CS731 SOFTWARE TESTING

PROJECT REPORT

Tanishka Singhal

MT2022123

Himanshu Chodhary

MT2022148

MUTATION TESTING REPORT

1. Problem Description:

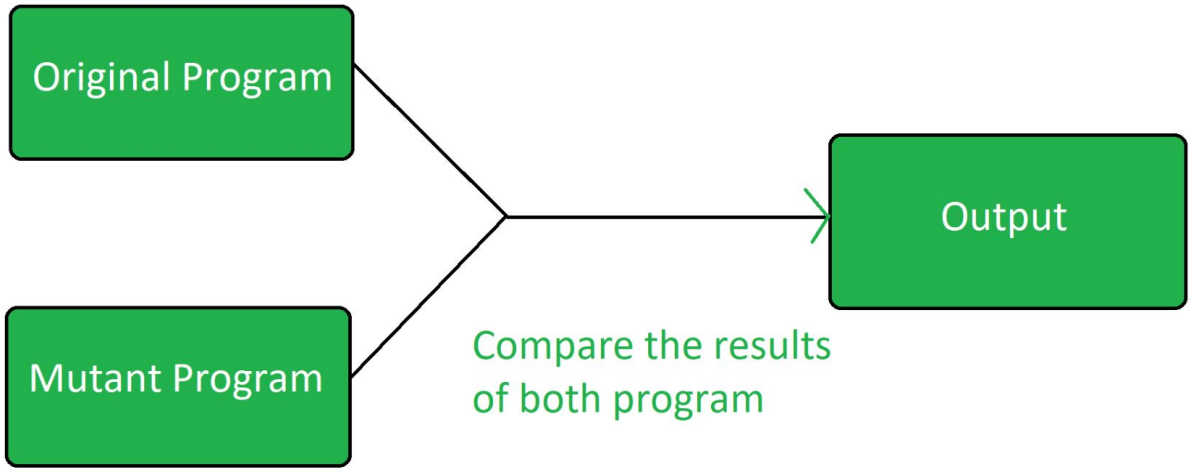
Mutation Testing for Source Code: Many projects adopt mutation testing, which involves applying mutation operators at the statement level within a method or function. The objective is to ensure that the mutated program is effectively identified as faulty by the designed test cases. The requirement includes using a minimum of three distinct mutation operators.

2. Introduction:

Mutation Testing:

Mutation testing is a technique used in software testing to evaluate the effectiveness of a test suite. The main idea behind mutation testing is to introduce small, intentional changes (mutations) into the source code and then check whether the existing test suite can

detect these changes. The goal is to ensure that the test suite is robust enough to identify potential bugs or changes in the code.



Mutant:

A mutant is a small, deliberate modification made to the original source code. Mutations can include changes like modifying operators, swapping statements, or introducing conditional changes.

Mutant Set:

A mutant set refers to a collection of mutants created by applying various mutation operators to the source code. Each mutant in the set represents a potential defect in the code.

Killing a Mutant:

If a test case detects a mutant, it is said to have "killed" the mutant. Killing a mutant means that the test case has identified the intentional change introduced by the mutation.

Mutation Score:

The mutation score is a metric that represents the percentage of mutants killed by the test suite. A higher mutation score indicates a more effective test suite.

Mutation Operators:

These are specific rules or techniques used to create mutants. Examples include changing arithmetic operators, swapping true/false conditions, or replacing variables with constants.

3. Tool:

We have employed PITest, a powerful mutation testing tool, to enhance the robustness of our test suites. By utilizing PITest, we aim to rigorously evaluate the effectiveness of our tests through the generation and assessment of mutated versions of our source code, ensuring comprehensive test coverage and identifying potential weaknesses in our testing strategies.

3.1 PITest

PiTest, short for "Pit Mutation Testing," is an open-source mutation testing tool for Java applications. It is designed to assess the quality of your test suite by introducing small, intentional changes (mutations) into the source code and determining whether the tests can detect these changes. PiTest helps you identify areas where your tests may be lacking in coverage or where the codebase is not adequately protected.

4. Source Code:

Source Code :
https://github.com/tanishkasinghal/Software-Testing

The EMPLOYEE MANAGEMENT SYSTEM has been developed to override the problems prevailing in the practising manual system. The system is designed for the particular need of the company to carry out operations in a smooth manner. This project will allow the admin to add new employees. Admin can also add new departments. While registering an employee it can assign them to different departments. ALong with employee database

management we have added leave management functionalities too.

Backend API's and EndPoints:

3.1 Department Service:

Department API's:

->/department/

Method: POST

Description: This API add department.

->/department/{id}

Method: GET

Description: This API gets a department object of a particular id.

->/department/

Method: GET

Description: This API gets all the departments list

3.2 Employee Service:

Employee API's:

->/employee/register

Method: POST

Description: This API adds an employee.

->/employee/

Method: GET

Description: This API gets all employees list

->/employee/{id}

Method: PUT

Description: This API updates the employee object.

->/employee/{id}

Method: DELETE

Description: This API deletes the employee with a particular id.

->/employee/{id}

Method: GET

Description: This API gets the employee object of a particular employee id

3.3 Leave Service:

Leave API's:

->/leave/

Method: POST

Description: This API adds a leave application.

->/leave/pending

Method: GET

Description: This API gets all the pending leave applications.

->/leave/{employeeId}

Method: GET

Description: This API gets all the leave applications of a particular employee.

->/leave/submitResponse

Method: PUT

Description: This API will update the leave application record with pending or rejected status with admin's remarks if any.

Apart from this we have implemented few services for calculating Base Salary, tex according to salary bracket, total salary.

Test Coverage:

This is the overall PIT test Coverage. Detailed report can be found in the target folder with pit-reports, which we have attached.

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
12	90%	197/218	75%	117/155	83%	117/141

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
com.speMajor.demo.controller	5	85%	63/74	71%	22/31	85%	22/26
com.speMajor.demo.controller.blackbox	1	100%	3/3	100%	1/1	100%	1/1
com.speMajor.demo.service.BlackBox	1	100%	3/3	100%	1/1	100%	1/1
com.speMajor.demo.service.Department	1	100%	20/20	67%	8/12	73%	8/11
com.speMajor.demo.service.Employee	1	87%	45/52	59%	19/32	70%	19/27
com.speMajor.demo.service.Leave	1	100%	19/19	75%	9/12	75%	9/12
com.speMajor.demo.service.Other	1	91%	29/32	85%	34/40	92%	34/37
com.speMajor.demo.service.calculator	1	100%	15/15	88%	23/26	88%	23/26

Report generated by PIT 1.9.11

Enhanced functionality available at arcmutate.com

7. Execution:

7.1 Pre-Setup Required:

• Make sure to have Java JDK 11 successfully setup.

• Also make sure to have maven installed with proper PATH variables Initialised.

7.2 How to Run:

Run the following command:

mvn install

mvn test −compile org . pitest:pitest −maven : mutationCoverage

REFERENCES:

1. **Mutation Testing Resources:**
   a. https://www.javatpoint.com/mutation-testing

   b. https://www.geeksforgeeks.org/software-testing-mutation-testing/

2. **Mutation Testing YouTube Tutorials:**
   a. https://www.youtube.com/watch?v=wZeZMtqVmck
   b. https://www.youtube.com/watch?v=AlOh1zcZmh8
   c. https://www.youtube.com/watch?v=Rom0n-lkT-c

3. ***PITest mutation coverage documentation* at:**

   https://www.baeldung.com/java-mutation-testing-with-pitest