Group Members:
Akshay Dake (MT2017011)
AmarPrakash Mishra (MT2017015)

Problem Statement:

Mutation-source-code: Projects that use mutation testing, based on mutation operators applied at the level of a statement within a method or a function. The mutated program needs to be strongly killed by the de- signed test cases. At least three different mutation operators should be used.

What is Mutation Testing?

Mutation testing is a structural testing technique, which uses the structure of the code to guide the testing process. On a very high level, it is the process of rewriting the source code in small ways in order to remove the redundancies in the source code.

Mutation Testing Types:

- **Value Mutations:** An attempt to change the values to detect errors in the programs. We usually change one value to a much larger value or one value to a much smaller value. The most common strategy is to change the constants.
- **Decision Mutations:** The decisions/conditions are changed to check for the design errors. Typically, one changes the arithmetic operators to locate the defects and also we can consider mutating all relational operators and logical operators (AND, OR , NOT)
- **Statement Mutations:** Changes done to the statements by deleting or duplicating the line which might arise when a developer is copy pasting the code from somewhere else.

About Project and Testing Strategy:

This project is about Testing Ticket Application.It contains various tickets of possible travel. I.e Train,Bus,Taxi,Ship and for the extension we also used these methods to do same for Movie,Hotel.

The strategy for test cases is that we consider all possible combination of tickets that can be form i.e Consider example for Bus ticket in bus ticket there are three possible type of ticket like

1.Ticket for child

2.Ticket for Adult

3.Ticket for Senior Citizen

So the test cases are for type of tickets

1.calculatePriceForChild

- Which basically pass only one adult persons age

2.calculatePriceForFamily

- For family it contains two adult and two child passengers

3.calculatePriceForChildNarrowCase

- Child Narrow case is like a person who’s age is exactly18 years.

4.calculatePriceForFreeTicketNarrowCase

- For free ticket narrow case is like person who’s age is exactly 3 years

5. shouldNotCalculatePriceForFamilyEdgeCaseWithAdults

- In this case we have 2 adult passengers and 1 Child

6. shouldNotCalculatePriceForFamilyEdgeCaseWithChildren

- Here we have 2 child passengers and 1 adult passenger

7.calculatePriceForOneAdult

- Here we have only 1 adult passenger.

So strategy used for other type of tickets like tickets for AirPlane,Taxi,Ship,Train and Similar can be used for Movie ,Hotels and Parks

Tools and Plugins used

1.PIT

2.JUnit

1.PIT

PIT runs your unit tests against automatically modified versions of your application code. When the application code changes, it should produce different results and cause the unit tests to fail. If a unit test does not fail in this situation, it may indicate an issue with the test suite.

2. JUnit

JUnit is a **Regression Testing Framework** used by developers to implement unit testing in Java, and accelerate programming speed and increase the quality of code. JUnit Framework can be easily integrated with either of the following

Contribution:

Code and Test Case logic : Akshay Dake (MT2017011)

Documentation :AmarPrakash Mishra (MT2017015)