CMSC 447: Software Design and Development

**UMBC Parking Pal**

**Code Inspection Report**

Customer:

Katherine Gibson

Team: UMBC Construction Workers

Abbie Minor

Constantin Koehler

Braxton Dubin

Naomi Schumacher

Sarah Kirby

UMBC Parking Pal

Code Inspection Report

**Table of Contents**

[**1. Introduction**](#_ibnzcto67nl1) **2**

[1.1 Purpose of This Document](#_87rg6pmjbhjd) [2](#_87rg6pmjbhjd)

[1.2 References](#_uu6mtccq5hvn) [2](#_uu6mtccq5hvn)

[1.3 Coding and Commenting Conventions](#_fl9bbqb2rxgh) [2](#_fl9bbqb2rxgh)

[1.4 Defect Checklist](#_nu75zzcvrnph) [2](#_nu75zzcvrnph)

[**2. Code Inspection Process**](#_qlczvqhyh4nj)[**3**](#_qlczvqhyh4nj)

[2.1 Description](#_2yojlow9b7v3) [3](#_2yojlow9b7v3)

[2.2 Impressions of the Process](#_eqtzxoozmrng) [4](#_eqtzxoozmrng)

[2.3 Inspection Meetings](#_yb0kelp5k5m6) [4](#_yb0kelp5k5m6)

[**3. Modules Inspected**](#_c0dqrju4v21v)[**5**](#_c0dqrju4v21v)

[**4. Defects**](#_xp7wljt3j37u)[**7**](#_xp7wljt3j37u)

[**Appendix A – Team Review Sign-off**](#_kziiks8yfjyr)[**9**](#_kziiks8yfjyr)

[**Appendix B – Document Contributions**](#_3wu2bajqea0x)[**10**](#_3wu2bajqea0x)

# 1. **Introduction**

## 1.1 Purpose of This Document

The purpose of this document is to outline the code commenting conventions and application of those conventions by the members of the UMBC Construction Workers. This document is intended for the members of this team and our client to help find any issues this project may have. This Report also outlines the Code Inspection done with respect to the Code Inspection Process formed by the team. The Code Inspection aided in the discovery of defects listed in further sections.

## 1.2 References

The UMBC ParkingPal System Requirements Specification

The UMBC ParkingPal System Design Document

ISOBAR Code Standards <https://isobar-idev.github.io/code-standards/>

## 1.3 Coding and Commenting Conventions

We based our coding convention on ISOBAR coding standards, with some adjustments. First, we had to make some adjustments to fit the Meteor framework. For example, Meteor forced us to solely use Class tags to reference our HTML elements in CSS, as opposed to using the ID tags. Another key difference is how Meteor requires the delineation of HTML, JavaScript, and CSS code into separate files. Each page must define a JavaScript template, which then includes the HTML and CSS files. The template structure in Meteor is also hierarchical. There is one main template that the other templates, or “views”, inherit from. This template hierarchy made it easier for us to create a consistent user interface.

In addition to making adjustments to use the Meteor framework, we made adjustments to use the Bootstrap CSS framework. We utilized many of the Bootstrap CSS components, including the grid layouts. Our CSS code therefore follows the naming conventions and standards of the Bootstrap framework.

Finally, we also did not follow the ISOBAR coding standards that relate to responsiveness and mobile development. The ISOBAR coding convention “mobile first” development, but because of the nature of our app and our relative inexperience we thought it best to make desktop our first priority for the user interface.

## 1.4 Defect Checklist

These list below details possible defects we used in the code inspection process. These are not necessarily what is in our product; rather, they are a list of defects that could potentially be found during the inspection process and ones that we have previously encountered in code inspection.

|  |  |
| --- | --- |
| **Defect** | **Category** |
| Inconsistent naming standards | Coding Convention Error |
| Not very thorough documentation | Commenting Error |
| Invalid/irrelevant documentation | Commenting Error |
| No checks to see if the user is logged in | Security Oversight Error |
| Input ranges don’t match the range specified for a field in the database | Validation Error |
| CSS overlaps and creates an unintuitive layout | Styling Error |
| Backend doesn’t receive data as expected | Validation Error |
| CSS does not scale well to mobile | Style Error |
| Sensitive information isn’t hashed or otherwise secured | Security Oversight |
| Numbers are hard-coded | Coding Convention Error |
| Indentation is inconsistent | Coding Convention Error |
| Syntax errors exist | Syntax Error |
| Input types and database field types don’t match | Validation Error |
| A module doesn’t subscribe to the necessary databases | Logic Error |
| Code is not readable | Coding Convention Error |
| JavaScript, HTML, and CSS are not kept in separate files | Coding Convention Error |

# 2. **Code Inspection Process**

This section details the code inspection process, the impression of the process, and meeting details for the code inspections.

## 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 who pulled the code on their machine and inspected the functionality. By having more than one person working on it, we were able to check each other’s work.

Code was often reviewed remotely with concerns put into a group chat with all the team members where we could all chime in to discuss best practice and ensure no one is lost. We only had the opportunity to meet for a reasonable period of time as a whole group twice. Code inspection also occurred as described in the paragraph above, often with the page of the web app being put on a large TV and the code on a laptop hooked up to it, where we discussed the code and its functionality.

## 2.2 Impressions of the Process

The process works well enough. Starting off, there was a substantial dichotomy in team members’ familiarity with web development technologies. Ideally, we would all have around the same level of comfort with Javascript, which is the primary language used. However, since UMBC doesn’t offer such a course, this wasn’t the case at all. For this reason, much of the review process felt like a crash course in Javascript and HTML rather than improving on existing skills.

Combined with the fact that we did not have much time to meet in person, we found the process to be too slow. We will certainly have to meet after this document is due to ensure the product has as few flaws as possible.

We feel we are about 80% done with the project at this point with much of the flaws still needing to be worked out in the transaction phase. The module containing transaction and exchange based code would be considered the worst in terms of likelihood of remaining flaws. Additionally, the filtering features of the application pose high probability of flaws. Thankfully, we feel very good about the logging in/authentication aspect of our web app as it ties in very nicely with our framework. We would consider this the best unit of our code so far and believe it has the least likelihood of flaws persisting.

## 2.3 Inspection Meetings

**Inspection #1: November 12th 2PM-5PM UMBC RLC**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Code Units** | **Author** | **Inspector** | **Reader** | **Moderator** | **Scribe** |
| Account page,  Ratings page,  Register page | Abbie | Sarah | Naomi | Braxton | Constantin |
| Login,  Parking Database | Constantin | Naomi | Abbie | Braxton | Sarah |

**Inspection #2: November 19th 10AM-4PM UMBC RLC**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Code Units** | **Author** | **Inspector** | **Reader** | **Moderator** | **Scribe** |
| Students Database | Sarah | Constantin | Abbie | Naomi | Braxton |
| Search page | Abbie,  Constantin | Sarah | Constantin | Naomi | Braxton |
| Sell page | Constantin,  Sarah | Abbie | Braxton | Naomi | Sarah |
| FAQ page | Naomi | Abbie | Braxton | Constantin | Sarah |

# 3. Modules Inspected

**Module name:** Parking Database

**Functionality:** Stores information for each listed parking space.

**Design:** Part of the backend. Front end pages that display, edit, or add listings interface with this database.

**Differences between SDD/UID and actual code:**

* Added “am/pm” field instead of using military time to make it easier to interface with HTML forms.
* Added a “description” field so a user can describe where their parking spot is instead of selecting it on a map. This was a trade off between accuracy and ease of implementation.
* There was a naming change. The database was referred to as the “Listings” collection in the SDD, and was implemented as “parkingSpaces.js”

**Module name:** Students Database

**Functionality:** Stores information for each user/student.

**Design:** Part of the backend. Pages that display, edit, or add users interface with this database.

**Differences between SDD/UID and actual code:**

* Added “Name” field for student’s name.
* Added “total ratings” field, so users can see both the average rating and number of ratings for a user. This was added for the customers to better gauge a user’s reliability.
* There was a naming change. The database was referred to as the “Users” database in the SDD, but was changed to “students.js” because of a naming issue in meteor.

**Module name:** Login

**Functionality:** Allows a user to login using myUMBC credentials.

**Design:** All other pages use the login to determine which students’ information to display and update in the Students database.

**Differences between SDD/UID and actual code:** None

**Module name:** Register Page

**Functionality:** New users can register for an account and enter in all necessary account details.

**Design:** This page inserts a new student into the Students database.

**Differences between SDD/UID and actual code:** None

**Module name:** Account page

**Functionality:** User can modify their account settings (e.g. driver and car information).

**Design:** This page updates a student’s information in the Students database.

**Differences between SDD/UID and actual code:**

* Now a form instead of having edit buttons on the side.
* Permit type now a dropdown.

**Module name:** Search Page

**Functionality:** User can browse all active listings, view a map showing all lots, and filter listing results.

**Design:** This page displays elements in the Parking database.

**Differences between SDD/UID and actual code:**

* Listings show up on the search page instead of on a new page. This is for a better user experience, so the user can edit their search criteria without navigating back.

**Module name:** Sell Page

**Functionality:** User can post a listing.

**Design:** The form inserts a new listing to the Parking database, and updates the Student database.

**Differences between SDD/UID and actual code:**

* The page now includes a map and a description field.

**Module name:** Rating Page

**Functionality:** Allows buyers and sellers to rate each other after an exchange.

**Design:** Part of the purchase/exchange confirmation. Connected to the Students database to update a student's rating.

**Differences between SDD/UID and actual code:**

* The rating is initialized to 0 and the range of rating is 1-5 (originally initialized to -1 and range 0-5).
* A separate page after the parking exchange has been made with stars instead of a dropdown added.

**Module name:** FAQ Page

**Functionality:** Gives users a page for help and instructions on application features.

**Design:** Purely front end, linked from homepage.

**Differences between SDD/UID and actual code:** None

**Modules not completed:**

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Description of functionality** | **Est. completion** |
| Welcome Screen | Home page for users not logged in | Dec 1 |
| Homepage | Home page after a user logs in | Dec 1 |
| Buy/Sell Confirmations | Pop-up that confirms a purchase or listing. | Dec 1 |

# 4. **Defects**

Below is the list of all defects we have found during the inspection process.

|  |  |  |
| --- | --- | --- |
| **Module** | **Description** | **Category** |
| Sell Page | Form goes through if you submit parking space while not logged on. Gets blocked on the backend and doesn’t make it into DB. However, no error is given to user indicating they’re not logged in. | User friendliness |
| Search Page | No search for login state when buying parking spaces. | Correctness |
| Search Page | Does not have ability to sort listings | User friendliness |
| Account Page | Does not update user’s account information | Correctness |
| Register Page | Does not save permit type or have email | Correctness |
| Homepage | User can have more than one listing active | Correctness |
| Register Page | User isn’t prompted to register at first login | Correctness |
| Search Page | Date displayed in the listings are not user-friendly | User friendliness |
| Account Page | Rating system isn’t implemented into User Profile | Correctness |
| Homepage | User’s own listings have the buy button on them | User friendliness |
| FAQ Page | The layout is clunky and no longer follows the previous format due to changes in style | User friendliness |
| Homepage | Cannot close your own listing early | Correctness |
| Sell Page | No confirmation pop-up to display when a listing has been put up | User friendliness |
| All | Missing correct comments | Coding Convention, Commenting |
| Navbar | User’s name is displayed in a low contrast typeset in the navbar after login | User friendliness |

# 

# Appendix A – Team Review Sign-off

All members of the team have reviewed this document and agree on its content and format. The comment areas below are to be used to state any minor points regarding the document that members may not agree with.

Abbie Minor \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_

Comments:

Constantin Koehler \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_

Comments:

Braxton Dubin \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_

Comments:

Naomi Schumacher \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_

Comments:

Sarah Kirby \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_

Comments:

# Appendix B – Document Contributions

This section identifies how each member contributed to the creation of this document. The percentages listed are an estimate of the percentage of work each person contributed.

Abbie Minor: 25%

Contribution: Modules Inspected, Defects

Constantin Koehler: 30%

Contribution: Coding and Commenting Conventions, Code Inspection Process, Defects

Braxton Dubin: 0

Contribution: Unclear

Naomi Schumacher: 5%

Contribution: Proofreading

Sarah Kirby: 40%

Contribution: Modules Inspected, Inspection Meetings, Defects, Conventions