current database. Necessary for porting database to different machine.
11	populatePlaneOption.php	Called from multiple php documents in order to populate list of planes.
12	search.php	Makes search queries on database tables. Shows relevant airports.
13	planesForHire.css	Defines layout elements of the website.

14	planesForHire.js	Defines functionality of html elements on website.
15	userProfile.php	Utilizes uploadFile.php Utilizes planesForHire.js for functionality. Utilizes planesForHire.css Utilizes updateUserProfile.php
16	updateUserProfile.php	Updates user information as well as provide all other database functionality for userProfile.php
17	admin.php	Utilizes adminHead.html Utilizes adminTail.html Utilizes updateTable.php
18	updateTable.php	Provide admin functionality for users and database functions.
19	checkInResult.php	Shows weather user owes late fee or not.

4. Defects

The following table reviews the defect found in our system.

Table 5.

Category: Defects	Comment	Location	Fixed
Commenting Error: Too many comments, missing comments	Comments are inconsistent and can range from little to excessive comments.	Most files	Fixed most of the files

Logic Error: User account	Multiple users can use same email ID as their username	Database, User profile page	No
Secured network	opened network	Database link.php	No
Logic Error: Unprotected database	Easy to manipulate records in the database by an unknown user	link.php Database	No
Logic Error: Calendar	User is able to select a date before the current when checking out a plane.	Main.php	Yes

Appendix A – Team Review Sign-off

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

Team

Name _____

Date _____

Sign _____

Comments _____

_____

_____

Name _____

Date _____

Sign _____

Comments _____

_____

_____

Name _____

Date _____

Sign _____

Comments _____

_____

_____

Name _____

Date _____

Sign _____

Comments _____

_____

_____

Name _____

Date _____

Sign _____

Comments _____
                   _____
                   _____

Appendix B – Team Review Sign-off

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

Team

Name _____
Date _____
Sign _____
Comments _____
                   _____
                   _____

Name _____
Date _____
Sign _____
Comments _____
                   _____
                   _____

Name _____
Date _____
Sign _____
Comments _____
                   _____
                   _____

Name _____
Date _____
Sign _____
Comments _____
                   _____
                   _____

Name _____
Date _____

Sign _____

Comments _____

_____

_____

Appendix C – Document Contributions

Roberto wrote the purpose of the document, appendices, and the very brief description for 2.1. Sundar wrote references. He played a role in editing and reviewing the document as well as creating and assembling the documents. Tam and Sundar wrote section 1.3.

For Spiral 2, everyone updated the information.