HACETTEPE UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

BBM384 Software Engineering Laboratory

Online Book Management System

Coding Standards

BOOKIFY

Ismet Okatar 21727542

Atakan Ayyıldız 21526681

Göktuğ CANDEMİR 21627064

Ege ÇINAR 21627136

Muhammed Aydoğan 21726952

Bookify
Coding Standards

1 Introduction

This is a document about the coding standards in Bookify application. The specifications of coding standards are discussed such as name standards and coding conventions.The developers satisfy these standards to regulate the project's code.

2 Description

Coding Standard Document is needed for formalization of project conventions. Maintenance is a major part of the project. When people working on a project leave, people who come to their place need to review some of the pre-written code standards to understand the project system.

3 Coding Standards Specifications

3.1 Naming Convention Standards

We used camelcase naming convention in our project.CamelCase is a naming convention in which a name is formed of multiple words that are joined together as a single word with the first letter of each of the multiple words (except for the first word) capitalized so that each word that makes up the name can easily be read.

3.1.1 Naming Variables

The variable's name needs to clearly show what it represents.Sometimes we can make some abbreviations when variable names are too long, but we need to make sure that the variable name is understandable.We must follow these rules when creating variables.

Here is the example variable name which is in accordance with CamelCase standard "String userName"

3.1.2 Naming Constants

Constants mean the variable which does not change in any steps of code. If the constant is ANSI constants should be all uppercase with words separated by underscores ("_") like int MIN_WIDTH = 4;

3.1.3 Naming Classes:

Class names should be nouns, in mixed case with the first letter of each internal word capitalized.

public Class Category

3.1.4 Naming Interfaces:

Interface names should be capitalized like class names.

public Interface BookRepository

3.1.5 Naming Collection

The Java collections framework is a group of classes and interfaces that implement common reusable aggregation data structures.Collection is defining arrays and array list in our code.

For example: Array<Book> booklist= new ArrayList<>()

3.2 Comment Standards

Beginning Comments: All source files should begin with a c-style comment that lists the programmers.

```
/*
* ProjectName.java
*
* Created on: dd/mm/yyyy
* Author: *
*/
```

Also line comments should be used when complex code is written.

```
complexCode();        // This method does something.
```

3.3 File Organization:

We use conventional Spring file organization. Source files are stored in the src folder. The test files are in the test folder. Since we use the MVC architecture model, views and controller files are stored in their respective folder. In addition exception and repository folders are present to hold respective file categories. For example:

We put a simple file that summarizes the contents of the directory. This was a README file.

In our project, we avoid creating files longer than 500 lines because those files are cumbersome and not efficient.

3.4 White Space

We have blank lines in our code to improve readability. But I don't mean opening and closing parentheses. We mostly used blank lines after a line which is finished with a comma. By doing this, each part of the code can be more legible. In addition we have white spaces to create better understandable software. And this white space is mostly used between operation conditions, initializing values, after opening and before closing parenthesis if there is a variable between them. For example;

```
while( x == y )
{
a += a + b;
y = (2 * y ) + ( 2 * x );
}
```

3.5 Coding Convention:

80% of the lifetime cost of a piece of software goes to maintenance. Our coding conventions contain file names such as package and class name, declarations for each convention, statement for each line of code, naming conventions that have standards. Our software contains many variable, constant, collection and component in its code. So we have standards for each of them to create a readable and understandable software.