**Team4**

**Iteration #3**

**SER 216**

**TestNG**

TestNG is a unit testing framework developed by Cédric Beust who built this tool in order to emulate and improve the JUnit unit testing framework. Since TestNG is an open source tool, it is free for users and can be obtained by downloading a plugin for a Java IDE (most Java IDEs, such as Eclipse and IntelliJ, support TestNG, or even come with it already installed). TestNG is a Java framework, and can be used by any language in JVM. Eclipse specifically, you can then see TestNG under the Java Views, the same category as JUnit.

TestNG’s main purpose is to provide a framework for unit testing just like JUnit, however, TestNG also covers other types of testing such as integration, acceptance and functional tests. It provides a wider range of functionality (than just unit tests) all in one package making it a useful tool for comprehensive testing of a program. Furthermore, TestNG also tries to improve the standard for unit tests in Java (set by JUnit) by making annotations more flexible and able to handle a wider range of cases. For example, instead of just having the standard before and after annotations like JUnit, it has before and after test, suite, class, method, and group. This allows developers to be able to run tests as groups since some tests may have different dependencies than others, which can be very useful for developers. TestNG also allows developers to write tests that accept parameters to be passed in, another feature not supported by JUnit. Other features of unit tests that TestNG supports are parallel test execution and more control over setting or modifying test fixtures.

The following is a list of TestNG strengths and benefits:

* Annotations
* Run tests in arbitrary big thread pools with various policies available (all methods in their own thread, one thread per test class, etc.)
* Test that code is multi-thread safe
* Flexible test configuration
* Data-driven testing
* Parameter testing
* Powerful execution mode
* Default HDK functions for runtime and logging
* Dependent methods for application server testing
* Essentially, covers all categories of tests

Additionally, you can also save/modify settings within a saved XML file. There are, of course, properties and settings for users to fine-tune their tests. Another nice addition to TestNG is that all JUnit tests can be easily converted to TestNG. The results for this test are similar to JUnit view with the addition of a statics tab. You can run this exactly like JUnit except the testing capabilities have been expanded (i.e. multi-threading, parameters, etc). TestNG has a similar interface to that of JUnit; there is almost no learning curve and you get all of the added benefits.

**Apache JMeter**

Apache JMeter software is developed by Apache Software Foundation, which is an American non-profit corporation and also a decentralized open source community of developers. This is a free software to its users, and it can be acquired from the following link: <http://jmeter.apache.org/download_jmeter.cgi>.

JMeter can be used on dynamic and static resources for load testing and measuring performance of several different services with an emphasis on web applications. It supports the ability to execute load and performance tests: HTTP, HTTPS, FTP, JDBC, LDAP, Message- oriented middleware via JMS, STMP, POP, IMAP, TCP, Native commands or shell scripts, and Java Objects. It can also be used to simulate a heavy load on server or a group of serves to test their strength or analyze overall performance under different load types. JMeter Software has the capability to support languages such as Java, JavaScript, PHP, ASP.NET(C#, Visual Basic)

It is said that you cannot start performance testing early enough when it comes to developing web applications. With JMeter, tests can be written and applied as soon as the application or java object is built. This allows for testing to begin in the development process and, most extensively, to be used during system testing and acceptance testing phases. The reports developed during performance testing can be used to demonstrate to clients that the web application can perform under specific levels of use.

In order to use the tool we need to follow the following procedures:

**Installing procedure**: In order to install this software, we need to download the apache-jmeter-3.2.zip file from the provided link. After it is downloaded, we need to unzip the file. Open the folder apache-jmeter-3.2 > bin > run “jmeter.bat” file. Software will run and ready to use.

**Configuration procedure**: To configure the software, we need to run the JMeter and go to Edit > Add > Threads(Users) > Thread Group.

**Using procedure:** By executing the file jmeter.bat. The GUI allows for easy creation of a test plan. The first step to creating a test is to create a thread group. Thread groups are the users that will be tested. The thread group can be modified at any time to add more users or gradually ramp up the number of user being tested. The second step is to create test cases. Test cases can be created for many different scenarios including testing an HTTP request, creating a JUnit request and many more. The final step to creating a test is creating listeners to generate reports on the results of the test. Once the test is created, it is then run and the results viewed.

We were able to install JMeter and write my first test in fifteen minutes. JMeter runs as a standalone application with no preparation required other than unzipping the file and running it. This ease of use and quick set up is a big positive aspect of the software. JMeter has a large range of test types and listeners, which allows for comprehensive performance testing of most projects. JMeter has most of the same features and tools of similar software packages that must be purchased where JMeter is free open source software. The ability to write performance tests and be able to manage the user thread separately from the tests and the listeners gives a lot of flexibility. It is easy to improve tests and customize them for different uses as the project moves through the testing phases. We would recommend JMeter for any projects that performance tested is needed.