**Code Inspection Report**

*Piece of Eden Rentals*

**Client**

John Winder

**Team 3**

Eric Forte

Rachel Cohen

Stephen Masterson

Matt Walker

Nick Keckeisen

4/11/2016

*Piece of Eden Rentals*

Code Inspection Report

**Table of Contents**

1. Introduction

1.1 Purpose of this document

1.2 References

1.3 Coding and Commenting Conventions

1.4 Defect Checklist

1. Code Inspection Process

2.1 Description

2.2 Impressions of the Process

2.3 Inspection Meetings

1. Modules Inspected
2. Defects

Appendix A - Agreement Between Customer and Contractor

Appendix B - Team Review Sign-off

Appendix C - Document Contributions

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 creation of the Piece of Eden application. It will include coding and commenting conventions, as well as any current defects in the software up to the current point. Additionally, meetings are outlined.

**1.2 References**

1. Piece of Eden System Requirements Document

2. Piece of Eden System Design Document

3. Python Style Guide

https://www.python.org/dev/peps/pep-0008/#code-lay-out

4. Defect/Bug Classification

http://www.softwaretestingstuff.com/2008/05/classification-of-defects-bugs.html

**1.3 Coding and Commenting Conventions**

We adhered to the standards described in the Python Style Guide (2013). We used 4 spaces per indentation level. CamelCase naming style were used for the naming of variables. Class names begin with a capital letter. Class and function definitions are surrounded by blank lines for readability. Other blank lines are used to separate logical chunks of code, also for readability and understanding. Uses of ‘import’ are done on separate lines and at the top of the file in which they are being used. Comments are preceded with a ‘#’ followed by a single space. Block comments utilize the “”” method. Constants are defined and written in all capital letters with underscores separating words.

**1.4 Defect Checklist**

Defects in code are problematic, as they can cause the software (or pieces of the software) to not work in the way they are intended to. Possible defects can range from logic errors caused by inadequate or ambiguous functionality in the source code, incorrect design, missing requirements, or lack of comment and documentation within the source code. The following describes the different categories of defects that we were looking for when inspecting our code.

*Category 1: Coding Standards*

Inadequate/incorrect/missing comments in the source code.

Variable/method/class names in line with standards described above.

*Category 2: Requirements*

Requirements needs are not met in the current implementation. Requirements are not clear to the reviewer.

*Category 3: Logic Error*

Missing or inadequate functionality in the source code.

*Category 4: Security Risk*

Vulnerabilities within the code.

|  |  |
| --- | --- |
| Security Risk | Validate all input |
| Security Risk | Make use of sanitized input to prevent against possible NoSQL injection |
| Coding Standards | Pass all strings in as variables instead of hardcoding in |
| Coding Standards | More comments could be used in HTML code sections |
| Security Risk | Handle potential injection of operators like $or and $where |
| Logic Error | Catch all exceptions |
| Requirements | Make sure all properties are on a designated coastline |
| Security Risk | Provide support against cross site scripting attacks |
| Coding Standards | Implement consistent use of curly braces |
| Security Risk | Should encrypt user passwords in the database |
| Coding Standards | Unclear comments |
| Security Risk | Access control/permissions checking |
| Coding Standards | For simplicity, code should not be duplicated but placed in a method for calling ease |
| Coding Standards | Broad CSS selectors for efficiency |
| Coding Standards | Group like CSS properties together |

1. **Code Inspection Process**

**2.1 Description**

Code inspections were completed by the coder themselves during the development process. Because code was uploaded through repository hosting service, Github, team members were able to inspect each other’s code and suggest or provide revisions.

**2.2 Impressions of the Process**

Due to the nature of the class and our conflicting schedules as undergraduate students, it was often difficult to find times where every team member could meet. Because of this, Github was a very useful tool in the review process, as it allowed us to see real-time changes to the code and suggest possible amendments.

Using Github and having several in-person meetings allowed us to have a fairly effective code review process. For Spiral 3, we will make it a point to have more code review meetings, as that spiral is important for cleaning up code and presentation of the web application.

**2.3 Inspection Meetings**

During inspection meetings code will be reviewed.

Table 1. Meeting Times

|  |  |  |
| --- | --- | --- |
| **Date** | **Time** | **Attendance** |
| April 6, 2016 | 5 PM | Eric Forte, Matthew Walker, Stephen Masterson, Rachel Cohen |
| April 11, 2016 | 5 PM | Eric Forte, Matthew Walker, Nick Keckeisen |
| April 13, 2016 | 5 PM | Eric Forte, Matthew Walker, Stephen Masterson, Nick Keckeisen, Rachel Cohen |
| April 20, 2016 | 5PM | Eric Forte, Matthew Walker, Stephen Masterson, Nick Keckeisen, Rachel Cohen |
| April 27, 2016 | 5PM | Eric Forte, Matthew Walker, Stephen Masterson, Rachel Cohen |

1. **Modules Inspected**

Piece of Eden is a Python based web application written with the Django web framework. Django follows the model-view-controller architecture and breaks code up into three segments. Code Exists to handle URL paths as well as the controller logic for each view. The table below explains each of the view files along with a brief description.

|  |  |
| --- | --- |
| **File** | **Description** |
| mysite/\_init\_.pyc | Initializes Python packages |
| mysite/settings.py | Includes the settings and configuration of our Django project. Contains module-level variables representing Django settings. |
| mysite/urls.py | Includes the URL declarations for the Django project. Matches URL paths to views. Creates patterns that matches the URLs. |
| mysite/wsgi.py | Configures module-level variable named “application” |
| mysite/views.py | Holds the logic of the application. Acts as the controller for the framework. |
| mysite/manage.py | Command line utility that allows interaction with the Django project |
| mysite/beach\_homepage/admin.py | Creates the admin interface where trusted users can manage content on the site |
| mysite/beach\_homepage/models.py | Contains essential data fields as a description of the data in the database |
| mysite/beach\_homepage/tests.py | Test cases |
| mysite/beach\_homepage/urls.py | Contains URL declarations for the beach homepage |
| mysite/beach\_homepage/views.py | Takes a web request for an index and returns a web response |
| mysite/login/admin.py | Creates the admin interface for the login |
| mysite/login/forms.py | Creates a Registration form |
| mysite/login/models.py | Contains essential data fields specific to login |
| mysite/login/tests.py | Login test cases |
| mysite/login/views.py | Takes a web request to register and returns a web response |
| mysite/templates/base.html | The base template, extended by all other templates to include the content that is necessary for every page or view. |
| mysite/templates/index.html | Web server default page |
| mysite/templates/left-sidebar.html | Creates the left sidebar |
| mysite/templates/left-sidebar.html | Creates the right sidebar |
| mysite/templates/no-sidebar.html | No sidebar |
| mysite/templates/list\_new\_property.html | Allows the user to list a new property and search for properties |
| mysite/templates/property\_search.html | Allows the user to search for a property |
| mysite/templates/user\_profile.html | Creates a user profile |

1. **Defects**

|  |  |  |  |
| --- | --- | --- | --- |
| **Category:Defect** | **Location** | **Comments** | **Fixed** |
| Coding Convention | list\_new\_property.html | Lacking comments | No |
| Coding Convention | views.py | Naming convention for variables not consistent | Yes |
| User Friendliness | views.py | Could add custom 404 error page | No |

1. **Appendix A - Agreement Between Customer and Contractor**

The customer agrees to Piece of Eden Rentals with the capabilities listen in the System Requirements Specification Document. Additional features will be provided in future development spirals. When and if future changes to this document occur, a newly drafted document will be created and presented to the client for review and approval.

**Client** (print name in first blank, signature in second blank)

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

**Team** (print name in first blank, signature in second blank)

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Appendix B - Team Review Sign-off**

All team members have reviewed this document and agree both on the content and the format. Any concerns are addressed in the comment section below.

**Team** (print name in first blank, signature in second blank)

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Appendix B - Document Contributions**

Rachel Cohen created this document in collaboration with other teammates.