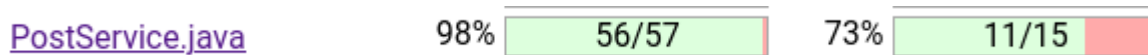


## Assignment 3 - Mutation Testing

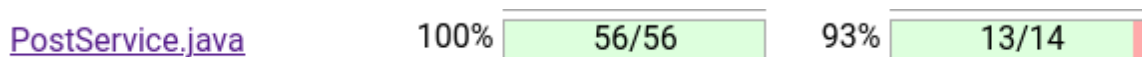
### Group 01A

#### **SSP: PostService.java (73%)**

**Figure of test suite mutation score before improvements:**



**Figure of test suite mutation score after improvements:**



#### **Explanation of Improvements:**

The decreased mutation score was caused by a few very similar mutants not being killed. Namely, there were mutants that deleted an important line of the code, which the tests failed to detect. As such, additional assertions were added to pre-existing tests in order to make sure the tests killed those mutants.

Another such mutant was linked to a `println()` statement forgotten behind during debugging. As it adds no useful functionality to the service, it was decided to remove the `println()` from the code.

There was an additional mutant that was not killed however that was due to the fact that there were no tests for the specific method, which was added during refactoring. In order to fix this a quick test was written.

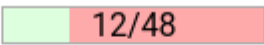
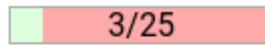
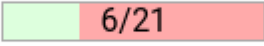
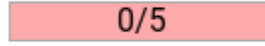
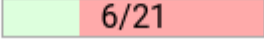
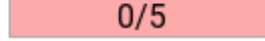
Finally, there was a mutant that was not able to be killed due to the fact that it was not possible to mock the certain method as it was a method call of an entity and unfortunately it was not possible to check the new value of the field after the fact.

#### **Links to commits:**

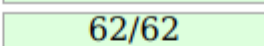
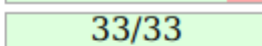
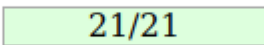
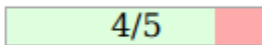
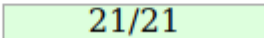
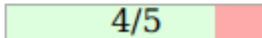
1. b905351d19f343dcc602abf7115459e935049a4e  
<https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/commit/b905351d19f343dcc602abf7115459e935049a4e>

### **SSP: Contract.java (12%)**

**Figure of test suite mutation score before improvements:**

<u>Contract.java</u>	25%		12%	
<u>ContractParties.java</u>	29%		0%	
<u>ContractTimes.java</u>	29%		0%	

**Figure of test suite mutation score after improvements:**

<u>Contract.java</u>	100%		100%	
<u>ContractParties.java</u>	100%		80%	
<u>ContractTimes.java</u>	100%		80%	

### ***Explanation of Improvements:***

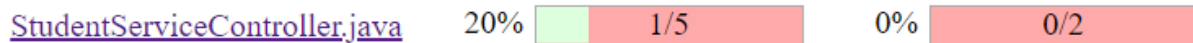
To increase mutation coverage of this class I started by increasing the normal coverage, this was done by testing all the constructors, getters, setters and some logic that constructed a Contract from an Offer. From there to increase mutation coverage I spied different classes and used dependency injection to test them, in particular the wrapper classes “ContractTimes” and “ContractParties”. By using spies I was able to check if everything was called correctly, as lots of mutants are based on removing calls to sub-methods. Boundary tests with assertions also helped in killing mutants in which return values were constant and conditions were changed.

### ***Links to commits:***

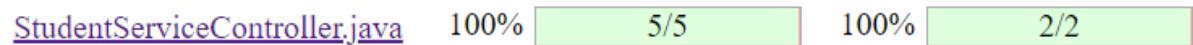
1. Bb17edc8e8841c1d2ae4519fe988974caeef90b  
<https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/commits/bb17edc8e8841c1d2ae4519fe988974caeef90b>

## **GSP: StudentServiceController.java (0%)**

**Figure of test suite mutation score before improvements:**



**Figure of test suite mutation score after improvements:**



### **Explanation of Improvements:**

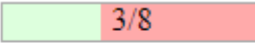
To increase the mutation score, as there were no tests for this class to begin with, I started with creating tests for the methods inside. As the methods I was targeting specifically were simple methods calling other methods to get their logic, it was easy to set up a mock with the expected result. By doing this, it eliminated the need for more tests, as the tests would fail if the input was anything else other than the specific objects I had created. So all the mutations were killed, and mutation coverage increased by 100%.

### **Links to commits:**

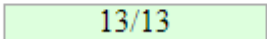
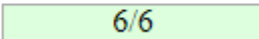
1. 2fc8ea8d21792b476cbe8b08db5fefb93830fe9c  
<https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/commits/2fc8ea8d21792b476cbe8b08db5fefb93830fe9c>
2. 41e59995bcbfbc2e2eb1c6b6cdda843ef3a7c176  
<https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/commits/41e59995bcbfbc2e2eb1c6b6cdda843ef3a7c176>
3. 05756e49c0959984692b9c239d4ddfe9e5de1d94  
<https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/commits/05756e49c0959984692b9c239d4ddfe9e5de1d94>
4. 28464703b946b70e4c46c56db81b7cf70439c071  
<https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/commits/28464703b946b70e4c46c56db81b7cf70439c071>

## **GSP: EditGenericPostService.java (38%)**

**Figure of test suite mutation score before improvements:**

Name	Line Coverage	Mutation Coverage
<a href="#">EditGenericPostService.java</a>	57% 	38% 

**Figure of test suite mutation score after improvements:**

Name	Line Coverage	Mutation Coverage
<a href="#">EditGenericPostService.java</a>	100% 	100% 

### ***Explanation of Improvements:***

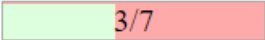
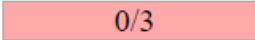
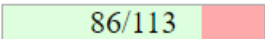
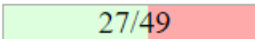
To increase the mutation score I started by increasing the Line Coverage of the Class to 100% by adding 2 new Test methods for exception cases. After adding the new Tests for Line Coverage, I modified them by mocking the GenericPostRepository, because of the method calls (findById(), save(), getGenericPostById()), which were used in the EditGenericPost method. By doing this all the mutations in the class were killed, which led to 100% line and mutation coverage.

### ***Links to commits:***

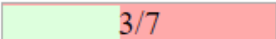
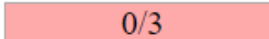
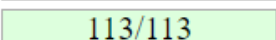
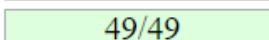
- 525f4d5d87df13f3baef4cbd957d77ad2fb955a7  
[https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge\\_requests/84/diffs?commit\\_id=525f4d5d87df13f3baef4cbd957d77ad2fb955a7](https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge_requests/84/diffs?commit_id=525f4d5d87df13f3baef4cbd957d77ad2fb955a7)
- bf22c516a1c3900b10ff54772aa4d919a9f234c7  
[https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge\\_requests/84/diffs?commit\\_id=bf22c516a1c3900b10ff54772aa4d919a9f234c7](https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge_requests/84/diffs?commit_id=bf22c516a1c3900b10ff54772aa4d919a9f234c7)
- ddae5ee56338f99f092926f5a81fa12a609b4c70  
[https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge\\_requests/84/diffs?commit\\_id=ddae5ee56338f99f092926f5a81fa12a609b4c70](https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge_requests/84/diffs?commit_id=ddae5ee56338f99f092926f5a81fa12a609b4c70)
- e624d190d5ef3b3058e10eb38d540c9e0b8616bf  
[https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge\\_requests/84/diffs?commit\\_id=e624d190d5ef3b3058e10eb38d540c9e0b8616bf](https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge_requests/84/diffs?commit_id=e624d190d5ef3b3058e10eb38d540c9e0b8616bf)

## **Users: UserService.java (55%)**

**Figure of test suite mutation score before improvements:**

Name	Line Coverage	Mutation Coverage
<a href="#">FeedbackService.java</a>	43%  3/7	0%  0/3
<a href="#">UserService.java</a>	76%  86/113	55%  27/49

**Figure of test suite mutation score after improvements:**

Name	Line Coverage	Mutation Coverage
<a href="#">FeedbackService.java</a>	43%  3/7	0%  0/3
<a href="#">UserService.java</a>	100%  113/113	100%  49/49

### ***Explanation of Improvements:***

To increase the mutation score I started by increasing the line coverage. Once that was done I augmented the tests with assertions as specific as possible. In this way, mutants that ignored calls to other methods or mutants that modified conditional statements were killed, because the final result of the execution was highly dependent on the chain of computations. The most difficult part was to kill mutants that removed statements. It turns out that I had to create separate instances of objects and make more assertions. The assertions needed to check all fields of the returned object. Furthermore, to deal with void methods I made use of Mockito.verify.

### ***Links to commits:***

1. [https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge\\_requests/80/diffs?commit\\_id=d3c122da9b629f82015b90008acc8f9041e81a59](https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge_requests/80/diffs?commit_id=d3c122da9b629f82015b90008acc8f9041e81a59)
2. [https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge\\_requests/80/diffs?commit\\_id=e846f4fa18839440406922b04cfbf9d3c2fe003d](https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/merge_requests/80/diffs?commit_id=e846f4fa18839440406922b04cfbf9d3c2fe003d)


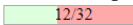
## **Contracts: PdfGeneratorService.java (19%)**

**Figure of test suite mutation score before improvements:**

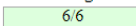
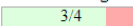
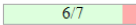
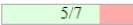
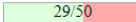
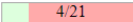
### **Pit Test Coverage Report**

#### **Package Summary**

nl.tudelft.sem.contracts.services

Number of Classes	Line Coverage	Mutation Coverage
3	65%  41/63	38%  12/32

#### **Breakdown by Class**

Name	Line Coverage	Mutation Coverage
<a href="#">ContractService.java</a>	100%  6/6	75%  3/4
<a href="#">DetailsCheckService.java</a>	86%  6/7	71%  5/7
<a href="#">PdfGeneratorService.java</a>	58%  29/50	19%  4/21


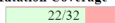
Report generated by PIT 1.5.1

**Figure of test suite mutation score after improvements:**

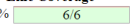
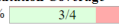
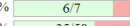
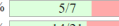
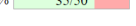
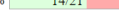
### **Pit Test Coverage Report**

#### **Package Summary**

nl.tudelft.sem.contracts.services

Number of Classes	Line Coverage	Mutation Coverage
3	75%  47/63	69%  22/32

#### **Breakdown by Class**

Name	Line Coverage	Mutation Coverage
<a href="#">ContractService.java</a>	100%  6/6	75%  3/4
<a href="#">DetailsCheckService.java</a>	86%  6/7	71%  5/7
<a href="#">PdfGeneratorService.java</a>	70%  35/50	67%  14/21

Report generated by PIT 1.5.1

## **Explanation of Improvements:**

In order to increase the Mutation Coverage I added an additional test for a method addToDocument which was previously untested. Additionally I modified the existing tests for formatTitle, formatDetails, formatDetailsHeader and formatIntro methods to catch mutants caused by tests not checking certain functionalities done by these methods like properties of the fonts. The combination of modifying and adding new tests increased the mutation coverage and improved the quality of the test suite.

## **Links to commits:**

1. <https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/commit/26d6cb83f2dd517971d698127e0ea7ac1faf3d2a>
2. <https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/commit/392959c8c7810e64a8dbb483fad3e704d9eade3e>
3. <https://gitlab.ewi.tudelft.nl/cse2115/2021-2022/sem-group-01a/sem-repo-01a/-/commit/392959c8c7810e64a8dbb483fad3e704d9eade3e>