Software Engineering Methods: Lab Assignment 3

Group 18A

January 2022

Chapter 1

Introduction

Finally, we have reached the last step of our project's implementation: mutation testing. You may wonder why we chose to go even further with our testing process, since we already had a good amount of branch coverage. The main reason is that code coverage does not tell us how many code elements are checked by their assertions, so we had to dive deeper into verifying the behaviour of our system. To simplify our work, we used a mutation testing tool: **Pitest**, to identify four classes which have a mutation score below 70 percent.

Next, we based our mutation on two major hypotheses: The Competent Programmer Hypothesis and the Coupling Effect, which both revolve around the idea of discovering complex types of faults by introducing simple faults into our test cases. We will showcase these changes in the next chapter, along with the improvements that they bring.

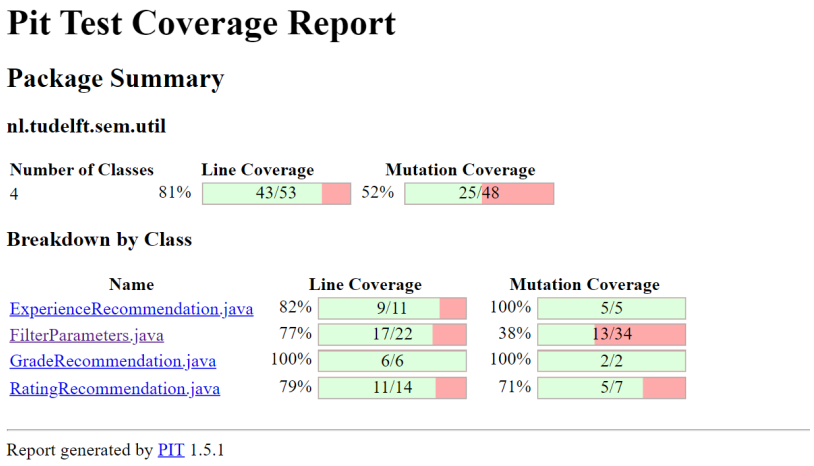
Chapter 2

Mutation Testing

2.1 `FilterParameters` Class

Prior Mutation Score

We choose to improve the mutation score of the `FilterParameter` class. Prior to adding new test cases, our mutation score is 38%.



Pit Test Coverage Report

Package Summary

nl.tudelft.sem.util

Number of Classes	Line Coverage		Mutation Coverage	
4	81%	43/53	52%	25/48

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
ExperienceRecommendation.java	82%	9/11	100%	5/5
FilterParameters.java	77%	17/22	38%	13/34
GradeRecommendation.java	100%	6/6	100%	2/2
RatingRecommendation.java	79%	11/14	71%	5/7

Report generated by PIT 1.5.1

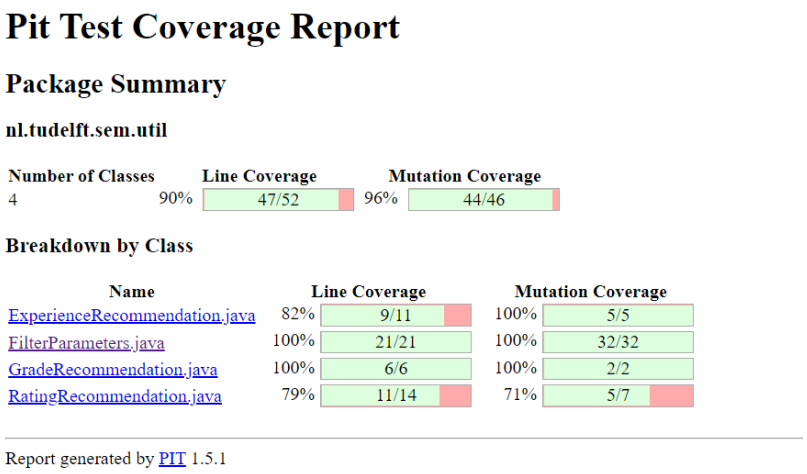
Figure 2.1: **Prior:** PIT Mutation Score Report For `FilterParameter` Class

Operations

To increase the mutation score, we added additional tests for the Equals and HashCode methods for the `FilterParameters` object. The mutations mostly arose from the `equals` and `hashCode` method. See GitLab commit below for full changes:
FilterParameters PIT

Post Mutation Score

With the additional test cases, we reached a mutation score of 100% for the `FilterParameter` class.



Pit Test Coverage Report

Package Summary

nl.tudelft.sem.util

Number of Classes	Line Coverage		Mutation Coverage	
4	90%	47/52	96%	44/46

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
ExperienceRecommendation.java	82%	9/11	100%	5/5
FilterParameters.java	100%	21/21	100%	32/32
GradeRecommendation.java	100%	6/6	100%	2/2
RatingRecommendation.java	79%	11/14	71%	5/7

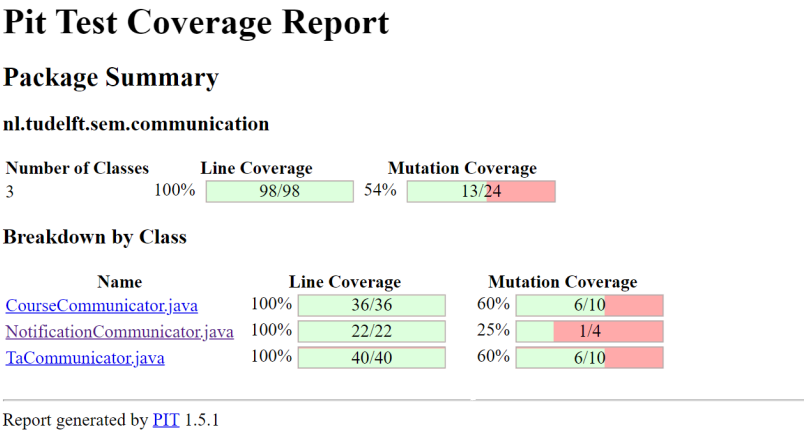
Report generated by PIT 1.5.1

Figure 2.2: **Post:** PIT Mutation Score Report For `FilterParameter` Class

2.2 `NotificationCommunicator` Class

Prior Mutation Score

We also chose to improve the mutation score of the `NotificationCommunicator` class. Prior to adding new test cases, our mutation score is 25%.



Pit Test Coverage Report

Package Summary

nl.tudelft.sem.communication

Number of Classes	Line Coverage		Mutation Coverage	
3	100%	98/98	54%	13/24

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
CourseCommunicator.java	100%	36/36	60%	6/10
NotificationCommunicator.java	100%	22/22	25%	1/4
TaCommunicator.java	100%	40/40	60%	6/10

Report generated by PIT 1.5.1

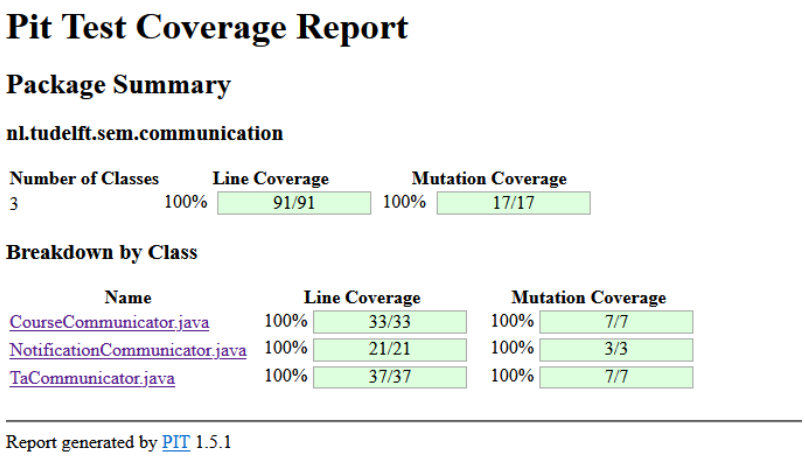
Figure 2.3: PIT Mutation Score Report For `NotificationCommunicator` Class

Operations

In order to improve the mutation score, we use stricter assertions. That is, we ensure that all attributes of the returned object is as expected. Moreover, we also test for more alternative paths in the Control Flow Graph of the methods. The link to the commit is: Notification Communicator PIT

Post Mutation Score

With the additional test cases, we reached a mutation score of 100% for the `Notification Communicator` class.



Pit Test Coverage Report

Package Summary

nl.tudelft.sem.communication

Number of Classes	Line Coverage		Mutation Coverage	
3	100%	91/91	100%	17/17

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
CourseCommunicator.java	100%	33/33	100%	7/7
NotificationCommunicator.java	100%	21/21	100%	3/3
TaCommunicator.java	100%	37/37	100%	7/7

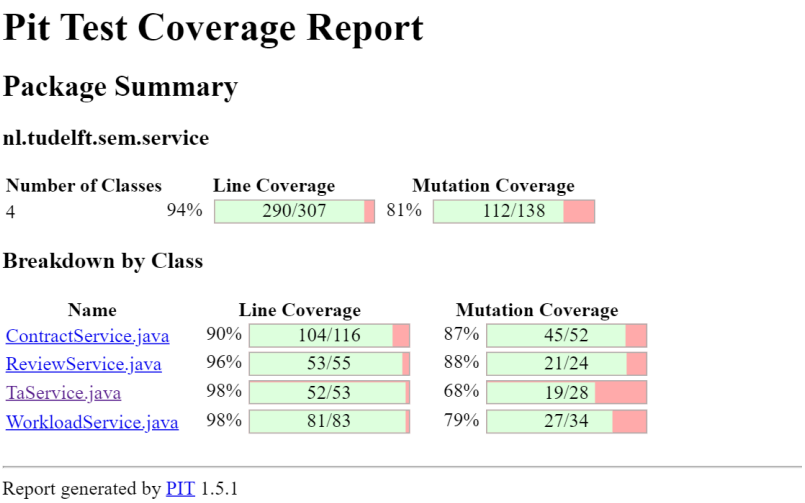
Report generated by PIT 1.5.1

Figure 2.4: PIT Mutation Score Report For `Notification Communicator` Class

2.3 `TaService` Class

Prior Mutation Score

We also chose to improve the mutation score of the `TaService` class. Prior to adding new test cases, our mutation score is 68%.



Pit Test Coverage Report

Package Summary

nl.tudelft.sem.service

Number of Classes	Line Coverage		Mutation Coverage	
4	94%	290/307	81%	112/138

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
ContractService.java	90%	104/116	87%	45/52
ReviewService.java	96%	53/55	88%	21/24
TaService.java	98%	52/53	68%	19/28
WorkloadService.java	98%	81/83	79%	27/34

Report generated by PIT 1.5.1

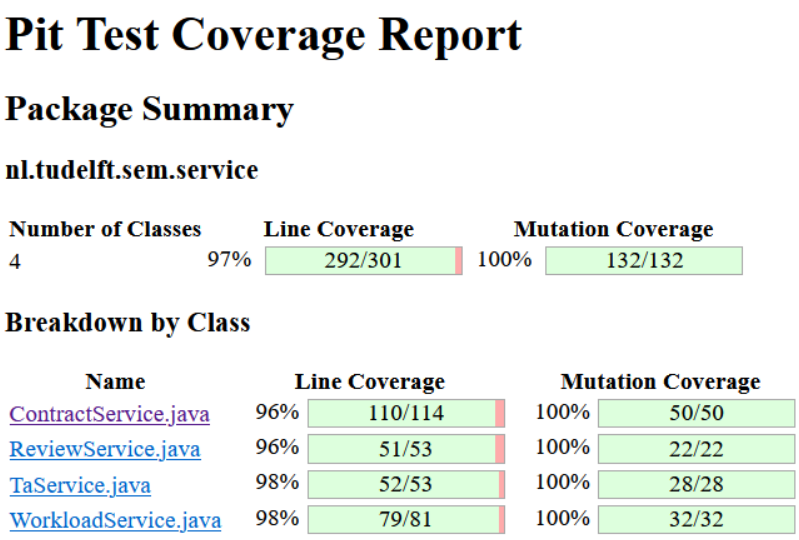
Figure 2.5: PIT Mutation Score Report For `TaService` Class

Operations

The main change was the Update CRUD operations. Previously, we only checked one of multiple things which could be updated. We now check for all attributes that could be changed. Link to the commit is the following: TaService PIT

Post Mutation Score

With the additional test cases, we reached a mutation score of 100% for the `TaService` class.



Pit Test Coverage Report

Package Summary

nl.tudelft.sem.service

Number of Classes	Line Coverage		Mutation Coverage	
4	97%	292/301	100%	132/132

Breakdown by Class

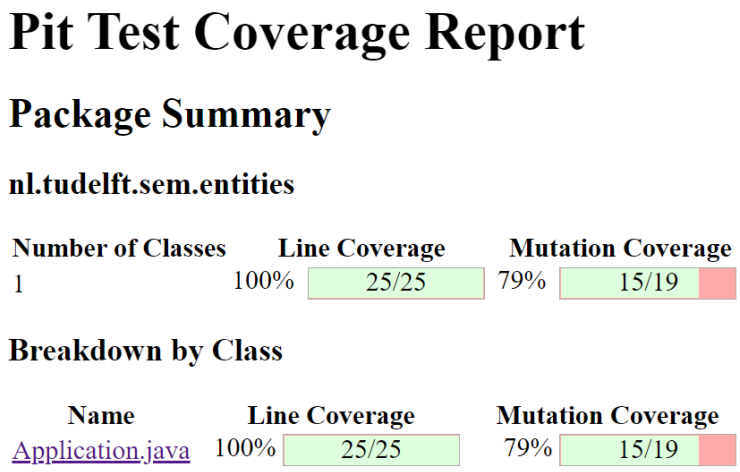
Name	Line Coverage		Mutation Coverage	
ContractService.java	96%	110/114	100%	50/50
ReviewService.java	96%	51/53	100%	22/22
TaService.java	98%	52/53	100%	28/28
WorkloadService.java	98%	79/81	100%	32/32

Figure 2.6: PIT Mutation Score Report For `TaService` Class

2.4 `Application Entity` Class

Prior Mutation Score

We also chose to improve the mutation score of the `Application Entity` class. Prior to adding new test cases, our mutation score is 79%.



Pit Test Coverage Report

Package Summary

nl.tudelft.sem.entities

Number of Classes	Line Coverage		Mutation Coverage	
1	100%	25/25	79%	15/19

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
Application.java	100%	25/25	79%	15/19

Figure 2.7: PIT Mutation Score Report For `Application Entity` Class

Operations

We wrote several unit tests to improve the mutation score. The link to the commit is the following: Application Entity PIT

Post Mutation Score

With the additional test cases, we reached a mutation score of 100% for the `Application Entity` class.

Pit Test Coverage Report

Package Summary

nl.tudelft.sem.entities

Number of Classes	Line Coverage		Mutation Coverage	
1	100%	25/25	100%	19/19

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
Application.java	100%	25/25	100%	19/19

Figure 2.8: PIT Mutation Score Report For `Application Entity` Class