**Software Testing Project Report (CS731)**



**Topic: Mutation Testing for Full Stack Web Development**

**GitHub Repository Link**: [**https://github.com/yashadayal/ClarityPlusPackageTesting**](https://github.com/yashadayal/ClarityPlusPackageTesting)

**Group Members: Yasha Dayal –MT2022137**

**Anisha Rani –MT2022153**

**Under guidance of: Prof. Meenakshi D'Souza**

**[11/27/2023]**

## **1. Introduction**

### **1.1 Background:**

Web applications are pivotal components in contemporary software ecosystems, necessitating robust testing methodologies. This report delves into the synergy of mutation testing with unit and integrated testing.

**1.2 Problem Statement:**

The report aims to elucidate the integration of mutation testing, unit testing, and integrated testing, particularly using the PIT tool. It provides insights into how this enhances the reliability and robustness of web applications.

## **2. Understanding Mutation Testing**

**2.1 Core Concept:**

Mutation testing involves the creation of mutants through controlled changes to the source code, assessing how well test suites detect and report these mutations. The report emphasizes the integration of mutation testing with unit and integrated testing for a comprehensive evaluation using PIT Test tool.

**2.2 Workflow:**

* **Mutant Generation:** Small changes (mutations) are introduced to the original code autogenerated by PIT Tool.
* **Unit Testing:** The PIT tool facilitates unit testing against both the original code and mutants.
* **Integrated Testing:** The integration of mutants into the overall application for a more holistic assessment.
* **Mutation Score:** Calculating the mutation score based on the effectiveness of the test suite against mutants.

**3. Web App Functionality**

**3.1 Application Description:**

The "**Clarity Plus Package**" application, built on a micro-services architecture, serves as the epitome of efficient package handling and management within a college campus.

**3.2 Major App Services:**

**Order Service**: This micro-service takes charge of managing and processing orders, playing a crucial role in the package handling workflow. It facilitates the organization and handling of orders by the on-duty guard, ensuring a streamlined process from order receipt to delivery.

**Recipient Service**: The Recipient Service is dedicated to managing recipient information and their corresponding order details. It serves as a repository of recipient data and acts as a bridge between the order processing and recipient interaction processes.

**3.3 Roles and Responsibilities:**

**Admin:** Registration of recipients and guard personnel. Admins collect crucial data, including email addresses and passwords, to serve as credentials for portal access.

**Recipient:** After logging in, recipients are required to complete a form for their pending orders, providing essential details to ensure efficient handling upon delivery.

**Guard:** Upon logging in, guards can add order, hand Over Package to Recipient, Search Functionality.

**4.Testing**

**4.1 Tools Used for Testing:**

Generation of test cases and Mutators for both unit and integration testing are done using the following tools and plugins:

* Sapient-Ai
* Diffblue
* JUNIT
* JEST

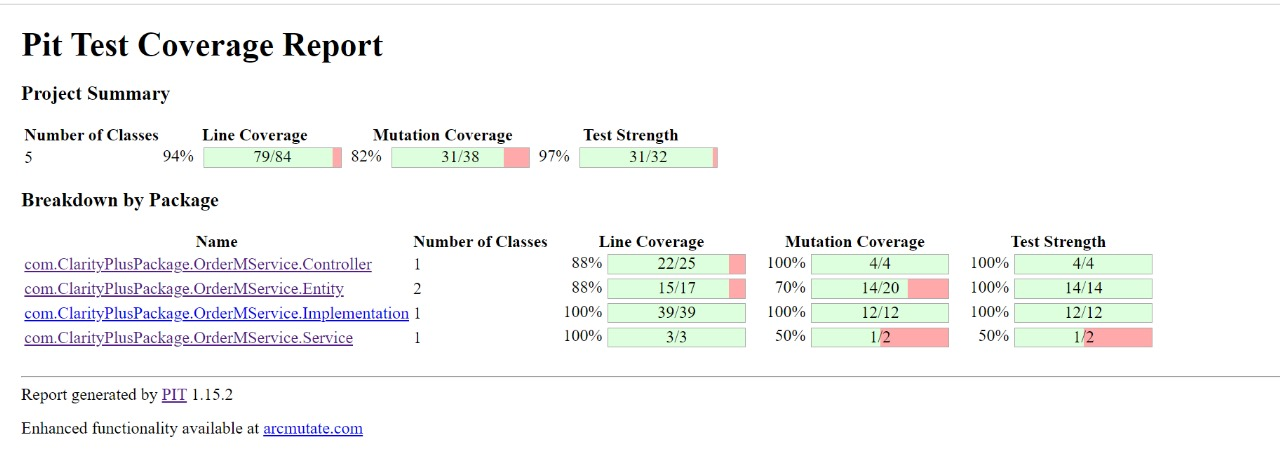
**4.2 Mutation operators applied:**

We have applied the following Mutation operators for testing the Web Application:

* CONDITIONALS\_BOUNDARY
* EMPTY\_RETURNS
* FALSE\_RETURNS
* INCREMENTS
* INVERT\_NEGS
* MATH
* NEGATE\_CONDITIONALS
* NULL\_RETURNS
* PRIMITIVE\_RETURNS
* TRUE\_RETURNS
* VOID\_METHOD\_CALLS

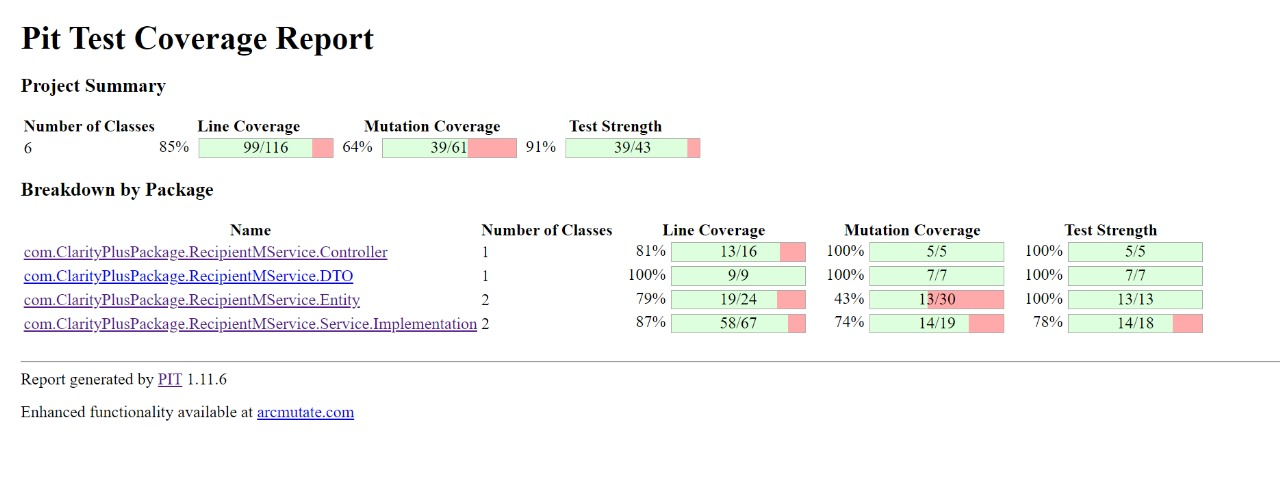
The above mutation operators are applied by PIT Runner by default.

**4.3 PIT Testing Reports:**



**Img 4.1**

The image above depicts the report generated by PIT TEST for the microservice-OrderMservice-which takes care of the roles for order services.

  **Img 4.2**

The image above depicts the report generated by PIT TEST for the microservice-RecipientMService -which takes care of the roles for Recipient services.

## **4.4 Conclusion:**

The amalgamation of mutation testing, unit testing, and integrated testing using the PIT tool offers a comprehensive testing approach for web applications. As the web development landscape evolves, adopting this integrated testing strategy can significantly contribute to building resilient and reliable web applications.

**5. References:**

1. Official Website of testing tool: https://pitest.org

2. Youtube Channel – Tech Primers and CodeFarm