# Test tools

Test tools are automated tools that help the activity of testing. Test tools provides the project higher productivity, quality, easier problem solving, more efficient testing and it expands the testing opportunities. With all these performance increasing perks, we made sure that we provide our product with enough useful resources.

In the following part of this subject, there will a be list of testing tools we use in for our application. The list will contain the testing tool and later a short description will be provided.

These are the current tools that are in use on the date of the document delivery:

|  |  |
| --- | --- |
| Test tool | Version |
| PHP Unit | 5.1 |
| Google Test | 1.7 |
| Firebug | 2.0.7 |
| Selenium | [2.49.1](http://goo.gl/rQhaxb) |

## PHP Unit & Google Test

**PHP Unit** is a unit testing framework for PHP. It is an instance of the **xUnit** architecture for unit testing frameworks.

**Google Test** is a unit testing library for the C++ programming language, based on the **xUnit** architecture.

With the use of Google Test we will fire unit tests on the C++ code written for the process.

Both of the unit testing frameworks are based on the **xUnit** architecture, which will result in better test cases for the code.

## Firebug

Firebug is an testing tool plugin for Mozilla Firefox. Its primarily used for testing web applications. You can inspect a webpage and re-execute Javascript files, change the css document on the fly, cookies tracking with many more features as well. Recently they came out with a lite version for other web browser but we made the test for this project in Mozilla Firefox for optimal compatibility, more features and better performance.

## Selenium

Selenium is a testing tool that automates browsers. It provides a record/playback tool. We will use this tool to test our front-end. With these tests we can make sure that the application works as intended with specific input from the user. These tests will be done based on test cases that are written by a developer. The test cases will mostly cover the parts where the applications’ components interact with each other and could fail to operate.

# Test cases

The test cases used in this document are written by the developers of the project before they are tested. Some of the tests will be done by end users who weren’t a part of the development and the others will either be automated or done by a developer manually.

Below is the test case template that will be used in this document:

# Unit tests

Unit tests are used to test hard and potential risky pieces of code. The goal of using unit tests is testing independent pieces of code based on input and output. The tests are based on test cases that are written beforehand by the developer.

|  |  |
| --- | --- |
| Uni-test | Description |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Unit test 1

#### Testcase

#### Result

# UAT

|  |  |
| --- | --- |
| Requirements | Acceptatie test |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Test rapports