Mutation Testing

CSE2115 Software Engineering Methods Group 08b

Y2Q2 2021/2022

1 Introduction

In this assignment a number of classes have been covered in terms of mutation testing coverage. The reports were generated using the Pitest plugin for IntelliJ. The four worst classes were picked for improvement, which are discussed below. To observe the changes, please consult the last commit before these changes were implemented, and the commit after these changes were implemented.

2 Mutation Score Before

2.1 RoomValidator Class

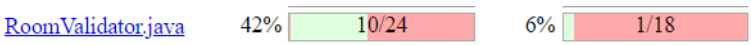

Figure 2.1: RoomValidator class in main gateway. (left: Line Coverage, right: Mutation Coverage)

2.2 User Class

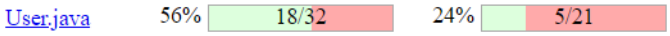

Figure 2.2: User class in main gateway. (left: Line Coverage, right: Mutation Coverage)

2.3 Role Class

Figure 2.3: Role class in main gateway. (left: Line Coverage, right: Mutation Coverage)

2.4 UserService Class

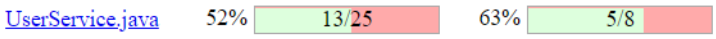

Figure 2.4: UserService class in user microservice. (left: Line Coverage, right: Mutation Coverage)

3 Mutation Score After

3.1 RoomValidator class

The RoomValidator class had the lowest mutation score of the tested methods. This was most likely caused by the chosen method for testing the validators as a whole, which was to test the methods that contact the validators thoroughly, with every possible option in mind.

This report indicated that this was not enough, and changes were implemented with newly added tests (commit bbaa2710). These changes managed to raise the mutation score of this class from 6% to 100%. The line coverage was also raised from 42% to 100%.

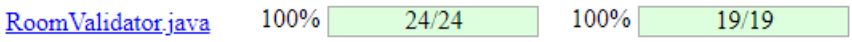

Figure 3.1: RoomValidator class in main gateway after improving testing. (left: Line Coverage, right: Mutation Coverage)

3.2 User class

Prior to improving the test class for the User object, the report showed a mutation score of only 24%. This was improved mainly by adding test cases for the methods and branches that were not previously covered. To achieve 100% mutation coverage, some modifications were necessary within the User class, such as slightly changing the toString method and creating an additional method for adding roles (commit ecaaddb3). Upon improving the test suite, the line coverage was also raised from 56% to 100%.

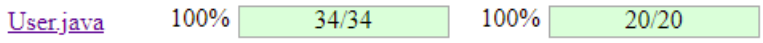

Figure 3.2: User class after improving testing. (left: Line Coverage, right: Mutation Coverage)

3.3 Role class

The role class prior to the improved testing had shown a mutation score of only 17% and line coverage of 47%. This was easily improved by adding tests to some methods which were untested for the role object. To achieve 100% mutation testing some tests had to be modified to make sure there was no way any mutations could survive (commit 34974562).

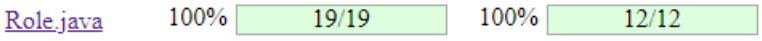

Figure 3.3: Role class after improving testing. (left: Line Coverage, right: Mutation Coverage)

3.4 UserService class

The report showed only 52% line coverage and 63% mutation coverage initially. To increase coverage tests were added for previously untested methods and existing tests were modified in order to achieve 100% for both line and mutation coverage(commit 65898b00).

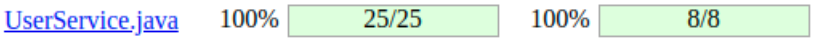

Figure 3.4: UserService class after improving testing. (left: Line Coverage, right: Mutation Coverage)