

Coding Conventions for VocabVault

May 24, 2024

Max Mayer-Mader and Jillienne Ness

Introduction

- This document provides coding standards and best practices for the VocabVault web application developed in Python and Javascript. Adherence to these standards is essential to ensure code quality, maintainability, and consistency across the codebase.

General Principles

- **Readability and Clarity:** Code should be easily understandable. Prioritize clarity over cleverness.
- **Consistency:** The same coding conventions must be applied consistently across the codebase.
- **Simplicity and Performance:** Write simple and efficient code without premature optimization.
- **Commenting and Documentation:** Code should be well-commented and documented where necessary.

Formatting and Naming Conventions

Indentation and White Space

- Use tabs for indentation, not 4 spaces
- Use blank lines between each method to increase readability
- Use tabs when going to the next

Braces

- Always use braces for if, else, and else if statements and also for for, while, do-while and other loops even if there is only one line in the body
- Opening braces should be on the same line as its statement (K&R style)
 - Example:

```
if (x == y){
    func1();
    func2();
}
```

Naming

- **Classes and Interfaces:** Capitalize the first letter of each word (Pascal case)
 - Example: FoodPantry
- **Methods:** Capitalize the first letter of each word *except* the first word (Camel case)
 - Example: addItem
- **Variables:** Capitalize the first letter of each word *except* the first word (Camel case)

- Example: assignedNumber
- **Constants:** All uppercase letters with underscores between each word
 - Example: MAX_ITEMS

Line Length

- Lines should not exceed 100 characters. Break lines if necessary.

Code Structure

Classes

- Keep the number of methods within a class as low as possible. Follow the Single Responsibility Principle (SRP).
- Group related methods together for readability.

Methods

- Methods should be short and perform a single task.
- The number of parameters should be kept to a minimum. Consider using a parameter object if a method requires more than three parameters.

Comments and Documentation

- Use standard JavaDoc comments for all classes, interfaces, methods and any significant variables
- Inline comments should explain complex logic or reasoning that might not be obvious to the reader
- Avoid comments that provide any obvious or no additional information

Version Control

- Commit related changes in a logical unit and include a meaningful commit message.
- Use branches for features, bug fixes, and releases.

Error Handling

- When implementing something that could possibly raise an exception, always use a try-catch block and make a custom error message detailing what the issue was
 - Example: opening a file

```
try  
{
```

```
        File file = new File("example.txt");
    }
    catch (FileNotFoundException e)
    {
        // Handle the case where the file is not found
        System.out.println("File not found: " + e.getMessage());
        // Optionally, you can print the stack trace for debugging purposes
        // e.printStackTrace();}
```

Python example:

```
try:
    print(x)
except:
    print("An exception occurred")
```

Code Reviews

- All code should be reviewed by at least one other team member.
- Reviewers should check for adherence to these standards, logical errors, and possible simplifications.
- Review should occur prior to committing to the main branch.

Testing

- Write unit tests for individual classes to make sure every piece of code works in isolation.
- Verify tests with different various user sizes and data sizes to ensure that it can work at scale efficiently.
- Write functional tests for the complete program to test the final product.

Security

- Do not share the GitHub edit access with any user outside of the group
- Implement effective error handling that hides unexpected sensitive data that can be exposed in error messages.
- Keys and passwords should not be publicly visible

Conclusion

- This coding standards document establishes the foundation for developing the VocabVault web application. All contributors are expected to adhere to these standards to maintain the quality and integrity of the codebase. Consistent application of these practices will enhance collaboration and the overall success of the project.