

Mutation Testing on UserPortal Management System in Java

-

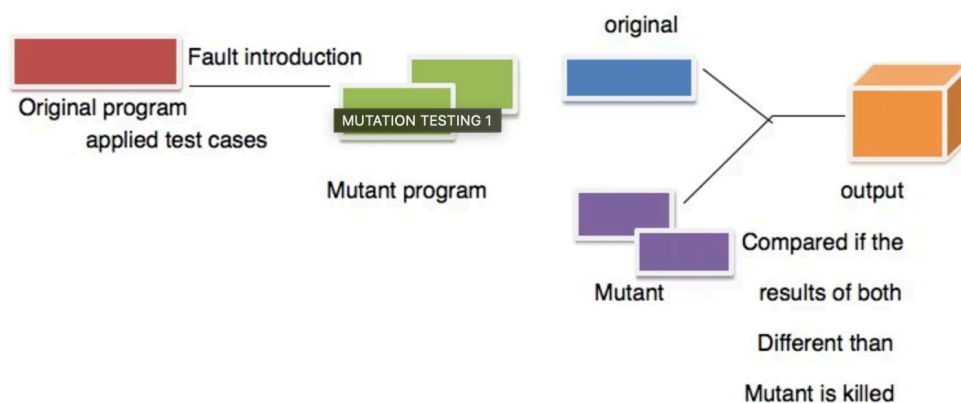
Aparajita Kumari-MT2023037

GitHub Repository - [UserPortal-Testing](#)

Introduction to Mutation Testing

A mutation is a small change in a program. Such small changes are intended to model low level defects that arise in the process of coding systems.

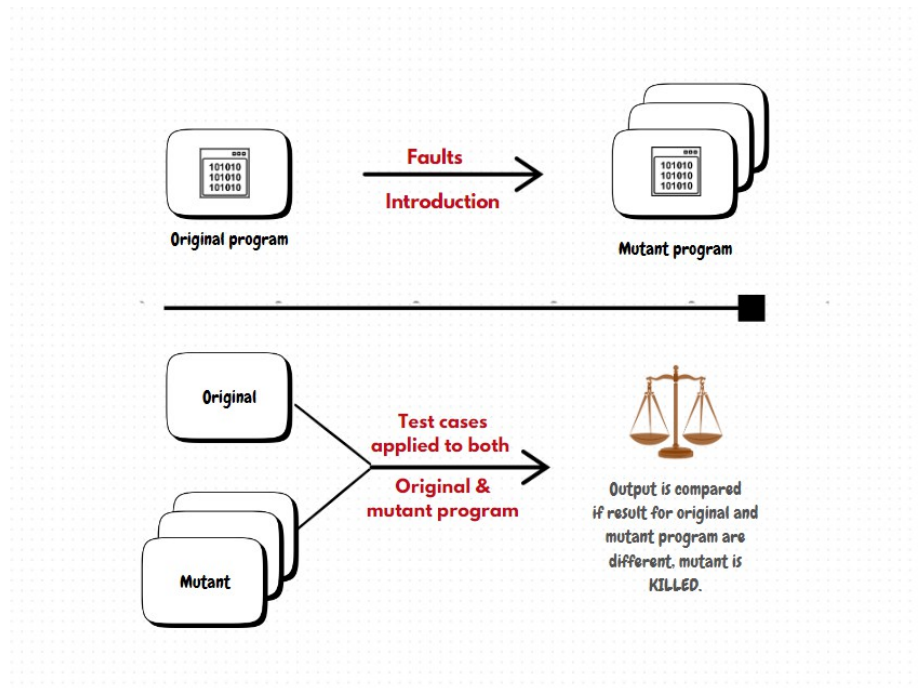
Mutation testing is a structural testing method aimed at assessing/improving the adequacy of test suites, and estimating the number of faults present in systems under test.



How mutation testing works:

Mutation testing basically involves introducing minor errors, called **mutants**, into the program code to check if the tests can detect these changes.

The ultimate goal is for robust tests to fail when they find these intentional changes. If the tests are unable to detect the mutants, it suggests they are ineffective at identifying problems.



Source - <https://www.stackspot.com/en/blog/mutation-testing>

The mutation testing process involves the following steps:

- **Introducing faults:** Changes are made to the program's source code to create mutants, such as replacing a plus sign with a minus sign in a mathematical expression.
- **Test application:** Test cases are applied to both the original code and the mutated program. If test results for the original code and the mutants differ, the mutant has been detected and eliminated, demonstrating the test's effectiveness.
- **Analysis of results:** If the tests fail to detect the change and the results for the original code and the mutant are the same, the tests lack sensitivity to identify errors, indicating a need to improve the test cases.

Tools Used:

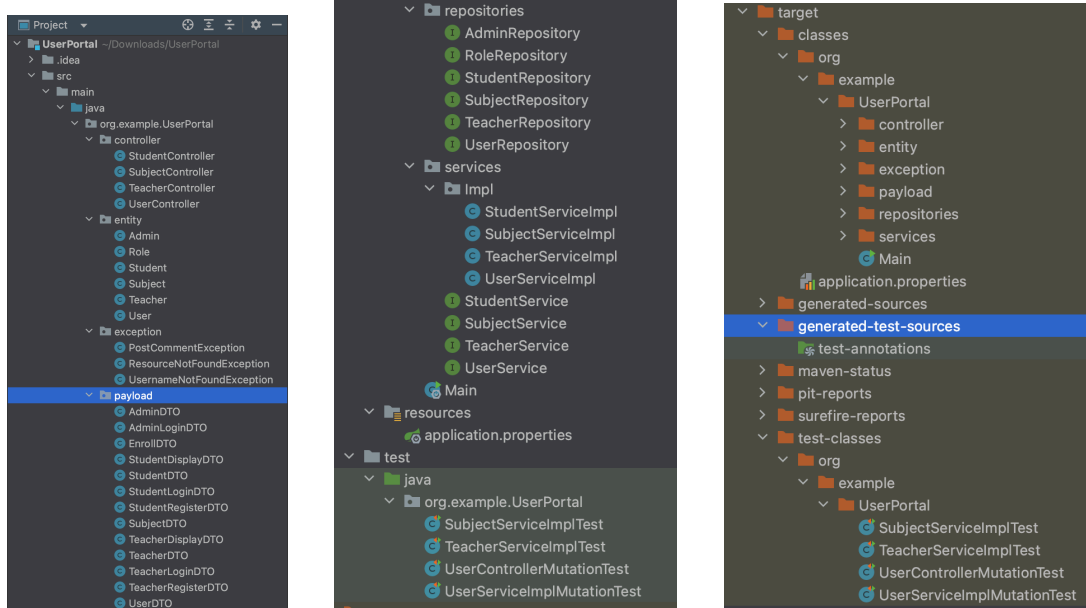
- **PITest** - PIT is a state of the art mutation testing system, providing gold standard test coverage for Java and the jvm. It's fast, scalable and integrates with modern test and build tooling.

PIT is -

- fast - can analyse in minutes what would take earlier systems days
- easy to use - works with ant, maven, gradle and others
- actively developed
- actively supported

- **JUnit** - It is a popular testing framework for Java. It allows developers to write and run unit tests to check if individual parts (like methods or classes) of their code work as expected. In simple terms, JUnit helps in testing code to catch bugs early by automating the testing process.

Project Structure:



The project directory follows a standard Maven structure.

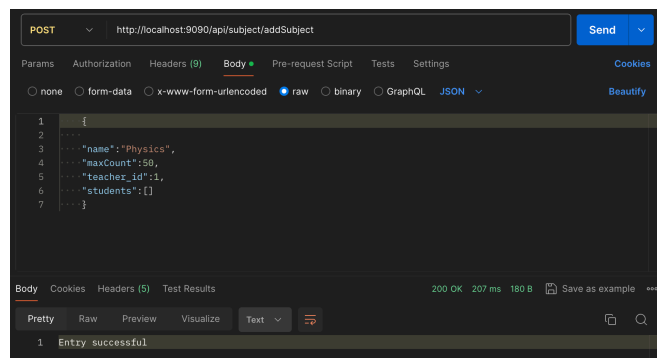
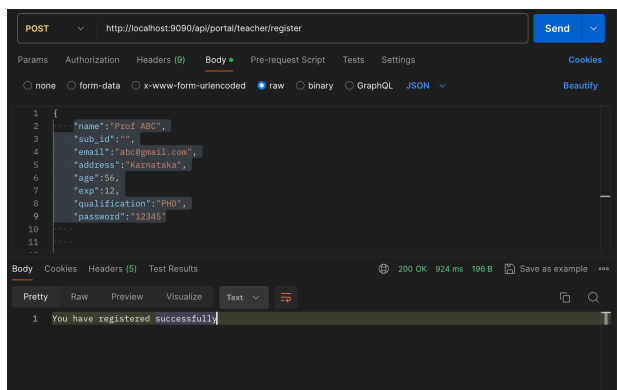
The src/main/java folder contains the application's main Java classes, The test/java folder contains tests for the corresponding classes in the

User Portal Management System:

The **User Portal** is a Spring Boot application designed to manage users such as students, teachers, and admins. It features secure role-based registration and login, enabling efficient user management. The application uses JPA for database interactions, MySQL for storage, and ModelMapper for DTO conversions. Testing is supported by JUnit and Mockito. With distinct functionalities for each role, the portal simplifies operations in educational or organizational contexts, offering a scalable and secure solution for user management.

Key Features:

1. **Registration:**
 - Users can register with role-specific attributes, such as qualifications for teachers and class levels for students.
2. **Authentication:**
 - Role-based login ensures secure access to resources.
3. **User Management:**
 - Seamless handling of user data using **Spring Data JPA** and a MySQL database.
4. **Testing and Quality Assurance:**
 - Includes **JUnit** and **Mockito** for unit and integration testing.
 - Integrated **Mutation Testing** using PIT to improve code quality.



Pit -mutation -plugin

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <excludes>
          <exclude>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
          </exclude>
        </excludes>
      </configuration>
    </plugin>
    <plugin>

      <groupId>org.pitest</groupId>
      <artifactId>pitest-maven</artifactId>
      <version>1.14.0</version> <!-- Adjusted version -->
      <dependencies>
        <dependency>
          <groupId>org.junit.jupiter</groupId>
          <artifactId>junit-jupiter</artifactId>
          <version>5.10.1</version>

        </dependency>
        <dependency>
          <groupId>org.pitest</groupId>
          <artifactId>pitest-junit5-plugin</artifactId>
          <version>1.2.1</version>
        </dependency>
        <dependency>
          <groupId>org.mockito</groupId>
          <artifactId>mockito-junit-jupiter</artifactId>
          <version>4.11.0</version>

        </dependency>

      </dependencies>
      <configuration>
        <targetClasses>org.example.*</targetClasses> <!-- Update as per your package -->
        <targetTests>org.example.*Test</targetTests> <!-- Update as per your tests -->
        <mutators>
          <mutator>DEFAULTS</mutator>

        </mutators>
        <outputFormats>
          <outputFormat>HTML</outputFormat>
        </outputFormats>
        <threads>4</threads>
        <verbose>true</verbose>
      </configuration>

    </plugin>
  </plugins>
</build>
```

Dependencies

```
<dependencies>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <optional>true</optional>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>org.modelmapper</groupId>
    <artifactId>modelmapper</artifactId>
    <version>3.2.0</version>
  </dependency>

  <!-- JUnit Jupiter Engine for running tests -->

  <!-- Mockito Core for creating mocks -->
</dependencies>
```

JUnit Test Results Summary:

```
mvn test
```

```
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.917 s -- in org.example.UserPortal.UserControllerMutationTest
[INFO] Running org.example.UserPortal.SubjectServiceImplTest
[INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.144 s -- in org.example.UserPortal.SubjectServiceImplTest
[INFO] Running org.example.UserPortal.TeacherServiceImplTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.008 s -- in org.example.UserPortal.TeacherServiceImplTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 15, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
```

Run the PIT Command -

```
mvn org.pitest:pitest-maven:mutationCoverage
```

```
>> Mutations with no coverage 209. Test strength 68%
>> Ran 50 tests (0.2 tests per mutation)
Enhanced functionality available at https://www.arcmutate.com/

!! The following issues were detected during the run !!
Project uses Spring, but the Arcmutate Spring plugin is not present (https://docs.arcmutate.com/docs/spring.html)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 13.021 s
[INFO] Finished at: 2024-11-29T20:43:47+05:30
[INFO] -----
```

After the execution, the mutation report will be generated in the target/pit-reports directory.

Open the index.html file in the browser to view the detailed PIT mutation coverage report.

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
31	29% <div><div></div></div> 129/448	11% <div><div></div></div> 26/247	68% <div><div></div></div> 26/38

Breakdown by Package

Name	Number of Classes	Line Coverage	Mutation Coverage	Test Strength
org.example.UserPortal	1	50% <div><div></div></div> 2/4	0% <div><div></div></div> 0/1	0% <div><div></div></div> 0/1
org.example.UserPortal.controller	4	38% <div><div></div></div> 15/39	14% <div><div></div></div> 3/21	100% <div><div></div></div> 3/3
org.example.UserPortal.entity	6	27% <div><div></div></div> 16/60	10% <div><div></div></div> 4/42	57% <div><div></div></div> 4/7
org.example.UserPortal.exception	3	38% <div><div></div></div> 10/26	0% <div><div></div></div> 0/8	0% <div><div></div></div> 0/0
org.example.UserPortal.payload	13	28% <div><div></div></div> 39/141	9% <div><div></div></div> 6/68	86% <div><div></div></div> 6/7
org.example.UserPortal.services.Impl	4	26% <div><div></div></div> 47/178	12% <div><div></div></div> 13/107	65% <div><div></div></div> 13/20

Report generated by PIT 1.14.0

UserController.java

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NULL_RETURNS
- PRIMITIVE_RETURNS
- REMOVE_CONDITIONALS_EQUAL_ELSE
- REMOVE_CONDITIONALS_ORDER_ELSE
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- org.example.UserPortal.UserControllerMutationTest.[engine:junit-jupiter]/[class:org.example.UserPortal.UserControllerMutationTest]/[method:testRegisterStudent()] (6 ms)
- org.example.UserPortal.UserControllerMutationTest.[engine:junit-jupiter]/[class:org.example.UserPortal.UserControllerMutationTest]/[method:testRegisterTeacher()] (74 ms)
- org.example.UserPortal.UserControllerMutationTest.[engine:junit-jupiter]/[class:org.example.UserPortal.UserControllerMutationTest]/[method:testLoginAdmin()] (4 ms)

Mutations -

```
52 }
```

Mutations

```
21 1. replaced return value with "" for org/example/UserPortal/controller/UserController::registerStudent → KILLED
27 1. replaced return value with "" for org/example/UserPortal/controller/UserController::registerTeacher → KILLED
33 1. replaced return value with "" for org/example/UserPortal/controller/UserController::loginTeacher → NO_COVERAGE
39 1. replaced return value with "" for org/example/UserPortal/controller/UserController::loginStudent → NO_COVERAGE
44 1. replaced return value with "" for org/example/UserPortal/controller/UserController::loginAdmin → KILLED
49 1. replaced return value with "" for org/example/UserPortal/controller/UserController::registerAdmin → NO_COVERAGE
```

References -

5. https://www.inf.ed.ac.uk/teaching/courses/st/2011-12/Resource-folder/09_mutatio_n.pdf
6. <https://www.stackspot.com/en/blog/mutation-testing>
7. <https://pitest.org/quickstart/mutators/>
8. <https://www.youtube.com/watch?v=DSv2vpvD-ds&t=2195s>
9. <https://www.youtube.com/watch?v=hVKDEXKLN2c&t=2713s>
10. <https://softengbook.org/articles/mutation-testing>