CS731 SOFTWARE TESTING

PROJECT REPORT

MUTATION TESTING ON A WEB APP PERFORMING CRUD OPERATIONS ON A TABLE OF PRODUCTS

SUBMITTED BY:

MT2022044 GAURANGI AGRAWAL

MT2022165 ANOUSHKA CHOUKSEY

PROJECT AIM

The aim of the project is to use **Mutation Testing** to test a real-world software project with the help of open-source tools. We aim to apply mutation operators at the level of a statement within a method or a function and at the integration level. The mutated program needs to be strongly killed by the designed test cases. The target is to use at least three different mutation operators at the unit level and at the integration level.

CODE

The source code for the project is in Java language (in the mentioned GitHub repository) can be found here.

STRATEGY & TOOLS USED

Mutation testing involves making small modifications to the source code, referred to as mutants, with the goal of demonstrating that these changes can impact the program's execution and outcomes. There are two ways to eliminate a mutant:

1. **Weakly killing a mutant**: This occurs when the memory state of the program after executing the mutated statement differs from the memory state when the statement was unaltered. Importantly, the program's output on a test case may remain the same, regardless of whether the statement was mutated.
2. **Strongly killing a mutant**: In this case, the program's output on a test case must change when a statement is mutated compared to when it is not. Strongly killing a mutant indicates that the error introduced by the mutation propagates through the program, resulting in distinct outputs with and without the mutation. Our chosen approach in mutation testing is to aim for strong mutant elimination.

The tools that we used for mutation testing are as follows:

1. IntelliJ for Java: IntelliJ IDEA is an integrated development environment (IDE) written in Java for developing computer software written in Java, Kotlin, Groovy, and other JVM-based languages.
2. PIT Mutation Testing Tool: An easy-to-use mutation testing tool, that works for Java. We used the PITclipse plugin for Eclipse to integrate this tool into the Eclipse IDE.
3. JaCoCo: JaCoCo is a free code coverage library for Java, which has been created by the EclEmma team based on the lessons learned from using and integration existing libraries for many years.

MUTATIONS USED

PIT, by default provides a set of mutation operators. These operators are listed below:

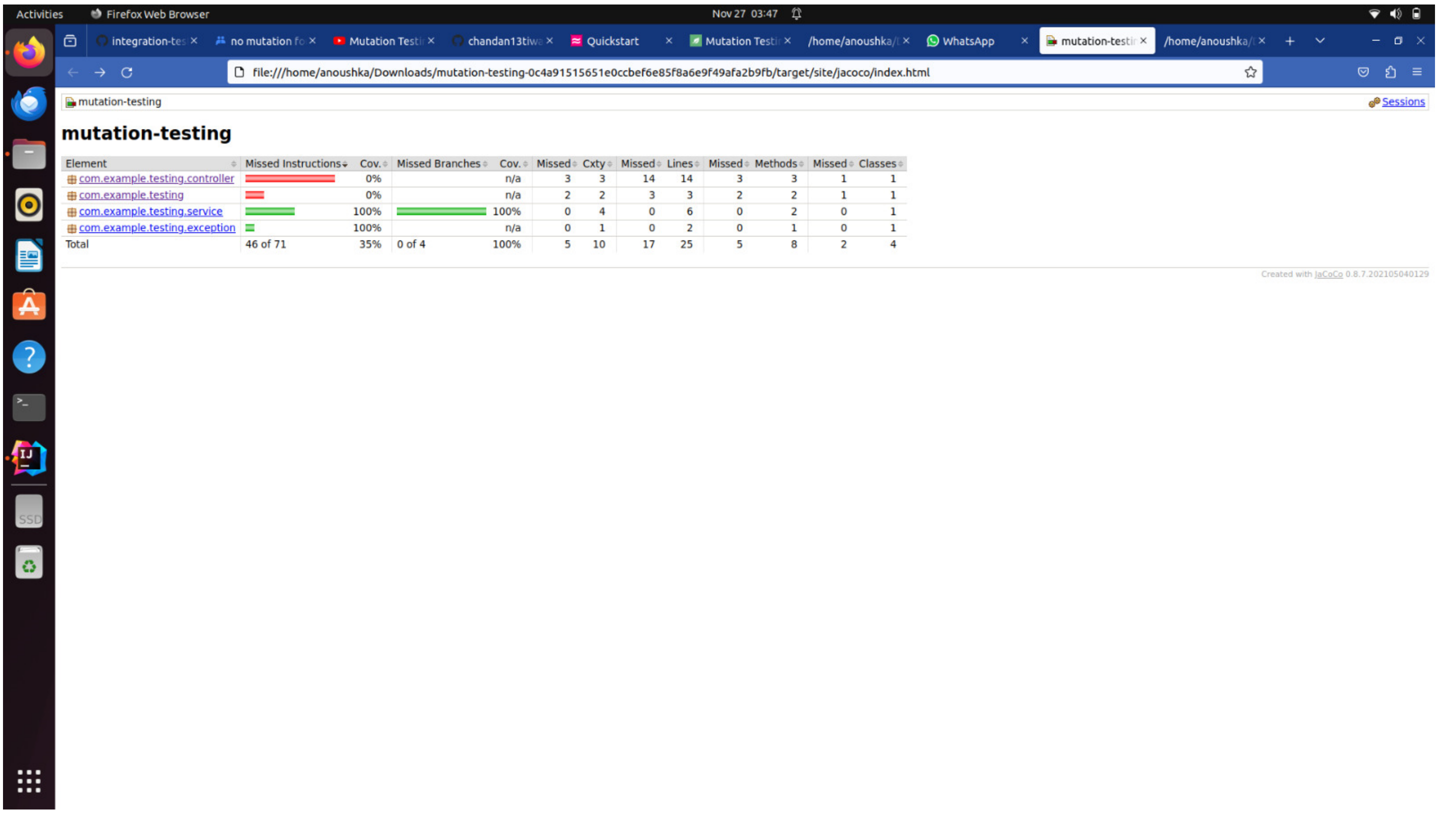
BOOLEAN_FALSE_RETURN	INCREMENTS_MUTATOR
BOOLEAN_TRUE_RETURN	INVERT_NEGS_MUTATOR
CONDITIONALS_BOUNDARY_MUTATOR	MATH_MUTATOR
EMPTY_RETURN_VALUES	NEGATE_CONDITIONALS_MUTATOR
PRIMITIVE_RETURN_VALS_MUTATOR	VOID_METHOD_CALL_MUTATOR
NULL_RETURN_VALUES	

TESTING RESULTS

UNIT TESTING

1. Unit Testing before killing mutants

a. Code coverage using Jacoco on the Project

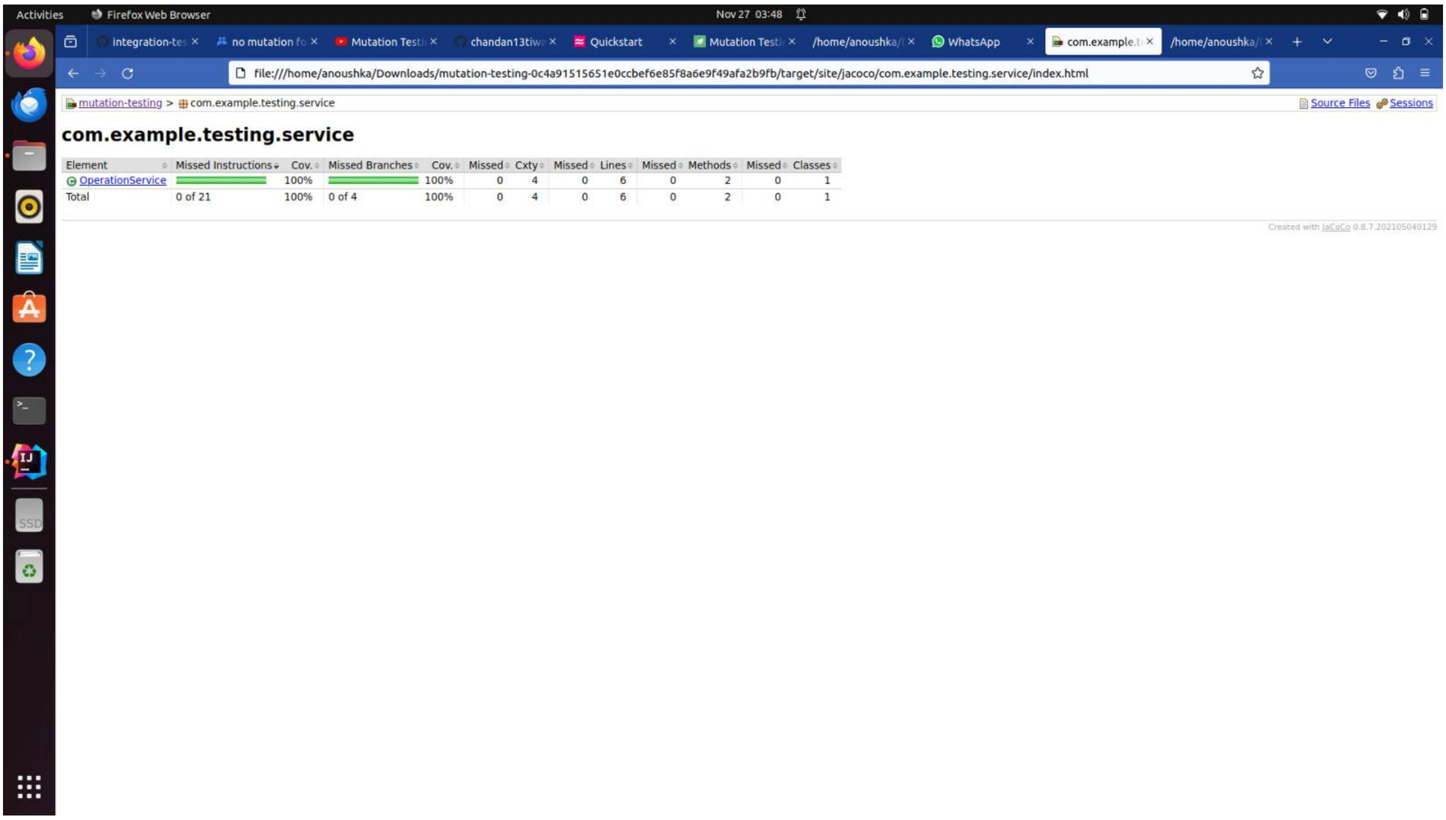


file:///home/anoushka/Downloads/mutation-testing-0c4a91515651e0ccbef6e85f8a6e9f49afa2b9fb/target/site/jacoco/index.html

mutation-testing

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.example.testing.controller	🟥	0%		n/a	3	3	14	14	3	3	1	1
com.example.testing	🟥	0%		n/a	2	2	3	3	2	2	1	1
com.example.testing.service	🟩	100%	🟩	100%	0	4	0	6	0	2	0	1
com.example.testing.exception	🟩	100%		n/a	0	1	0	2	0	1	0	1
Total	46 of 71	35%	0 of 4	100%	5	10	17	25	5	8	2	4

b. Code coverage using Jacoco for package *com.example.testing.service*



file:///home/anoushka/Downloads/mutation-testing-0c4a91515651e0ccbef6e85f8a6e9f49afa2b9fb/target/site/jacoco/com.example.testing.service/index.html

mutation-testing > com.example.testing.service

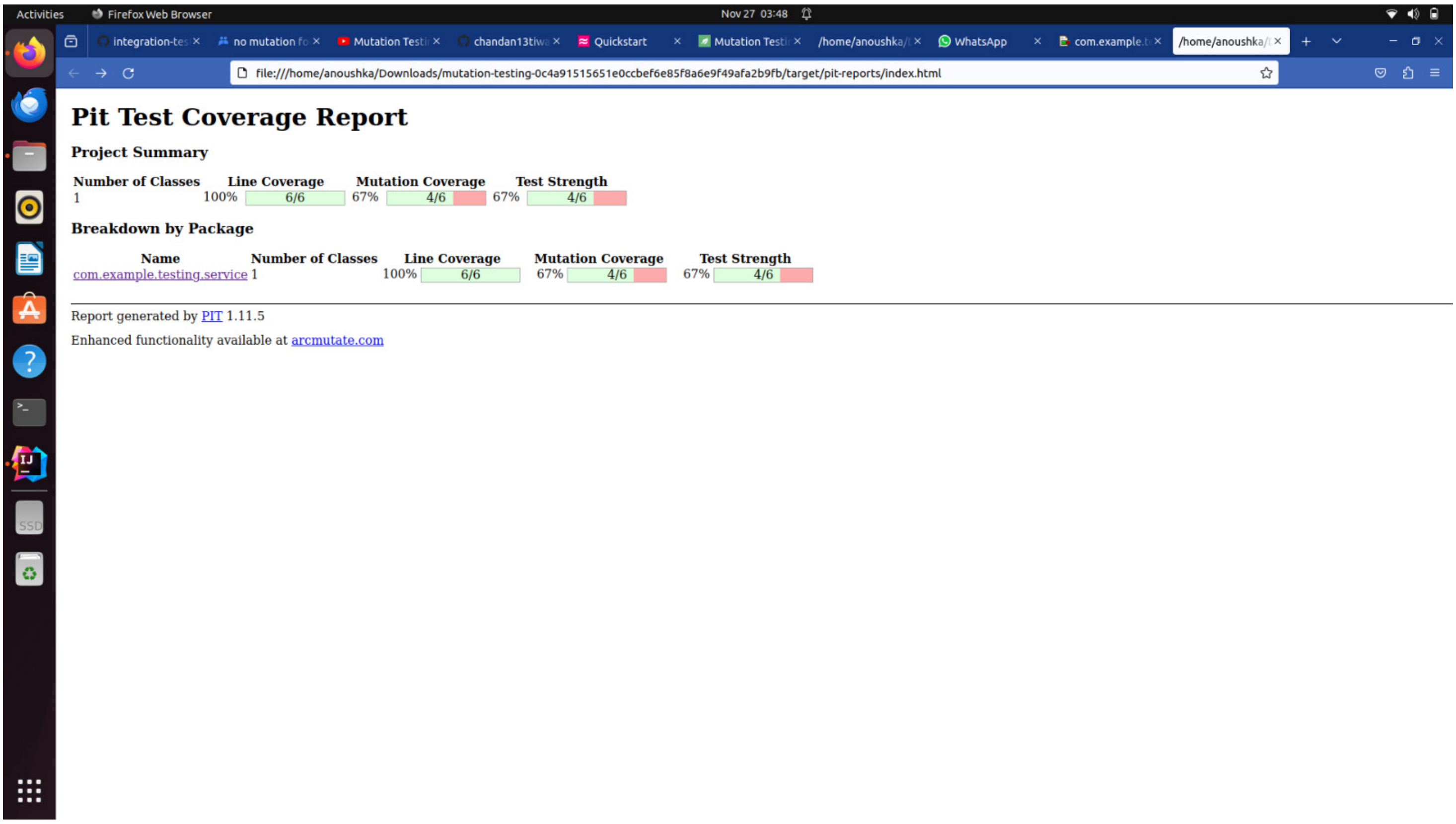
Source Files Sessions

com.example.testing.service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
OperationService	🟩	100%	🟩	100%	0	4	0	6	0	2	0	1
Total	0 of 21	100%	0 of 4	100%	0	4	0	6	0	2	0	1

Created with JaCoCo 0.8.7.202105040129

c. Pitest coverage Project Summary



file:///home/anoushka/Downloads/mutation-testing-0c4a91515651e0ccbef6e85f8a6e9f49afa2b9fb/target/pit-reports/index.html

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
1	100% 6/6	67% 4/6	67% 4/6

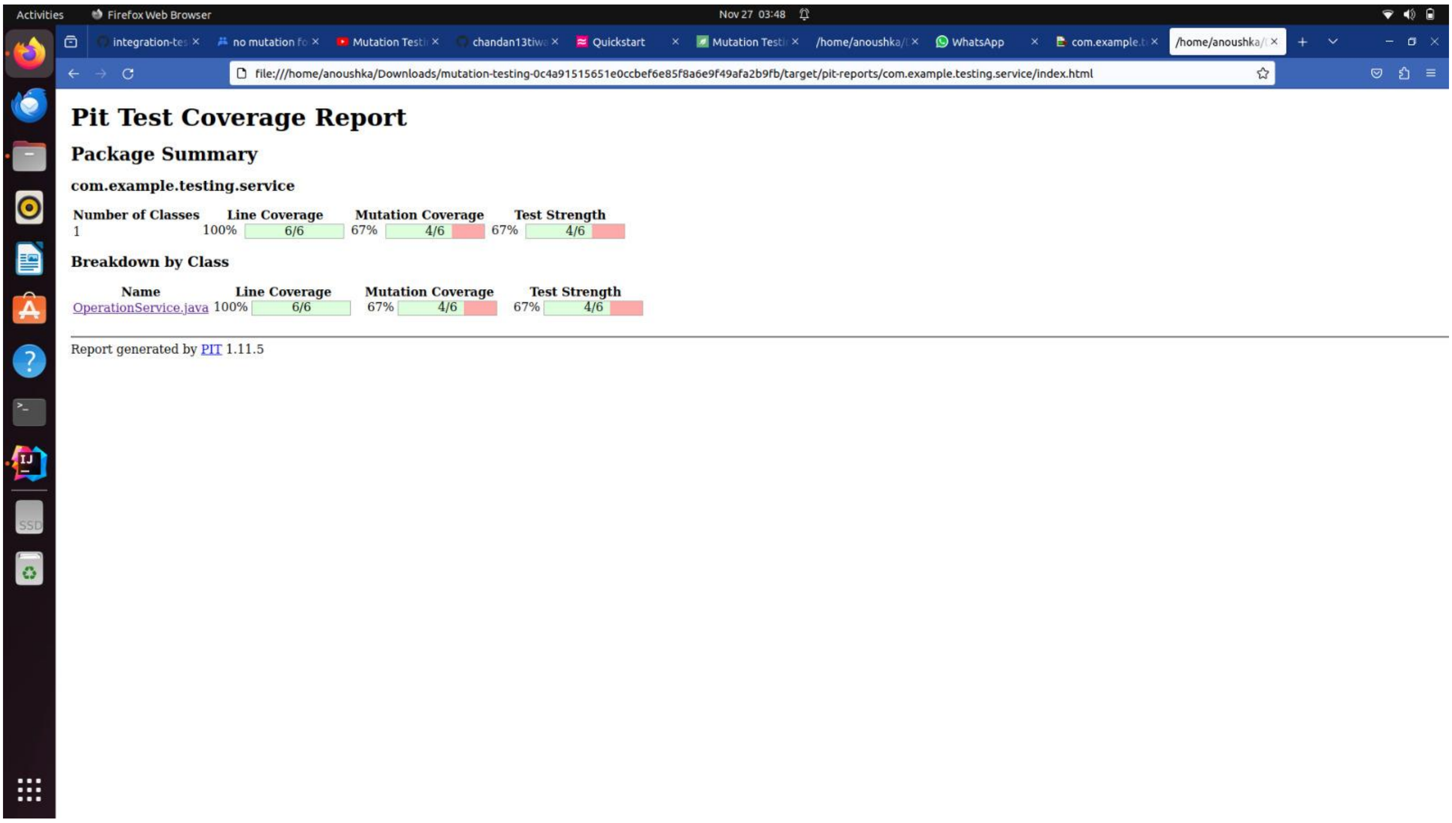
Breakdown by Package

Name	Number of Classes	Line Coverage	Mutation Coverage	Test Strength
com.example.testing.service	1	100% 6/6	67% 4/6	67% 4/6

Report generated by PIT 1.11.5

Enhanced functionality available at arcmutate.com

d. Pitest coverage Package Summary



Pit Test Coverage Report

Package Summary

com.example.testing.service

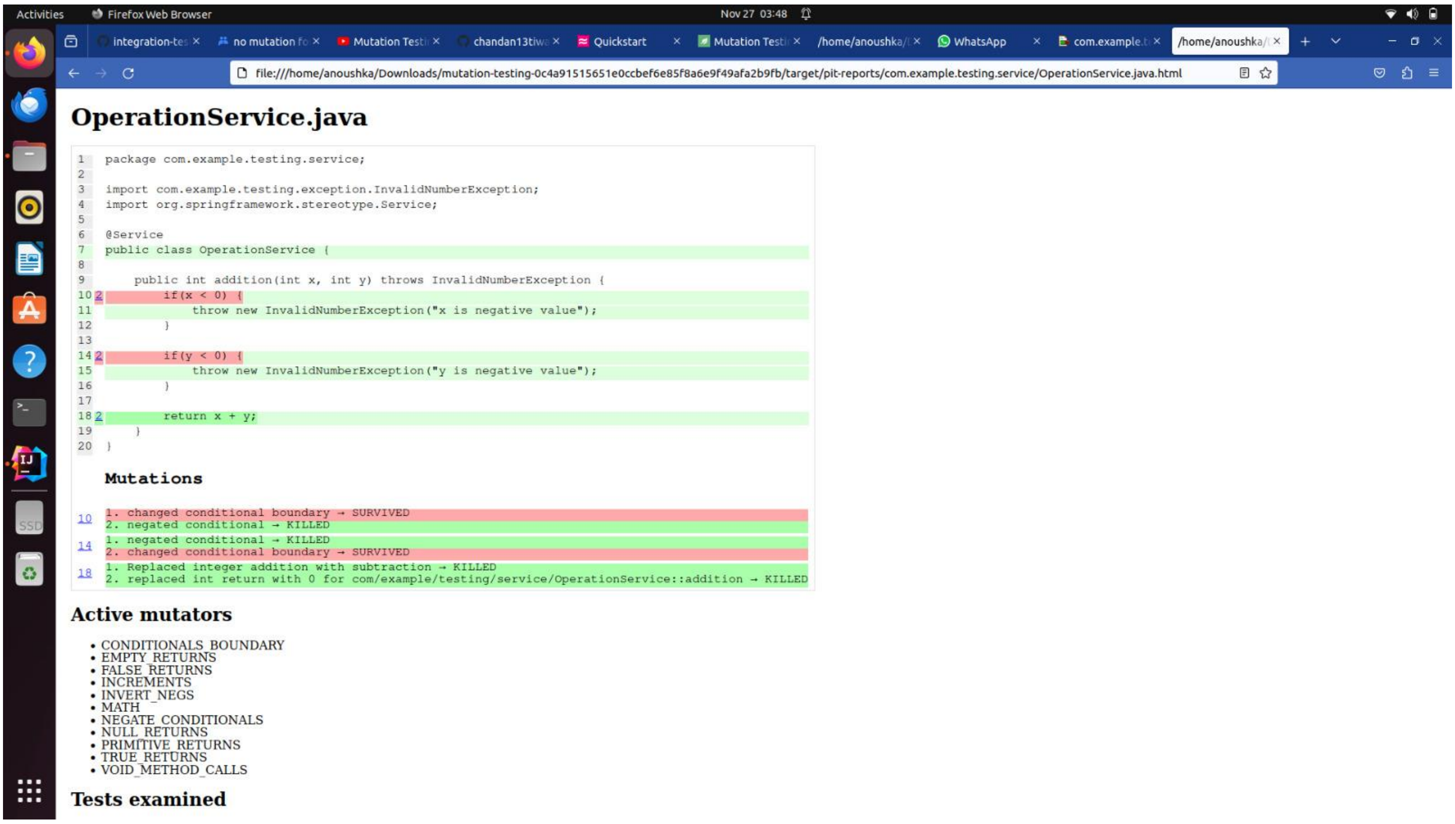
Number of Classes	Line Coverage	Mutation Coverage	Test Strength
1	100% 6/6	67% 4/6	67% 4/6

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
OperationService.java	100% 6/6	67% 4/6	67% 4/6

Report generated by PIT 1.11.5

e. Pitest Report for the package



OperationService.java

```
1    package com.example.testing.service;
2
3    import com.example.testing.exception.InvalidNumberException;
4    import org.springframework.stereotype.Service;
5
6    @Service
7    public class OperationService {
8
9        public int addition(int x, int y) throws InvalidNumberException {
10 2         if(x < 0) {
11               throw new InvalidNumberException("x is negative value");
12           }
13
14 2         if(y < 0) {
15               throw new InvalidNumberException("y is negative value");
16           }
17
18 2         return x + y;
19       }
20   }
```

Mutations

```
10  1. changed conditional boundary → SURVIVED
    2. negated conditional → KILLED
14  1. negated conditional → KILLED
    2. changed conditional boundary → SURVIVED
18  1. Replaced integer addition with subtraction → KILLED
    2. replaced int return with 0 for com/example/testing/service/OperationService::addition → KILLED
```

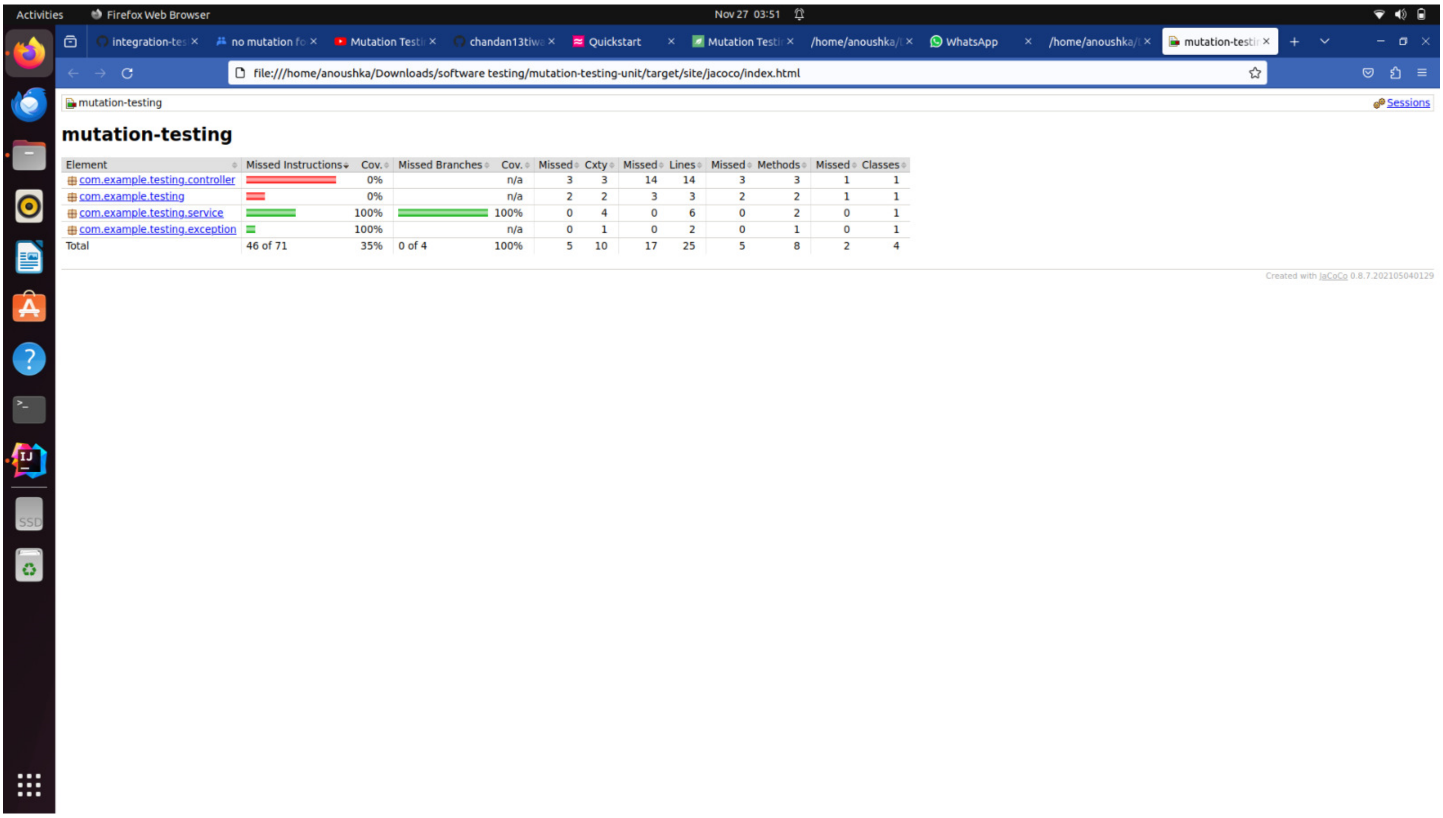
Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

2. Unit Testing after killing mutants

a. Code coverage using Jacoco on the Project



file:///home/anoushka/Downloads/software testing/mutation-testing-unit/target/site/jacoco/index.html

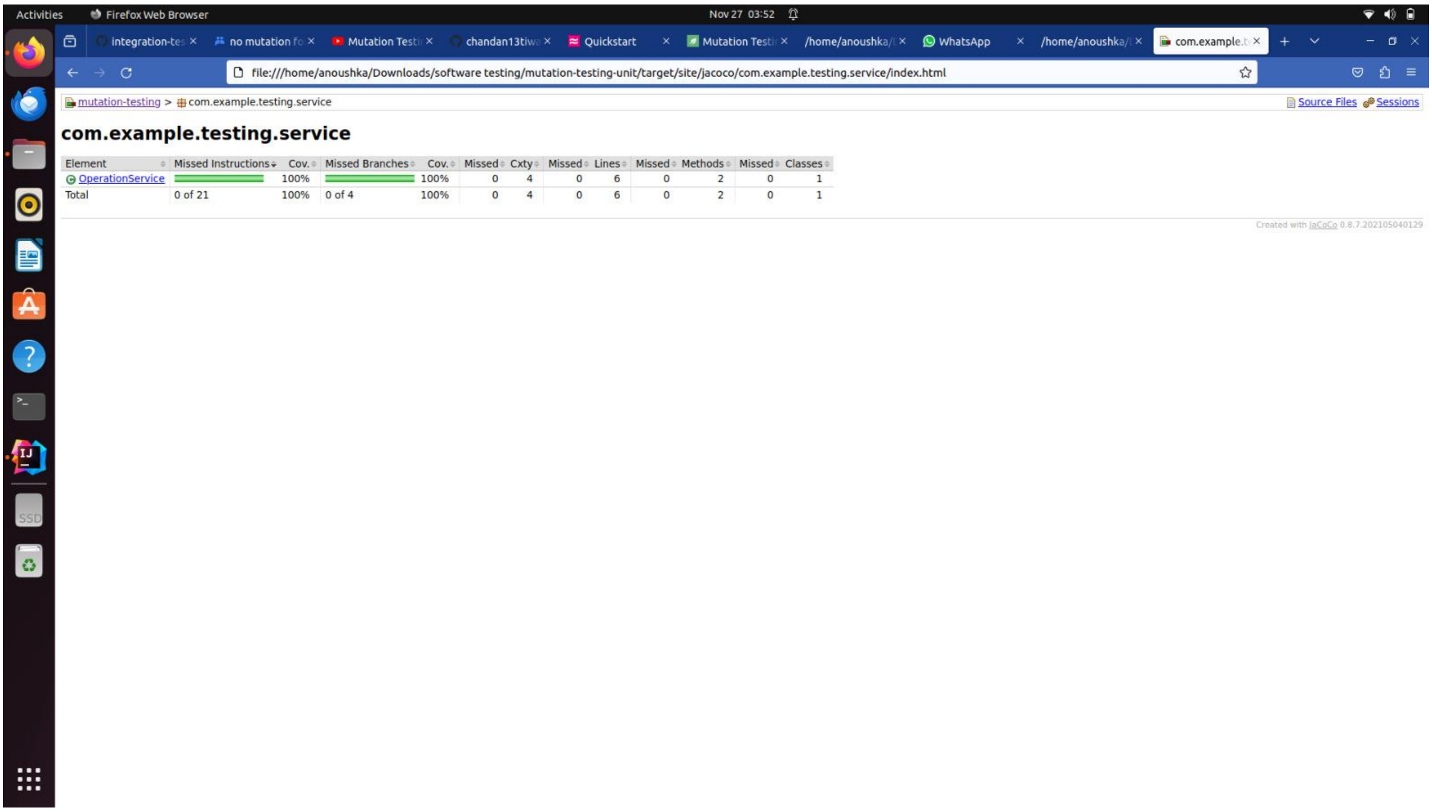
mutation-testing

mutation-testing

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.example.testing.controller	🟥	0%		n/a	3	3	14	14	3	3	1	1
com.example.testing	🟥	0%		n/a	2	2	3	3	2	2	1	1
com.example.testing.service	🟩	100%	🟩	100%	0	4	0	6	0	2	0	1
com.example.testing.exception	🟩	100%		n/a	0	1	0	2	0	1	0	1
Total	46 of 71	35%	0 of 4	100%	5	10	17	25	5	8	2	4

Created with JaCoCo 0.8.7.202105040129

b. Code coverage using Jacoco for package *com.example.testing.service*



file:///home/anoushka/Downloads/software testing/mutation-testing-unit/target/site/jacoco/com.example.testing.service/index.html

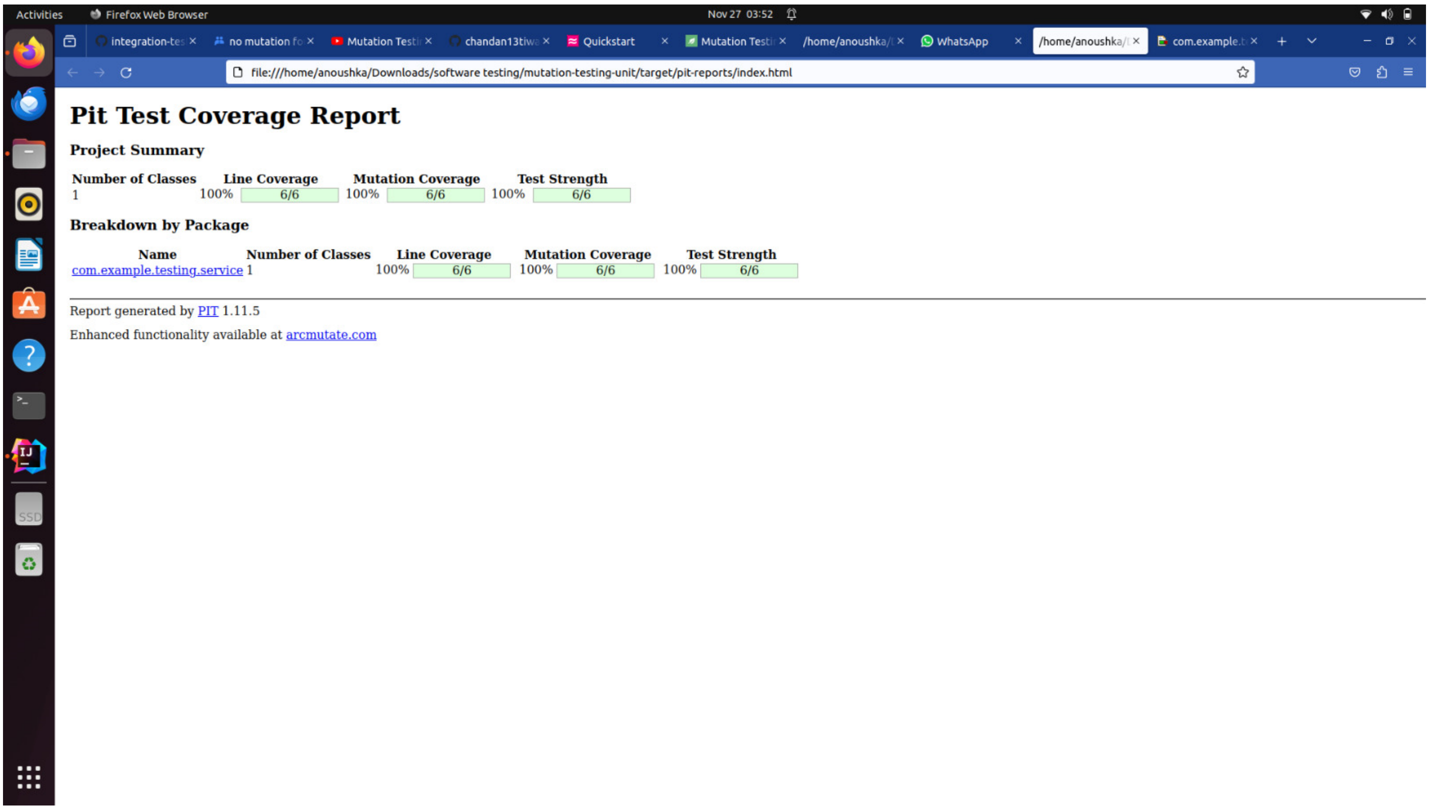
mutation-testing > com.example.testing.service

com.example.testing.service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
OperationService	🟩	100%	🟩	100%	0	4	0	6	0	2	0	1
Total	0 of 21	100%	0 of 4	100%	0	4	0	6	0	2	0	1

Created with JaCoCo 0.8.7.202105040129

c. Pitest coverage Project Summary



Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
1	100% 6/6	100% 6/6	100% 6/6

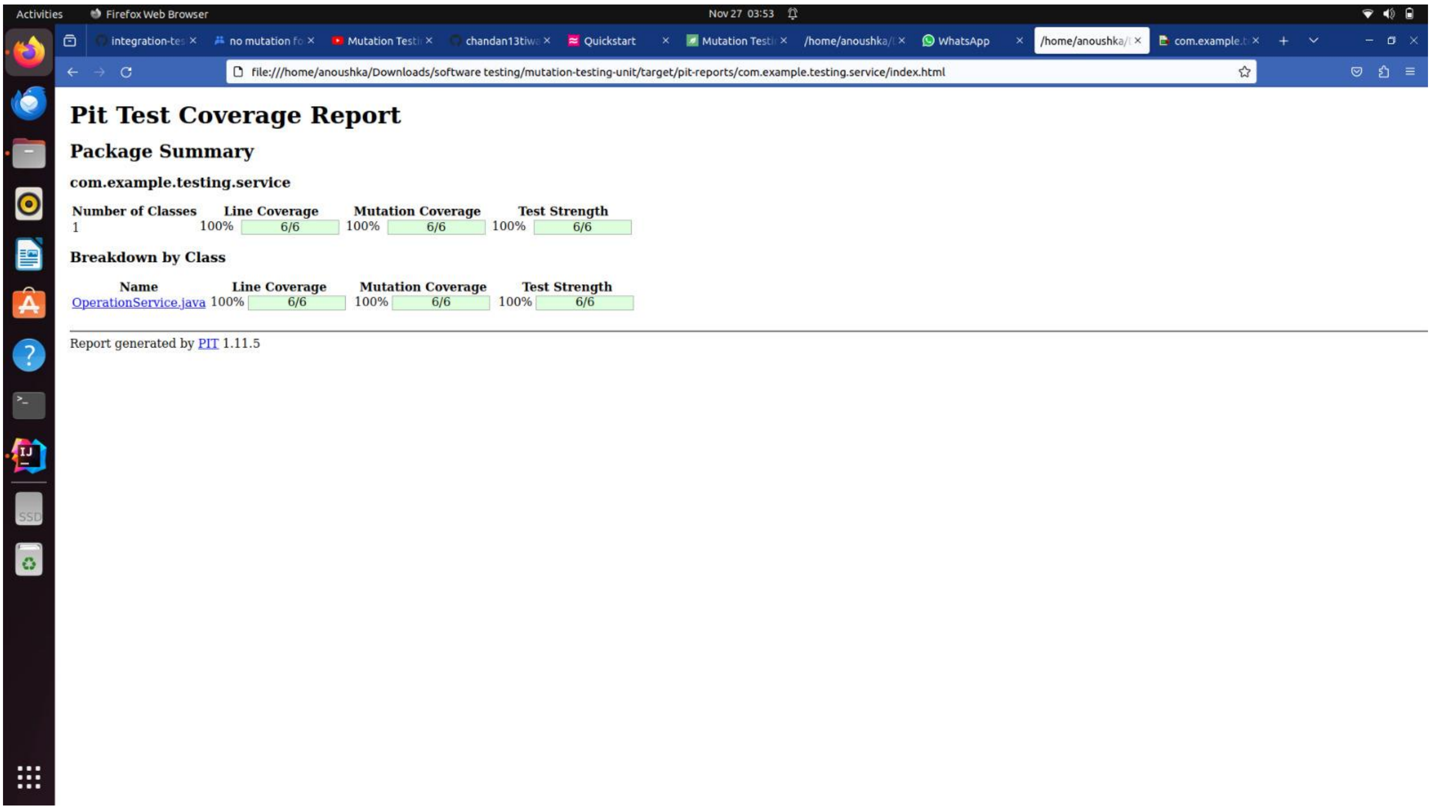
Breakdown by Package

Name	Number of Classes	Line Coverage	Mutation Coverage	Test Strength
com.example.testing.service	1	100% 6/6	100% 6/6	100% 6/6

Report generated by PIT 1.11.5

Enhanced functionality available at arcmutate.com

d. Pitest coverage Package Summary



Pit Test Coverage Report

Package Summary

com.example.testing.service

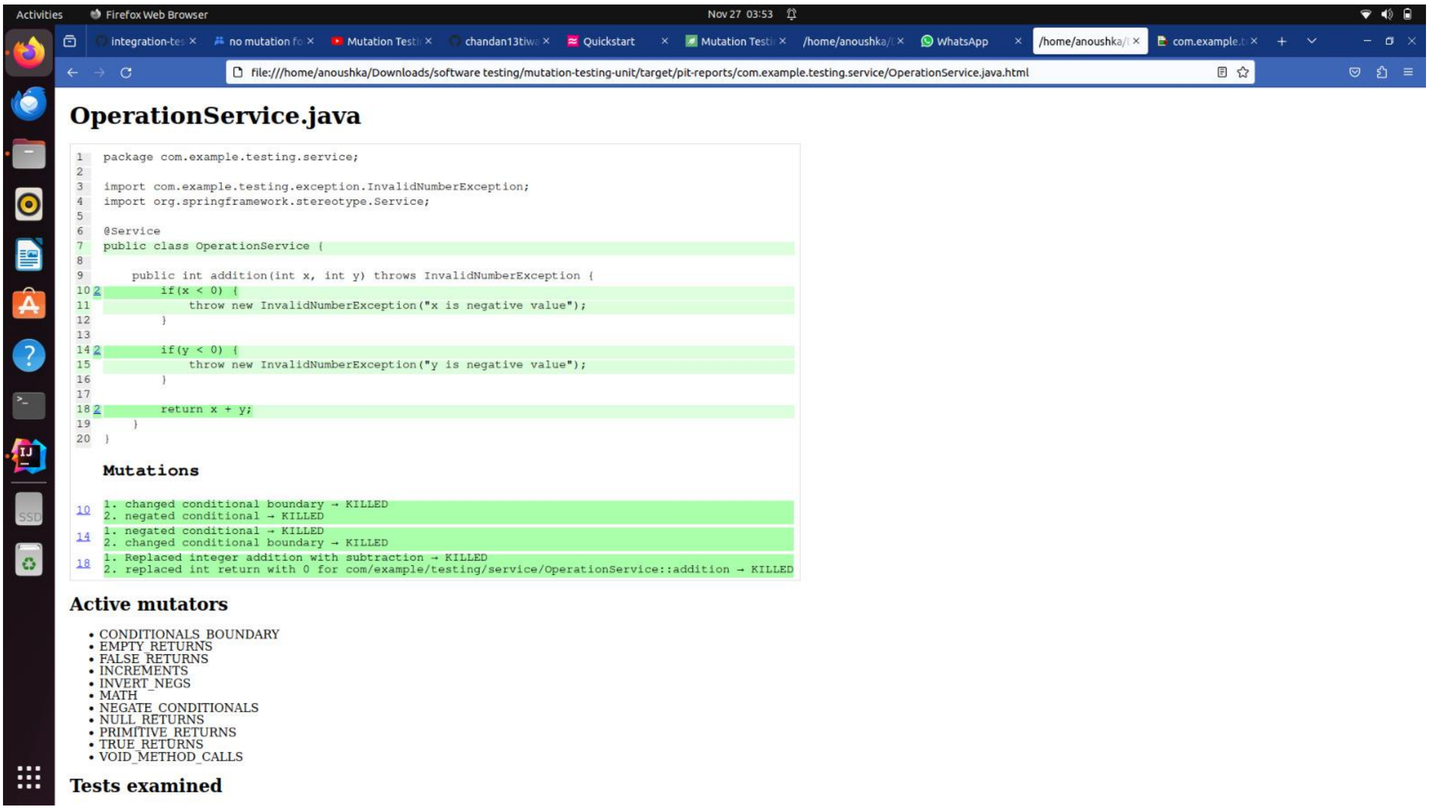
Number of Classes	Line Coverage	Mutation Coverage	Test Strength
1	100% 6/6	100% 6/6	100% 6/6

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
OperationService.java	100% 6/6	100% 6/6	100% 6/6

Report generated by PIT 1.11.5

e. Pitest Report for the package



file:///home/anoushka/Downloads/software testing/mutation-testing-unit/target/pit-reports/com.example.testing.service/OperationService.java.html

OperationService.java

```
package com.example.testing.service;

import com.example.testing.exception.InvalidNumberException;
import org.springframework.stereotype.Service;

@Service
public class OperationService {

    public int addition(int x, int y) throws InvalidNumberException {
2       if(x < 0) {
            throw new InvalidNumberException("x is negative value");
        }

2       if(y < 0) {
            throw new InvalidNumberException("y is negative value");
        }

2       return x + y;
    }
}
```

Mutations

```
10  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
14  1. negated conditional → KILLED
    2. changed conditional boundary → KILLED
18  1. Replaced integer addition with subtraction → KILLED
    2. replaced int return with 0 for com/example/testing/service/OperationService::addition → KILLED
```

Active mutators

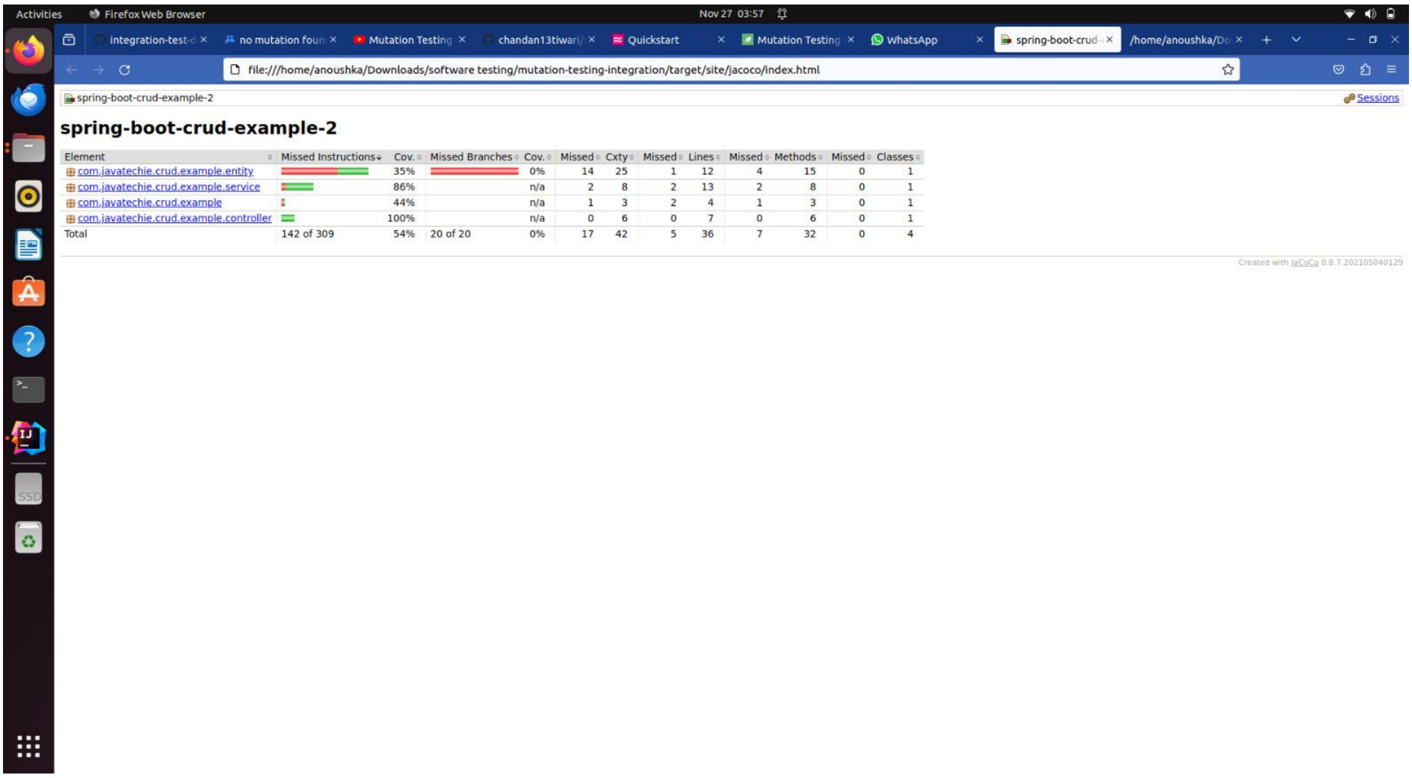
- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

INTEGRATION TESTING

1. Integration Testing before killing mutants

a. Code coverage using Jacoco on the Project



file:///home/anoushka/Downloads/software testing/mutation-testing-integration/target/site/jacoco/index.html

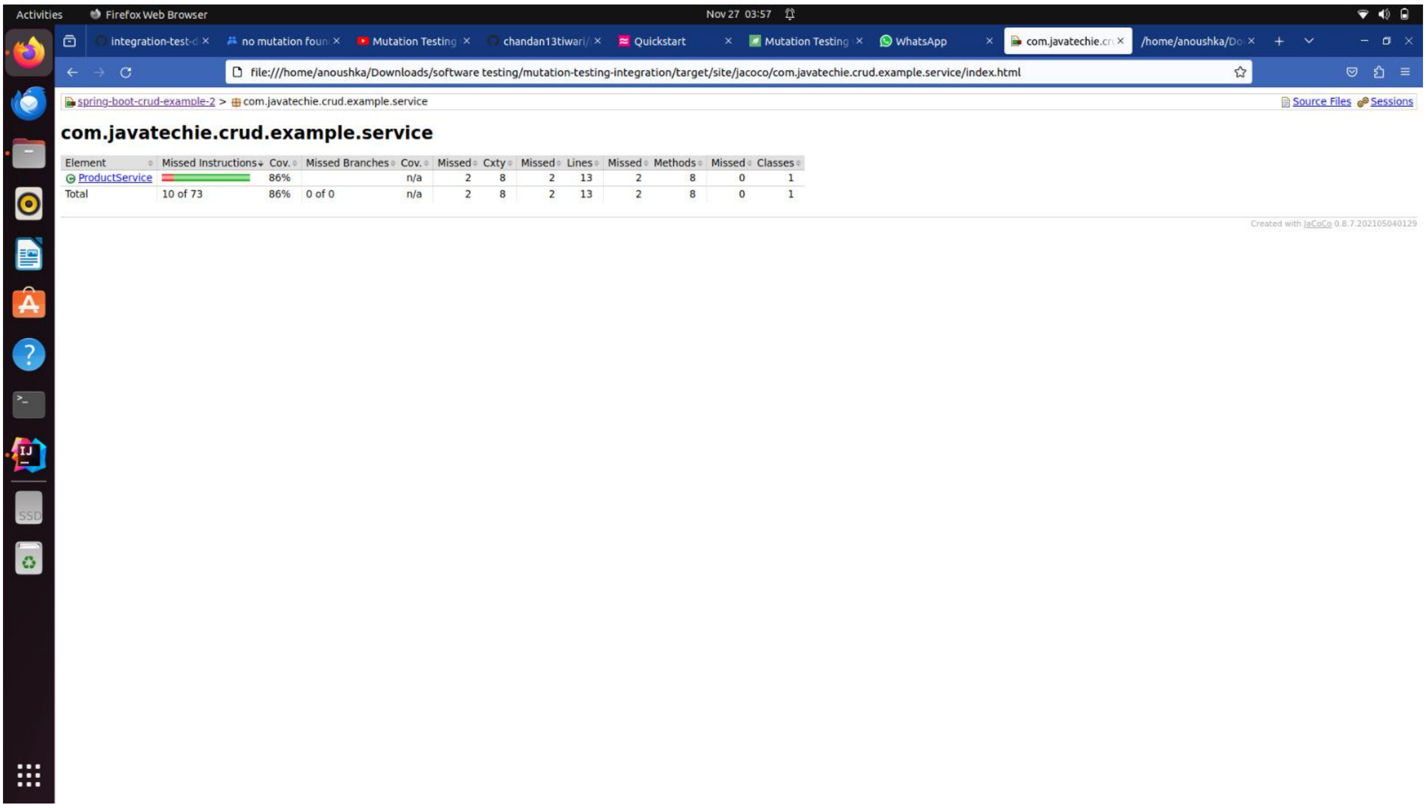
spring-boot-crud-example-2

spring-boot-crud-example-2

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.javatechie.crud.example.entity		35%		0%	14	25	1	12	4	15	0	1
com.javatechie.crud.example.service		86%		n/a	2	8	2	13	2	8	0	1
com.javatechie.crud.example		44%		n/a	1	3	2	4	1	3	0	1
com.javatechie.crud.example.controller		100%		n/a	0	6	0	7	0	6	0	1
Total	142 of 309	54%	20 of 20	0%	17	42	5	36	7	32	0	4

Created with JaCoCo 0.8.7.202105040129

b. Code coverage using Jacoco for package *com.javatechie.crud.example.service*



spring-boot-crud-example-2 > com.javatechie.crud.example.service

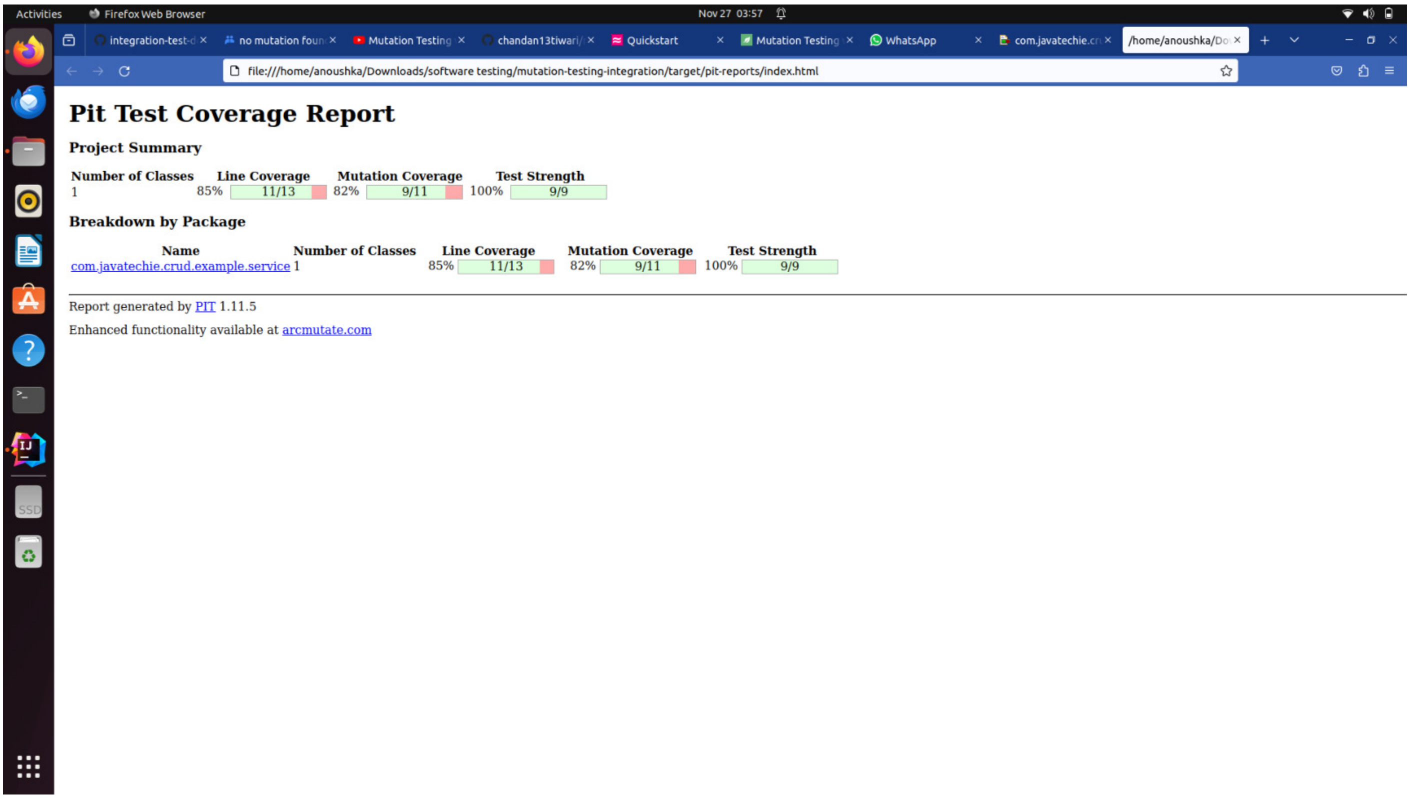
Source Files Sessions

com.javatechie.crud.example.service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
ProductService	🟥🟩	86%		n/a	2	8	2	13	2	8	0	1
Total	10 of 73	86%	0 of 0	n/a	2	8	2	13	2	8	0	1

Created with JaCoCo 0.8.7.202105040129

c. Pitest coverage Project Summary



Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
1	85%	11/13	82%	9/11	100%	9/9

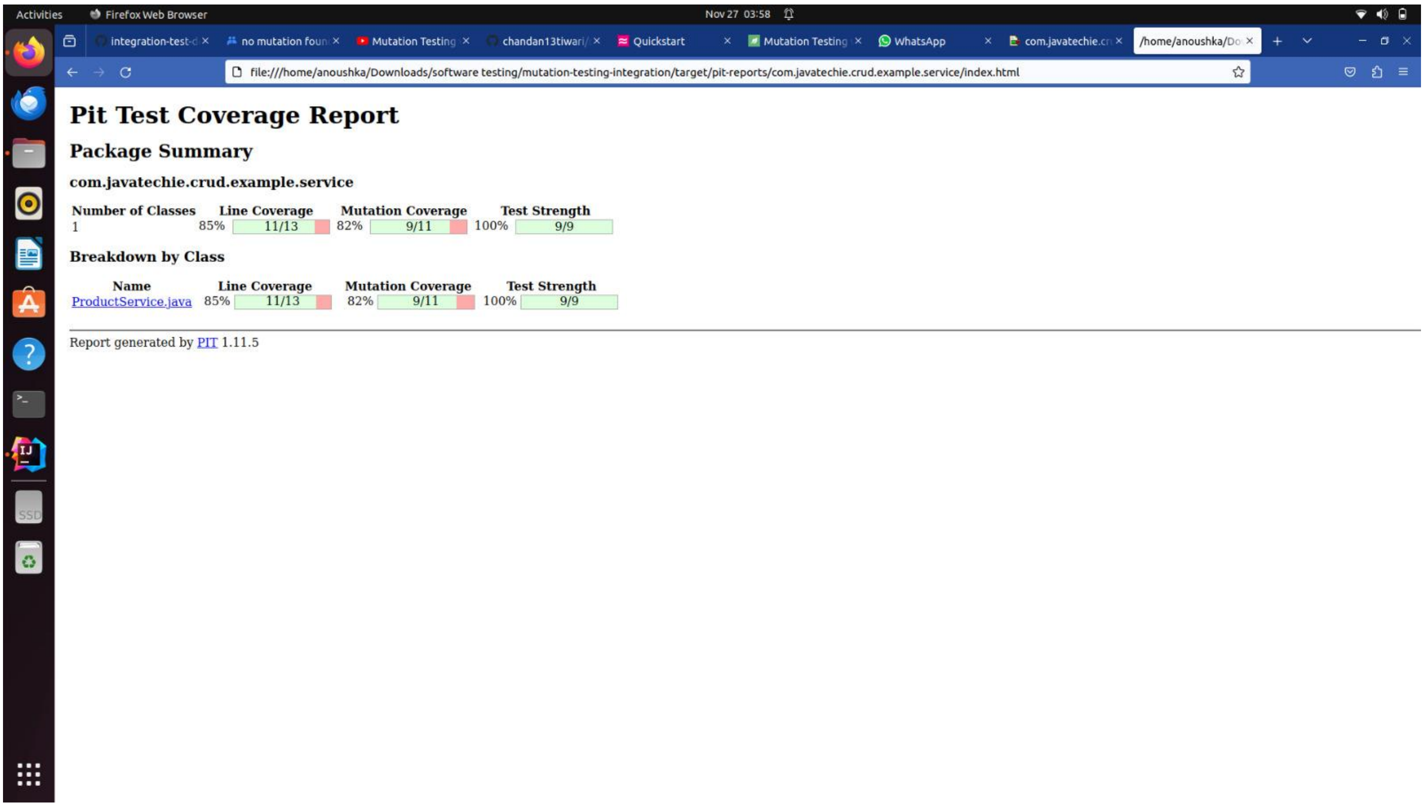
Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
com.javatechie.crud.example.service	1	85%	11/13	82%	9/11	100%	9/9

Report generated by PIT 1.11.5

Enhanced functionality available at arcmutate.com

d. Pitest coverage Package Summary



Pit Test Coverage Report

Package Summary

com.javatechie.crud.example.service

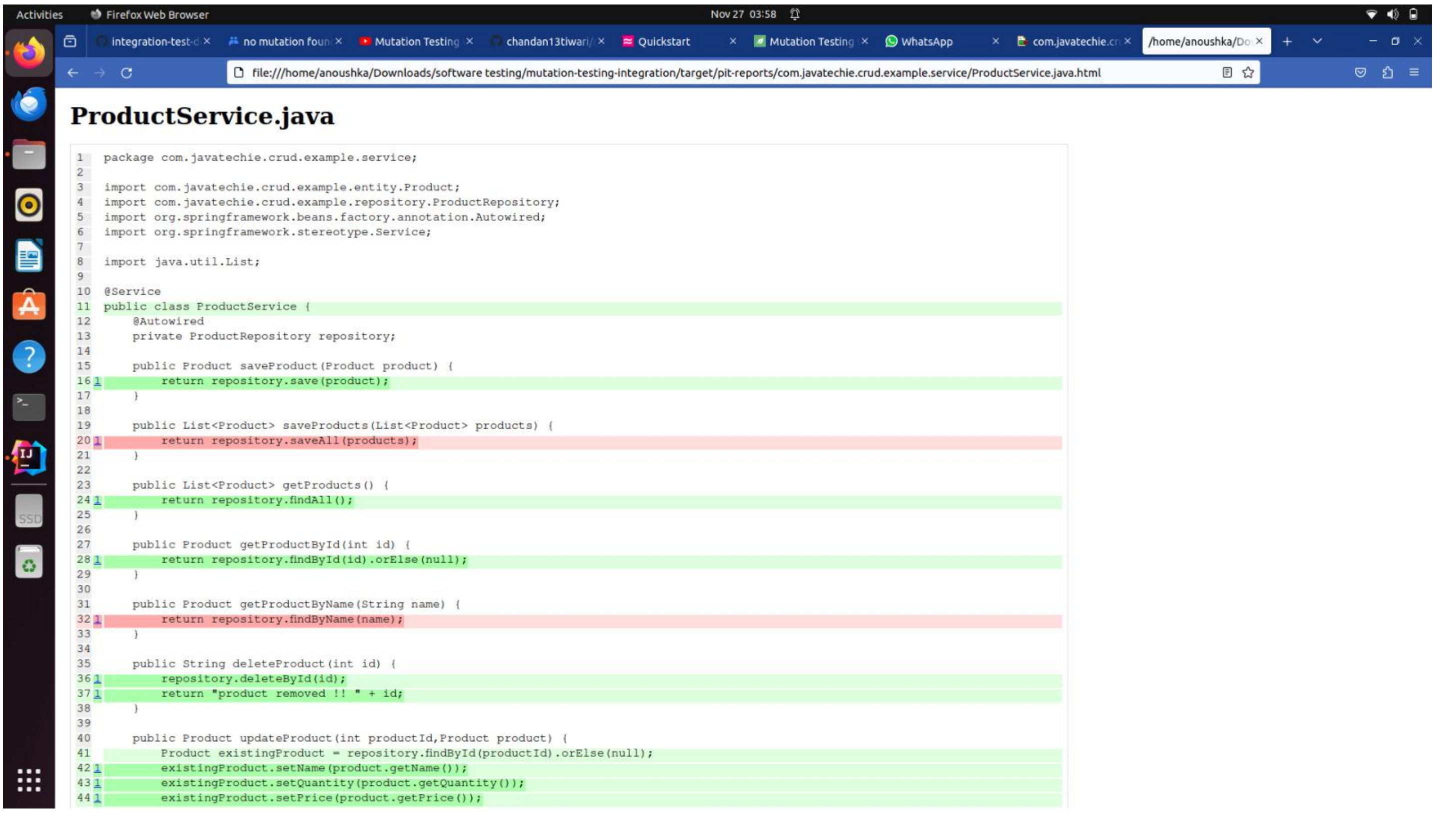
Number of Classes	Line Coverage	Mutation Coverage	Test Strength
1	85% 11/13	82% 9/11	100% 9/9

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
ProductService.java	85% 11/13	82% 9/11	100% 9/9

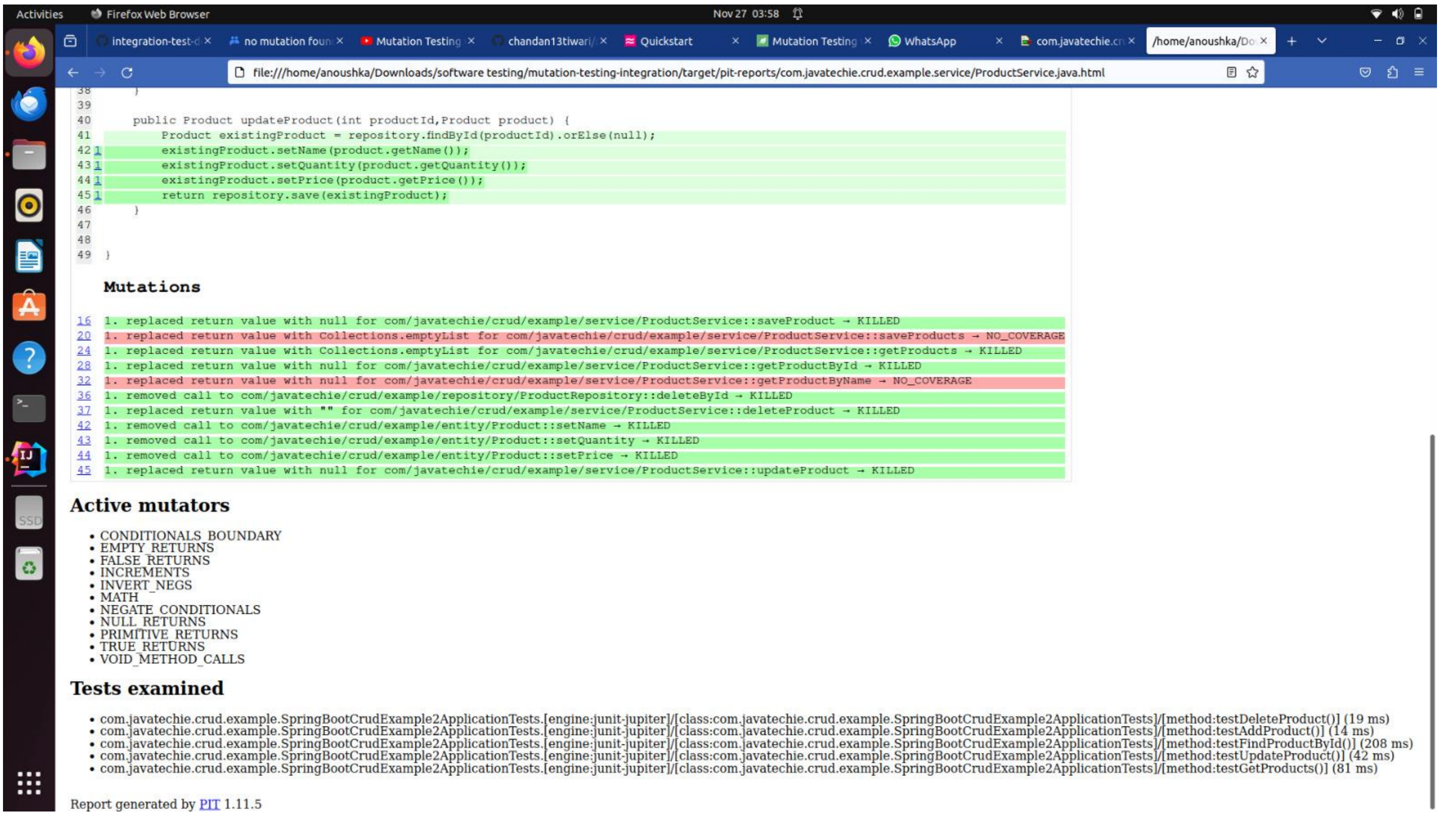
Report generated by PIT 1.11.5

e. Pitest Report for the package



ProductService.java

```
package com.javatechie.crud.example.service;

import com.javatechie.crud.example.entity.Product;
import com.javatechie.crud.example.repository.ProductRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class ProductService {
    @Autowired
    private ProductRepository repository;

    public Product saveProduct(Product product) {
        return repository.save(product);
    }

    public List<Product> saveProducts(List<Product> products) {
        return repository.saveAll(products);
    }

    public List<Product> getProducts() {
        return repository.findAll();
    }

    public Product getProductById(int id) {
        return repository.findById(id).orElse(null);
    }

    public Product getProductByName(String name) {
        return repository.findByName(name);
    }

    public String deleteProduct(int id) {
        repository.deleteById(id);
        return "product removed !! " + id;
    }

    public Product updateProduct(int productId,Product product) {
        Product existingProduct = repository.findById(productId).orElse(null);
        existingProduct.setName(product.getName());
        existingProduct.setQuantity(product.getQuantity());
        existingProduct.setPrice(product.getPrice());
```



file:///home/anoushka/Downloads/software testing/mutation-testing-integration/target/pit-reports/com.javatechie.crud.example.service/ProductService.java.html

```
38        }
39
40        public Product updateProduct(int productId,Product product) {
41            Product existingProduct = repository.findById(productId).orElse(null);
42 1          existingProduct.setName(product.getName());
43 1          existingProduct.setQuantity(product.getQuantity());
44 1          existingProduct.setPrice(product.getPrice());
45 1          return repository.save(existingProduct);
46        }
47
48
49 }
```

Mutations

```
16 1. replaced return value with null for com/javatechie/crud/example/service/ProductService::saveProduct → KILLED
20 1. replaced return value with Collections.emptyList for com/javatechie/crud/example/service/ProductService::saveProducts → NO_COVERAGE
24 1. replaced return value with Collections.emptyList for com/javatechie/crud/example/service/ProductService::getProducts → KILLED
28 1. replaced return value with null for com/javatechie/crud/example/service/ProductService::getProductById → KILLED
32 1. replaced return value with null for com/javatechie/crud/example/service/ProductService::getProductByName → NO_COVERAGE
36 1. removed call to com/javatechie/crud/example/repository/ProductRepository::deleteById → KILLED
37 1. replaced return value with "" for com/javatechie/crud/example/service/ProductService::deleteProduct → KILLED
42 1. removed call to com/javatechie/crud/example/entity/Product::setName → KILLED
43 1. removed call to com/javatechie/crud/example/entity/Product::setQuantity → KILLED
44 1. removed call to com/javatechie/crud/example/entity/Product::setPrice → KILLED
45 1. replaced return value with null for com/javatechie/crud/example/service/ProductService::updateProduct → KILLED
```

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

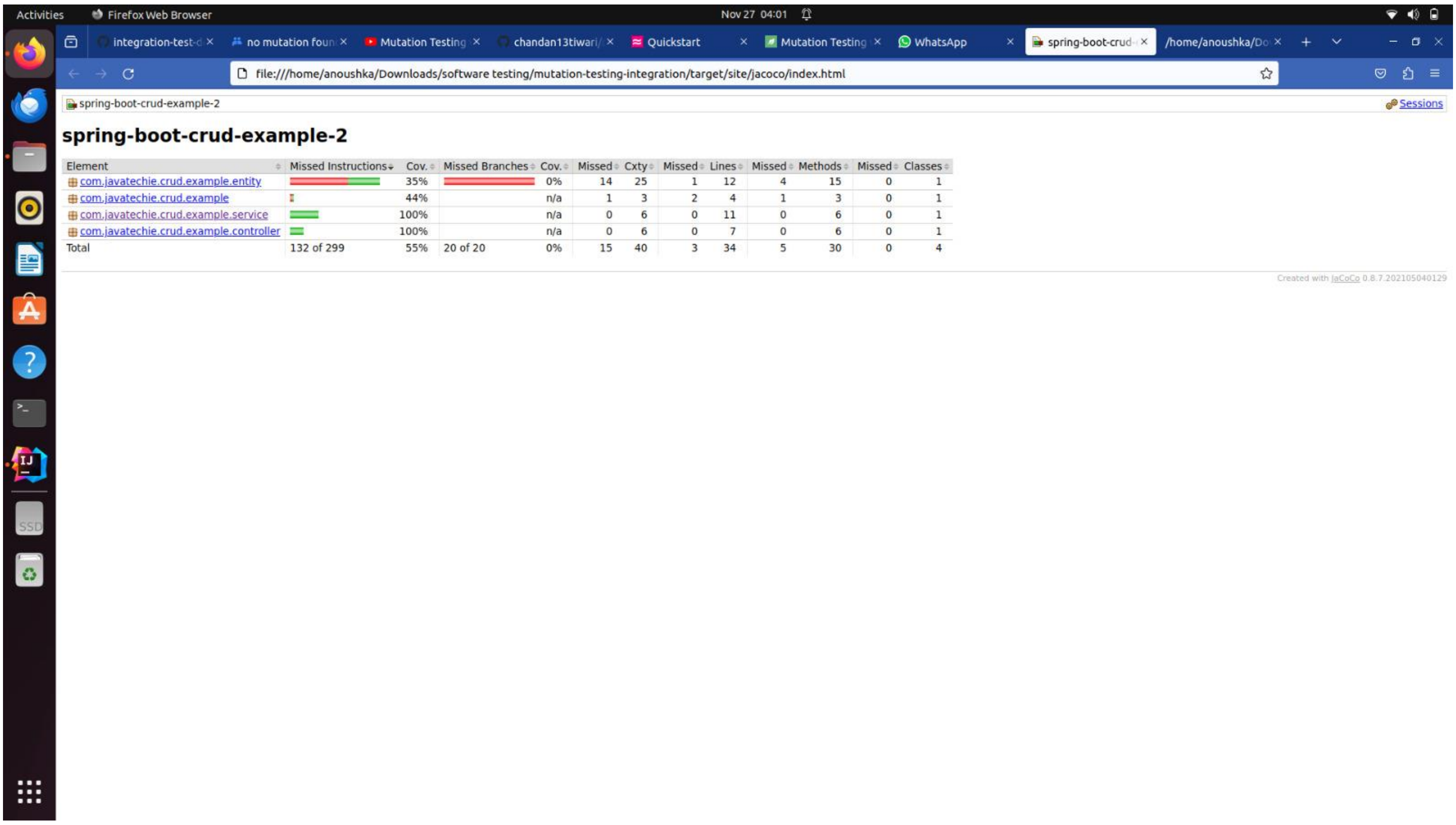
Tests examined

- com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests.[engine:junit-jupiter]/[class:com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests]/[method:testDeleteProduct()] (19 ms)
- com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests.[engine:junit-jupiter]/[class:com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests]/[method:testAddProduct()] (14 ms)
- com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests.[engine:junit-jupiter]/[class:com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests]/[method:testFindProductById()] (208 ms)
- com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests.[engine:junit-jupiter]/[class:com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests]/[method:testUpdateProduct()] (42 ms)
- com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests.[engine:junit-jupiter]/[class:com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests]/[method:testGetProducts()] (81 ms)

Report generated by PIT 1.11.5

2. Integration Testing before killing mutants

a. Code coverage using Jacoco on the Project



file:///home/anoushka/Downloads/software testing/mutation-testing-integration/target/site/jacoco/index.html

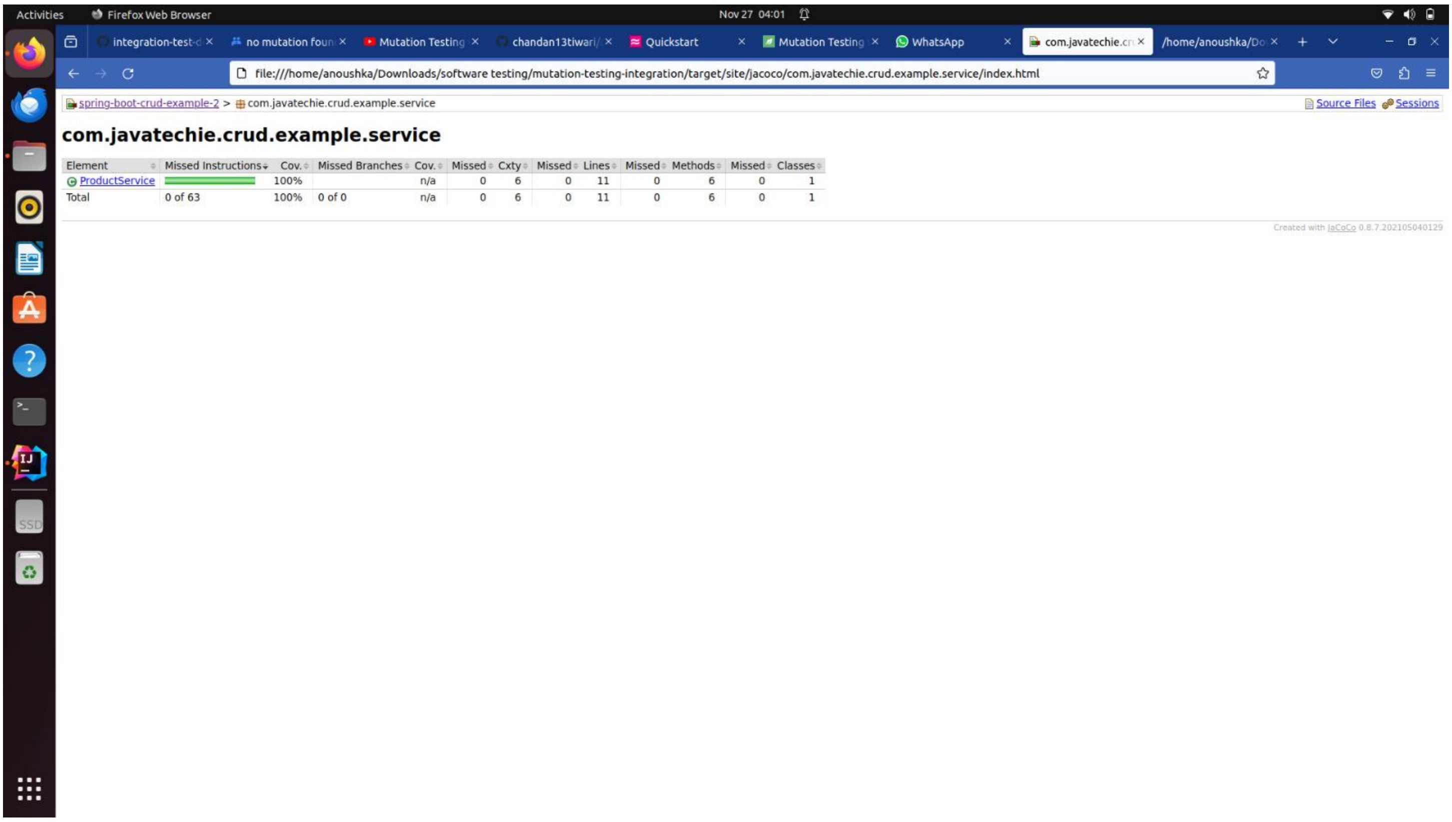
spring-boot-crud-example-2 Sessions

spring-boot-crud-example-2

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.javatechie.crud.example.entity		35%		0%	14	25	1	12	4	15	0	1
com.javatechie.crud.example		44%		n/a	1	3	2	4	1	3	0	1
com.javatechie.crud.example.service		100%		n/a	0	6	0	11	0	6	0	1
com.javatechie.crud.example.controller		100%		n/a	0	6	0	7	0	6	0	1
Total	132 of 299	55%	20 of 20	0%	15	40	3	34	5	30	0	4

Created with JaCoCo 0.8.7.202105040129

b. Code coverage using Jacoco for package *com.javatechie.crud.example.service*



file:///home/anoushka/Downloads/software testing/mutation-testing-integration/target/site/jacoco/com.javatechie.crud.example.service/index.html

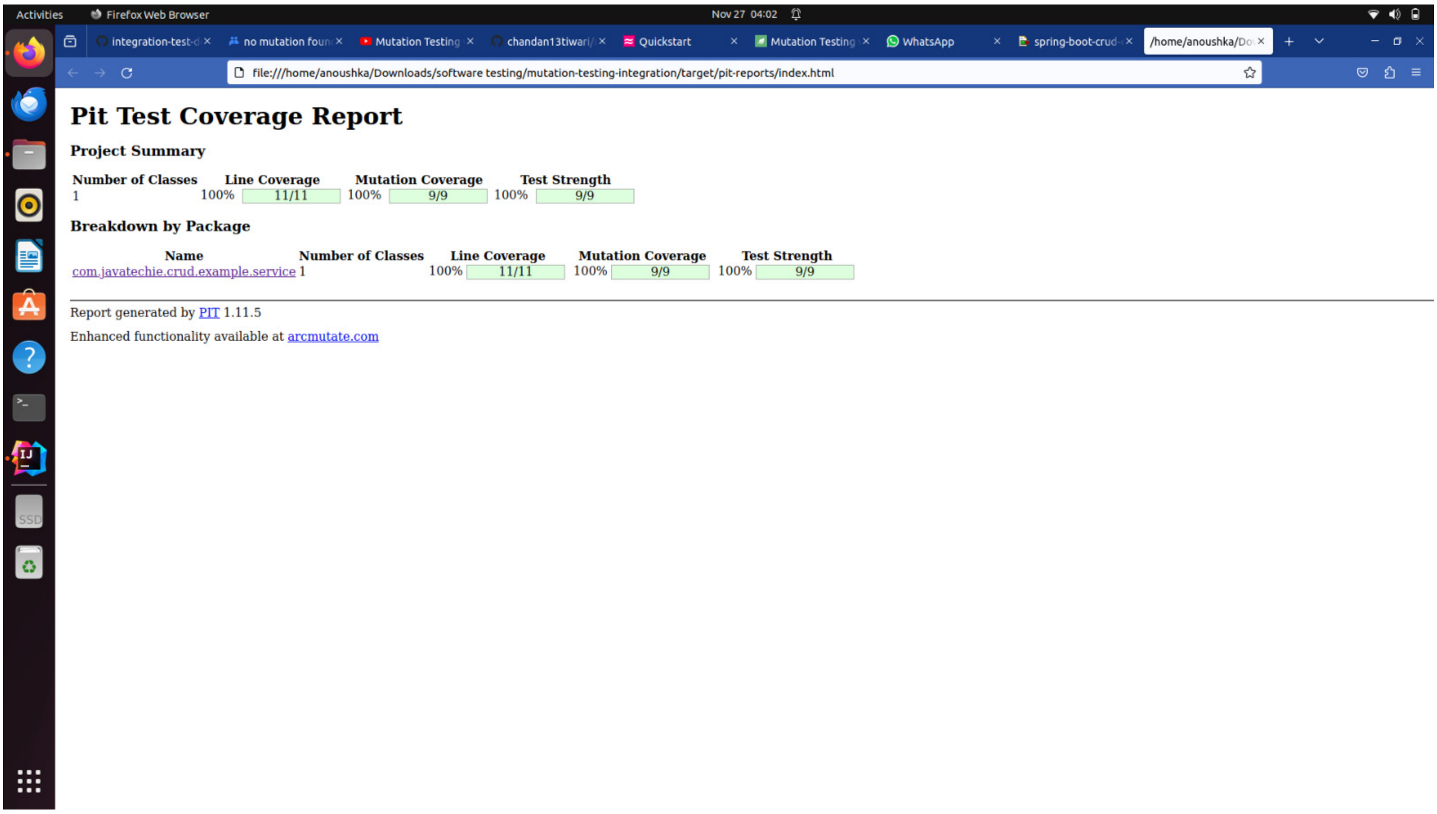
spring-boot-crud-example-2 > com.javatechie.crud.example.service

com.javatechie.crud.example.service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
ProductService	🟩	100%		n/a	0	6	0	11	0	6	0	1
Total	0 of 63	100%	0 of 0	n/a	0	6	0	11	0	6	0	1

Created with JaCoCo 0.8.7.202105040129

c. Pitest coverage Project Summary



file:///home/anoushka/Downloads/software testing/mutation-testing-integration/target/pit-reports/index.html

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
1	100%	11/11	100%	9/9	100%	9/9

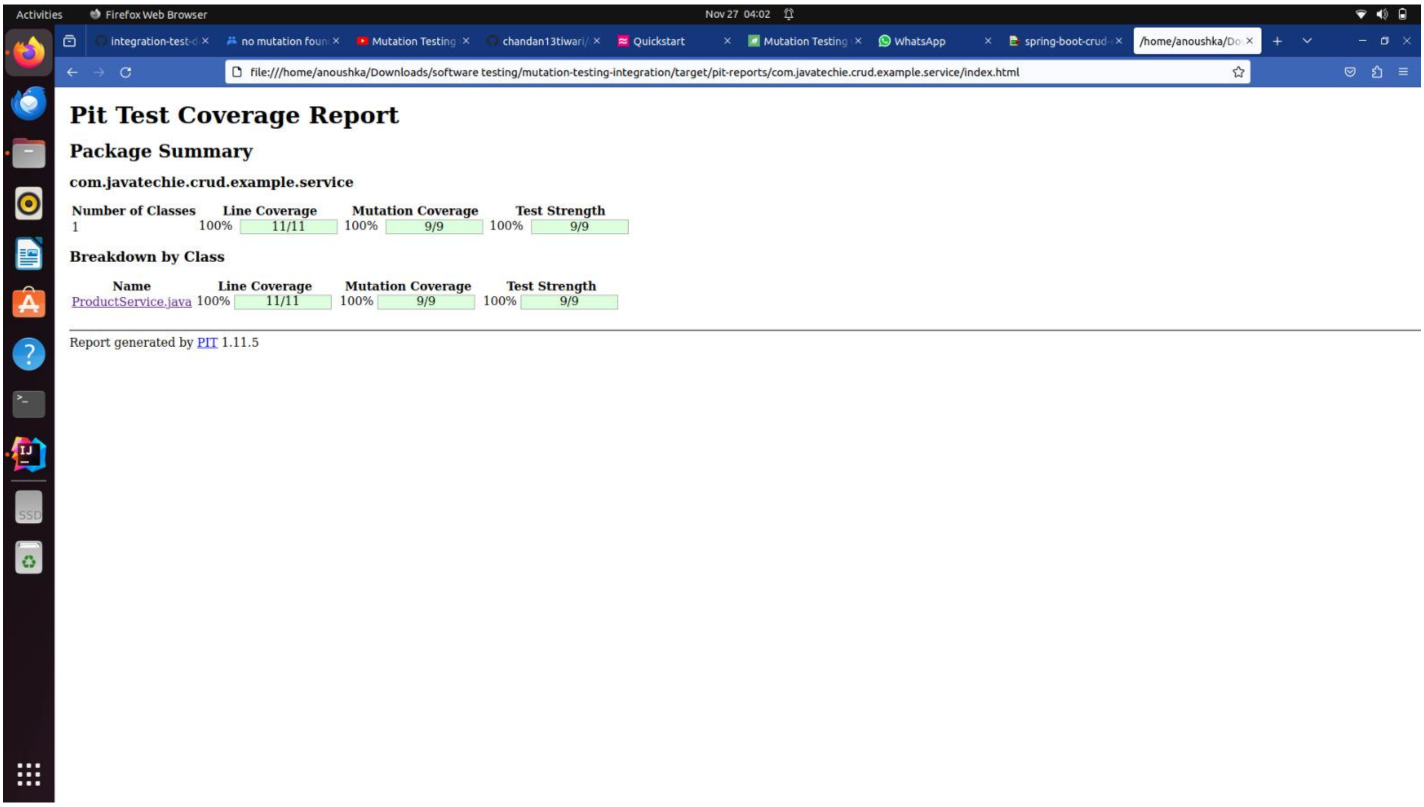
Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
com.javatechie.crud.example.service	1	100%	11/11	100%	9/9	100%	9/9

Report generated by PIT 1.11.5

Enhanced functionality available at arcmutate.com

d. Pitest coverage Package Summary



file:///home/anoushka/Downloads/software testing/mutation-testing-integration/target/pit-reports/com.javatechie.crud.example.service/index.html

Pit Test Coverage Report

Package Summary

com.javatechie.crud.example.service

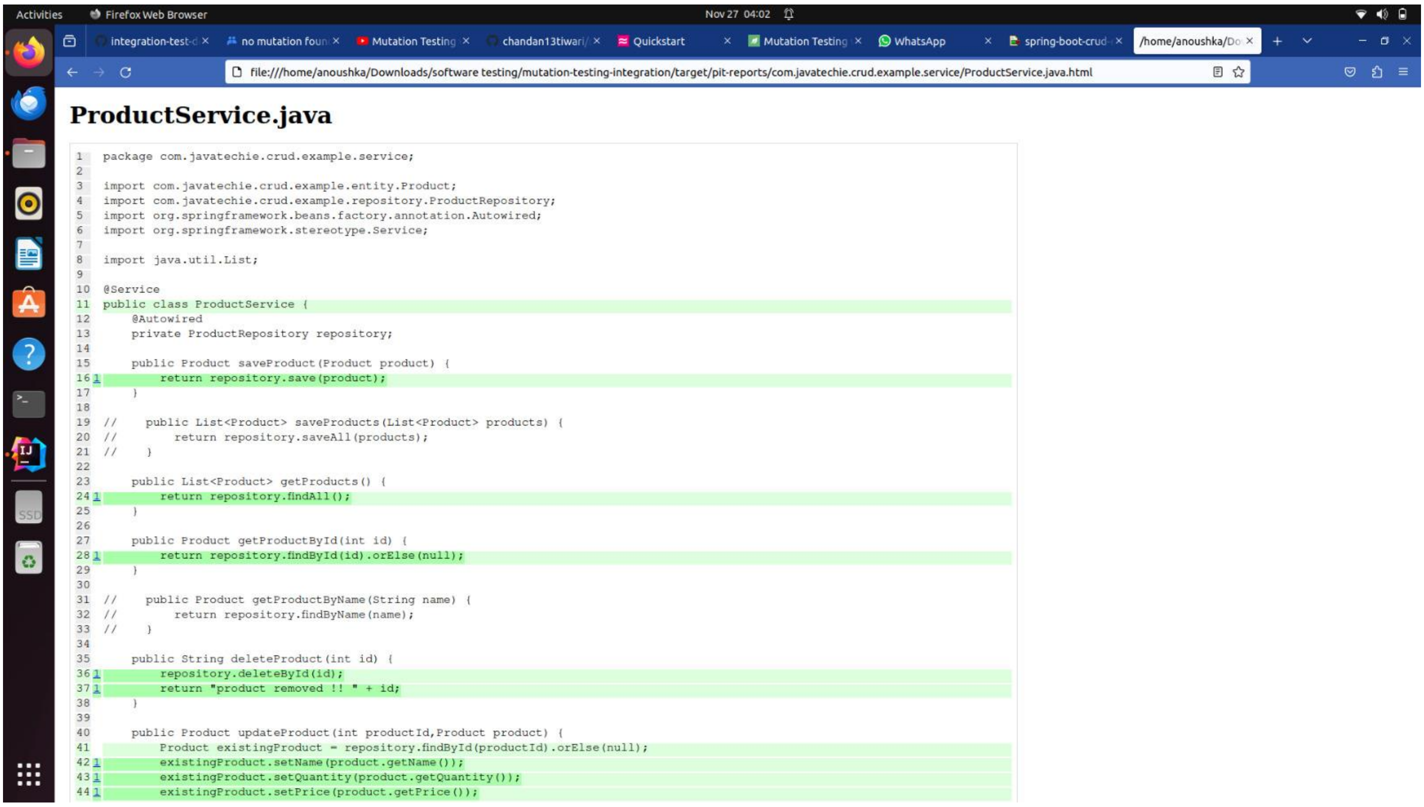
Number of Classes	Line Coverage	Mutation Coverage	Test Strength
1	100% 11/11	100% 9/9	100% 9/9

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
ProductService.java	100% 11/11	100% 9/9	100% 9/9

Report generated by PIT 1.11.5

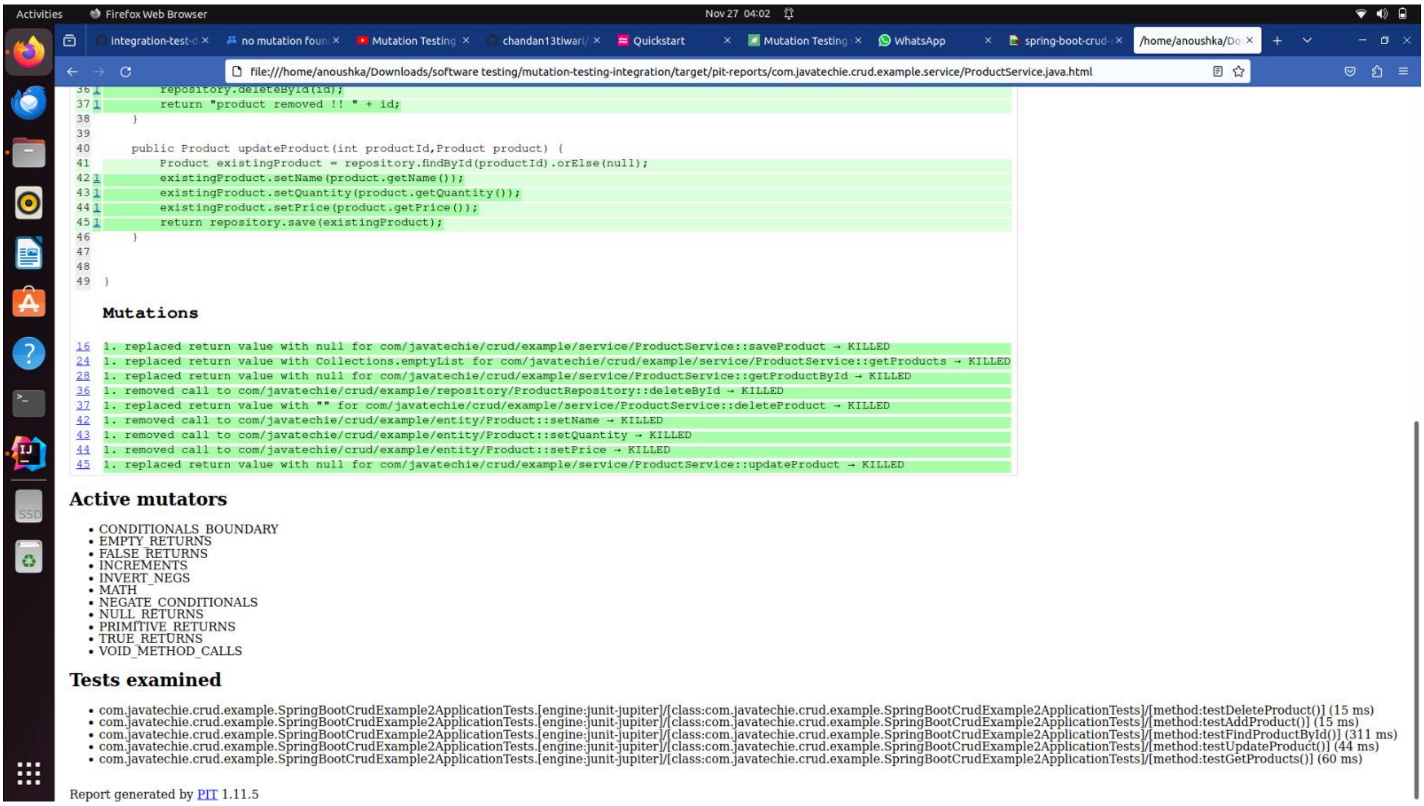
e. Pitest Report for the package



file:///home/anoushka/Downloads/software testing/mutation-testing-integration/target/pit-reports/com.javatechie.crud.example.service/ProductService.java.html

ProductService.java

```
package com.javatechie.crud.example.service;

import com.javatechie.crud.example.entity.Product;
import com.javatechie.crud.example.repository.ProductRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class ProductService {
    @Autowired
    private ProductRepository repository;

    public Product saveProduct(Product product) {
        return repository.save(product);
    }

//    public List<Product> saveProducts(List<Product> products) {
//        return repository.saveAll(products);
//    }

    public List<Product> getProducts() {
        return repository.findAll();
    }

    public Product getProductById(int id) {
        return repository.findById(id).orElse(null);
    }

//    public Product getProductByName(String name) {
//        return repository.findByName(name);
//    }

    public String deleteProduct(int id) {
        repository.deleteById(id);
        return "product removed !! " + id;
    }

    public Product updateProduct(int productId,Product product) {
        Product existingProduct = repository.findById(productId).orElse(null);
        existingProduct.setName(product.getName());
        existingProduct.setQuantity(product.getQuantity());
        existingProduct.setPrice(product.getPrice());
```



```
36 1         repository.deleteById(id);
37 1         return "product removed !! " + id;
38       }
39
40       public Product updateProduct(int productId,Product product) {
41           Product existingProduct = repository.findById(productId).orElse(null);
42 1         existingProduct.setName(product.getName());
43 1         existingProduct.setQuantity(product.getQuantity());
44 1         existingProduct.setPrice(product.getPrice());
45 1         return repository.save(existingProduct);
46       }
47
48
49   }
```

Mutations

```
16 1. replaced return value with null for com/javatechie/crud/example/service/ProductService::saveProduct → KILLED
24 1. replaced return value with Collections.emptyList for com/javatechie/crud/example/service/ProductService::getProducts → KILLED
28 1. replaced return value with null for com/javatechie/crud/example/service/ProductService::getProductById → KILLED
36 1. removed call to com/javatechie/crud/example/repository/ProductRepository::deleteById → KILLED
37 1. replaced return value with "" for com/javatechie/crud/example/service/ProductService::deleteProduct → KILLED
42 1. removed call to com/javatechie/crud/example/entity/Product::setName → KILLED
43 1. removed call to com/javatechie/crud/example/entity/Product::setQuantity → KILLED
44 1. removed call to com/javatechie/crud/example/entity/Product::setPrice → KILLED
45 1. replaced return value with null for com/javatechie/crud/example/service/ProductService::updateProduct → KILLED
```

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests.[engine:junit-jupiter]/[class:com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests]/[method:testDeleteProduct()] (15 ms)
- com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests.[engine:junit-jupiter]/[class:com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests]/[method:testAddProduct()] (15 ms)
- com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests.[engine:junit-jupiter]/[class:com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests]/[method:testFindProductById()] (311 ms)
- com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests.[engine:junit-jupiter]/[class:com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests]/[method:testUpdateProduct()] (44 ms)
- com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests.[engine:junit-jupiter]/[class:com.javatechie.crud.example.SpringBootCrudExample2ApplicationTests]/[method:testGetProducts()] (60 ms)

Report generated by PIT 1.11.5

REFERENCES

1. Mutation Testing Theory:
   a. https://www.geeksforgeeks.org/software-testing-mutation-testing/
2. Mutation Testing Tutorial:
   a. https://youtu.be/wZeZMtqVmck
   b. https://youtu.be/lDeTsMIN8As
3. PITclipse:
   a. https://github.com/pitest/pitclipse
   b. https://medium.com/meco-engineering/simple-mutation-testing-with-pit-f9ffbcf16cbe#:~:text=PIT%20is%20a%20Java%20mutation,providing%20detailed%20information%20on%20mutants
4. JUnit Assert: https://junit.org/junit4/javadoc/4.13/org/junit/Assert.html
5. PITest: https://medium.com/geekculture/mutation-testing-for-maven-project-using-pitest-f9b8fef03a05