

---

# Project Review Room4you

---

BY TEAM 8

*Reviewers:*  
Michael GRÜNIG  
Sara PEETERS  
Daniel ZILTENER

November 16, 2014

# Contents

<b>1</b>	<b>Design</b>	<b>2</b>
1.1	Violation of MVC pattern . . . . .	2
1.2	Usage of helper objects between view and model . . . . .	2
1.3	Rich OO domain model . . . . .	2
1.4	Clear responsibilities . . . . .	2
1.5	Sound invariants . . . . .	2
1.6	Overall code organization and reuse . . . . .	2
<b>2</b>	<b>Coding style</b>	<b>2</b>
2.1	Consistency . . . . .	2
2.2	Intention-revealing names . . . . .	2
2.3	Do not repeat yourself . . . . .	2
2.4	Exception, testing null values . . . . .	2
2.5	Encapsulation . . . . .	2
2.6	Assertion, contracts, invariant checks . . . . .	2
2.7	Utility methods . . . . .	2
<b>3</b>	<b>Documentation</b>	<b>2</b>
3.1	Understandable . . . . .	2
3.2	Intention-revealing . . . . .	2
3.3	Describe responsibilities . . . . .	2
3.4	Match a consistent domain vocabulary . . . . .	2
<b>4</b>	<b>Tests</b>	<b>2</b>
4.1	Clear and distinct test cases . . . . .	2
4.2	Number/coverage of test cases . . . . .	3
4.3	Easy to understand the case that is tested . . . . .	3
4.4	well crafted set of test data . . . . .	3
4.5	readability . . . . .	3
<b>5</b>	<b>controller class evaluation</b>	<b>3</b>

## **1 Design**

- 1.1 Violation of MVC pattern**
- 1.2 Usage of helper objects between view and model**
- 1.3 Rich OO domain model**
- 1.4 Clear responsibilities**
- 1.5 Sound invariants**
- 1.6 Overall code organization and reuse**

## **2 Coding style**

- 2.1 Consistency**
- 2.2 Intention-revealing names**
- 2.3 Do not repeat yourself**
- 2.4 Exception, testing null values**
- 2.5 Encapsulation**
- 2.6 Assertion, contracts, invariant checks**
- 2.7 Utility methods**

## **3 Documentation**

- 3.1 Understandable**
- 3.2 Intention-revealing**
- 3.3 Describe responsibilities**
- 3.4 Match a consistent domain vocabulary**

## **4 Tests**

### **4.1 Clear and distinct test cases**

There are 8 test classes, distributed over two groups(each in their own package), controller tests, and service tests. There are also two testsuites, each with a different number of tests. Although some tests appear in both testsuites, none of the testsuites contains all 8 tests. Moreover, we note that some of the tests in the service test package seem tot be testing data access objects directly and not services. Although the idea behind the organisation of the tests is good, the implementation lacks consistency and is confusing. The fact that no javadoc is provided for most of the tests doesn't help either.

- 4.2 Number/coverage of test cases
- 4.3 Easy to understand the case that is tested
- 4.4 well crafted set of test data
- 4.5 readability
- 5 controller class evaluation