



Peer-graded Assignment: Capstone Assignment 2.3 – Identify and Fix Code Smells

Submit by January 13, 11:59 PM PST

Important Information

It is especially important to submit this assignment before the deadline, January 13, 11:59 PM PST, because it must be graded by others. If you submit late, there may not be enough classmates around to review your work. This makes it difficult - and in some cases, impossible - to produce a grade. Submit on time to avoid these risks.

Instructions

My submission

Now that we've built an application that implements the Command Pattern and MVC Pattern, it's time to evaluate the code.

Discussions

In the first part of this assignment you will be challenged to identify Code Smells in the given code base and brainstorm ways to fix them.

In the second part of this assignment you are going to fix the Long Method code smell.

Review criteria

[less ^](#)

You will be graded on the completeness, clarity and correctness of your submission.

Step-By-Step Assignment Instructions

[less ^](#)

Setup instructions

Before you begin, you will need to download and extract the solution to the previous Peer Review assignment:

SharingApp_mvc_pattern

Start Android Studio and open SharingApp as an existing project.

To give you a better idea of the scope of this version of the app please review the following user stories:

User Stories with Contacts.pdf

Guidelines for the assignment

Review these lectures to aid you on your assignment:

2.3.7 - Anti-patterns & Code Smells

**How to create your assignment****Part 1**

Use the given code base to identify two code smells that are described in the Anti Patterns & Code Smells lecture -- **excluding Long Method**.

For each of the two code smells you have identified:

- Document them by clearly stating where the code smell is (which class, which method, etc.)
- Which code smell from the lecture it is, and why you think the code fits the description of the code smell.
- Give your reasoning on why each code smell is a problem and should be fixed.
- Suggest a solution to fix the code smell.

Part 1 Notes:

- If the same piece of code has more than one code smell in it, all code smells will count, as long as each has their own code smell type and solution. However, identifying the same code smell twice in different parts of the code will not count.
- We do not want you to actually fix these two code smells!
- You should only have to write one paragraph per code smell and submit no more than 600 words for your answer.
- If you would like you may use UML diagrams alongside your explanation to illustrate how something is a smell or how it should be fixed.

Part 2

In this second part of the assignment, the code smell you will be fixing is Long Method.

To fix this code smell, you will Implement a validateInput() method in the following activities:

- AddItemActivity
- EditItemActivity
- AddContactActivity
- EditContactActivity

These four activities all contain "save" methods that consist of several lines of code for validating user input. The lines of code responsible for validating the input should be moved to the validateInput() method. This new method can then be called from inside the save method.

Part 2 Hints:

1. Replace multiple lines of input validation in the “save” method with the following lines:



```
1  If ( !validateInput() ) {  
2      return;  
3  }
```

2. Create and implement the validateInput() method with the following signature:

```
1  public boolean validateInput() {  
2      // Input validation goes here...  
3      // ...  
4  }
```

3. You will have to modify the scope of some of variables in each activity to implement this. For example, in EditItemActivity you will have to modify the scope of title_str, maker_str, description_str, length_str, width_str, height_str, and contact.

Lastly, don't forget -- the overall functionality of the application should not change!

Submission Instructions

Part 1: Upload a PDF of your solution where prompted.

Part 2: Include the following files in a folder:

- AddItemActivity.java
- EditItemActivity.java
- AddContactActivity.java, and
- EditContactActivity.java

Compress the folder into a ZIP folder. Windows users can use 7zip or WinRAR. Upload it where prompted.

