**A Project of AUTOMATED TESTING TOOLS**

**Under the guidance of Professor Navpreeth Kaur**



**JUNIT TESTCASES AND ITS REPORT**

**VEMPATI SAI:1995877**

**NIKITHA YANDRAPATI:1994194**

* **Introduction**

JUnit is one of those unit frameworks which were initially used by many Java applications as a Unit test framework. By default, JUnit tests generate simple report XML files for its test execution. These XML files can then be used to generate any custom reports as per the testing requirement. We can also generate HTML reports using the XML files. Ant has such a utility task, which takes these JUnit XML files as input and generates an HTML report.

One could begin the report with an introductory note describing the product that was tested and the purpose of the report. It could also include highlights of some of the important functions and features that were tested.

* Importance – Gives the reader an idea about the product beingtested.

How to define a test in JUnit?

A JUnit test is a method contained in a class which is only used for testing. This is called a Test class. To define that a certain method is a test method, annotate it with the cases.

This method executes the code under test. You use an assert method, provided by JUnit or another assert framework, to check an expected result versus the actual result. These calls are typically called asserts or assert statements.

Assert statements typically allow to define messages which are shown if the test fails. You should provide here meaningful messages to make it easier for the user to identify and fix the problem. This is especially true if someone looks at the problem, who did not write the code under test or the test code.

**5.2. Example JUnit 5 test**

A product could be tested for various aspects some of which are listed below. It makes a lot of difference to handle each of these aspects while reporting the results. Each of them has its own impact on the state of the application. Strengths and Limitations of a product could be better presented to the customer only if they are handled individually.

* **AcceptanceCriteria**

Testing is never complete if acceptance criteria are not defined. This should be an integral part of any report.

* Importance – These are deciding factors for test completion andconclusion

**3.4 TestStrategy**

It is essential to describe the strategy adopted for testing. This could also highlight the basis of testing.

i.e. the input documents that the testing team received based on which the test plan preparation and the testing itself was conducted.

* Importance – Gives the reader an idea about the adopted strategy/methodology for testing.This
* could be used for verifying if the adopted strategy/methodology for its correctness and completeness.
* **TestSummary**

Have a section to address at a high level the test details like the test start/end dates, the tool used for defect reporting, the number of builds released for testing, the number of bugs reported and importantly the **certification** part of the testing. This section could be used to present the conclusion about the readiness of the product.

* Importance – Gives the reader an idea at a very high level the efficiency of the developmentand also about the state of the product.
* **Qualitative grading**

This section should follow immediately after the test summary as this again presents the reader at a high level a qualitative picture of various parameters. The table below could list all the parameters related to a software so that the reader is able to get at a glance an idea about those aspects which are working as expected and those which are have issues.

* **Analysis**

Provide at a high level, some analysis of the kind of defects found during testing. Describe the basis on which the conclusion of readiness/non readiness was arrived at.

* Importance – shows transparency in analysis and methodology followed for concluding the readiness.
* **Strengths, Limitations,Recommendations**
* JUnit is a Java unit testing framework that's one of the best test methods for regression testing. An open-source framework, it is used to write and run repeatable automated tests. As with anything else, the JUnit testing framework has evolved over time
* Thiswoulddefinitelybeavalueadditiontothecustomerfromthepointofmarketingthe product.Dependingonthecriticalityofthepoints,thecustomercouldevendecidetodefer thereleaseandworkonthelimitationsandrecommendationsandreleaseabetterproduct.
* Unsaidbenefitsare,increasedcustomerconfidenceinthetestingteamandhenceincreased business for thevendor.
* Quantitative Analysis

In order to present the findings of the testing to the customer, the team has to anyway do a lot of analysis on the various data collected during the testing. So, wouldn’t it be better that they be presented in the report to

* Supportconclusions
* Reduce rework effort on the same kind of analysis that the development team may end updoing
* Implicitly suggest the areas/process that requireimprovement
* Some of the aspects that could be reported under various types of testing are given below.

About Junit

JUnit is an open source Unit Testing Framework for JAVA. ... As the name implies, it is

used for Unit Testing of a small chunk of code. Developers who are following test-

driven methodology must write and execute unit test first before any code. Once you

are done with code, you should execute all tests, and it should pass.

Uses:

JUnit is an open source framework, which is used for writing and running tests.

Provides annotations to identify test methods. Provides assertions for testing expected

results. Provides test runners for running tests.

Report:

In all the test cases of an application, there is always a way to create a report of it one way

or the other. JUnit is not an exception to this and there’s a ton of ways to create reports

from it. It can even be incorporated to the maven site and have it part of the overall java

tech documentations of a Java Application.

One of the most widely used JUnit reporting plugin is the surefire and in this post, I’ll be

showcasing how it can be used on your

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID | Module | Test Executed by | Tested Website | Action | Expected Result | Actual result | Status |
| #201 | BMI Functionality | Sai vempati | <https://www.calculator.net/bmi-calculator.html> | Enter weight, height and age | BMI value | As Expected, | GOOD |
| #202 | Find Broken Links | Sai vempati | <https://google.co.in/> | Visit all links to verify the are opening | All links in working condition | As Expected, | GOOD |
| #203 | Calendar Date Selection | Sai vempati | <http://jqueryui.com/resources/> demos/datepicker/other-months.html | Launch calendar and pick a date | Calendar displayed and date chosen | As Expected, | GOOD |
| #204 | Profile Name Check | Sai vempati | <https://www.facebook.com/> | Login with credentials and verify profile name | Dashboard page with profile | As Expected, | GOOD |
| #205 | Interest Calculator | Sai vempati | <http://www.calculator.net/interest-calculator.html> | Enter principal, num of years | Simple Interest and Compund interest | As Expected, | GOOD |
| #206 | Test Social Network Connections | Nikitha yandrapati | <https://www.linkedin.com/> mynetwork/invite-connect/connections | Login m, find connections and confirm a contact | List all and  Found Contact | As Expected, | GOOD |
| #207 | Mortgage Calculator | Nikitha yandrapati | <http://www.calculator.net/mortgage-payoff-calculator.html> | Enter number of years | EMI displayed | As Expected, | GOOD |
| #208 | Percent Calculator | Nikitha yandrapati | <http://www.calculator.net/> | Enter percent value and maxvalue | Percent Value | As Expected, | GOOD |
| #209 | Wait for title to load | Nikitha Yandrapati | <https://www.jntuk.edu.in/> | Ensure page title loads with time | Loads with acceptable time | As Expected, | GOOD |
| #210 | Net Salary Calculator | Nikitha yandrapati | <https://www.calculator.net/salary-calculator.html> | Enter basic salary | Annual Salary | As Expected, | GOOD |