



<b>Project Name:</b>	Moodle LMS
<b>Product Name:</b>	Moodle Testing Framework
<b>Product Release Version:</b>	Version 3.7.2

Document Version 4.0

Created by: Alex Laughlin, Andrew, and Chandler Long

# INTRODUCTION

---

## ***The Goal:***

The testing framework will run on Ubuntu (Linux/Unix). The testing framework will be invoked by a single script from within the top level folder using “./scripts/runAllTests.sh” and will access a folder of test case specifications, which will contain a single test case specification file for each test case. Each of these files will conform to a test case specification template that you develop based on the example template below. Each test case specification file thus contains the meta-data that your framework needs to setup and execute the test case and to collect the results of the test case execution.

## ***The Project:***

Moodle is a free and open-source learning management system (LMS) written in PHP and distributed under the GNU General Public License. Developed on pedagogical principles, Moodle is used for blended learning, distance education, flipped classroom and other e-learning projects in schools, universities, workplaces and other sectors. Moodle was originally developed by Martin Dougiamas to help educators create online courses with a focus on interaction and collaborative construction of content, and it is in continual evolution. The first version of Moodle was released on 20 August 2002. Nowadays the Moodle Project is led and coordinated by Moodle HQ, an Australian company of 50 developers which is financially supported by a network of eighty-four Moodle Partner service companies worldwide. Moodle's

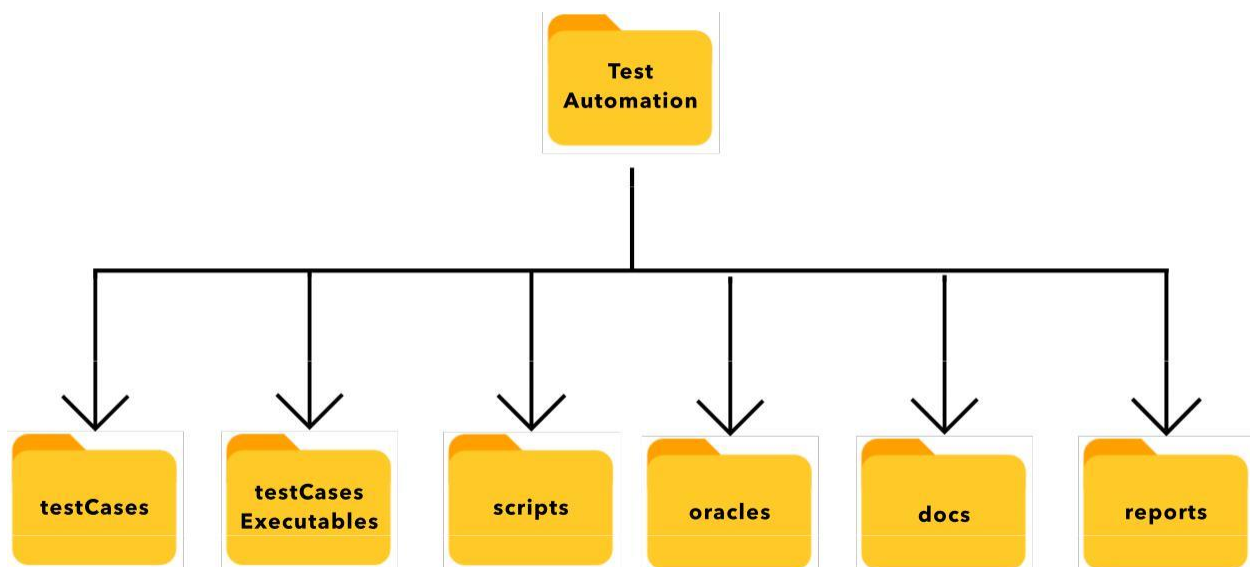
development has also been assisted by the work of open-source programmers. Moodle as a learning platform can enhance existing learning environments. As an E-learning tool, Moodle has a wide range of standard and innovative features such as a calendar and a Gradebook. Moodle is a leading virtual learning environment and can be used in many types of environments such as education, training and development and in business settings.

# FILE STRUCTURE

---

## *Framework Directories:*

Directories inside of the testing framework are structures in such a way that allows the framework to reach into certain directory and extract information. The framework as also capable of storing files so that can be referenced later.



### ***testCases:***

The “testCases” directory houses all the tests cases that can be run in the framework

### ***testCasesExecutables:***

The “testCasesExecutables” directory houses all the methods that are extracted from the Moodle directory. Files inside of “testCasesExecutables” are initialized when there are new methods that need to be tested. These methods are then instantiated in the driver.

### ***scripts:***

The “scripts” directory houses all the scripts needed to properly run the testing framework. This is where ./runAllTests.sh is stored which runs all test cases in the framework. Driver.php is also located in this directory. Driver.php initializes all the classes that are need to properly run Moodle methods and calls upon them.

### ***oracles:***

The “oracles” directory houses all the expected results that are obtained from the testCase.txt files.

### ***docs:***

The “docs” directory houses the README.rm file.

### ***reports:***

The “reports” directory houses the generated HTML file which is produced by the testing framework.

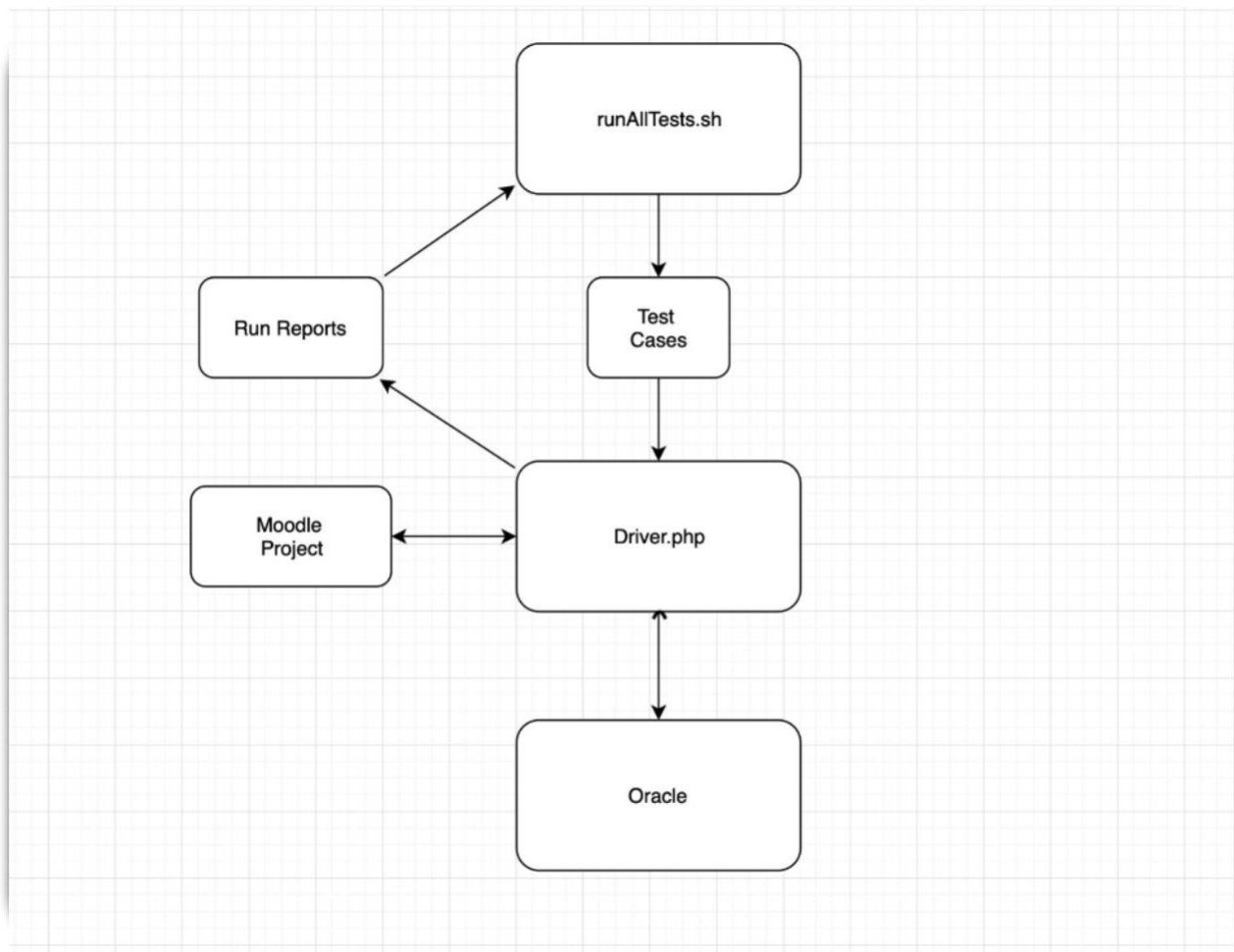
# ARCHITECTURE

---

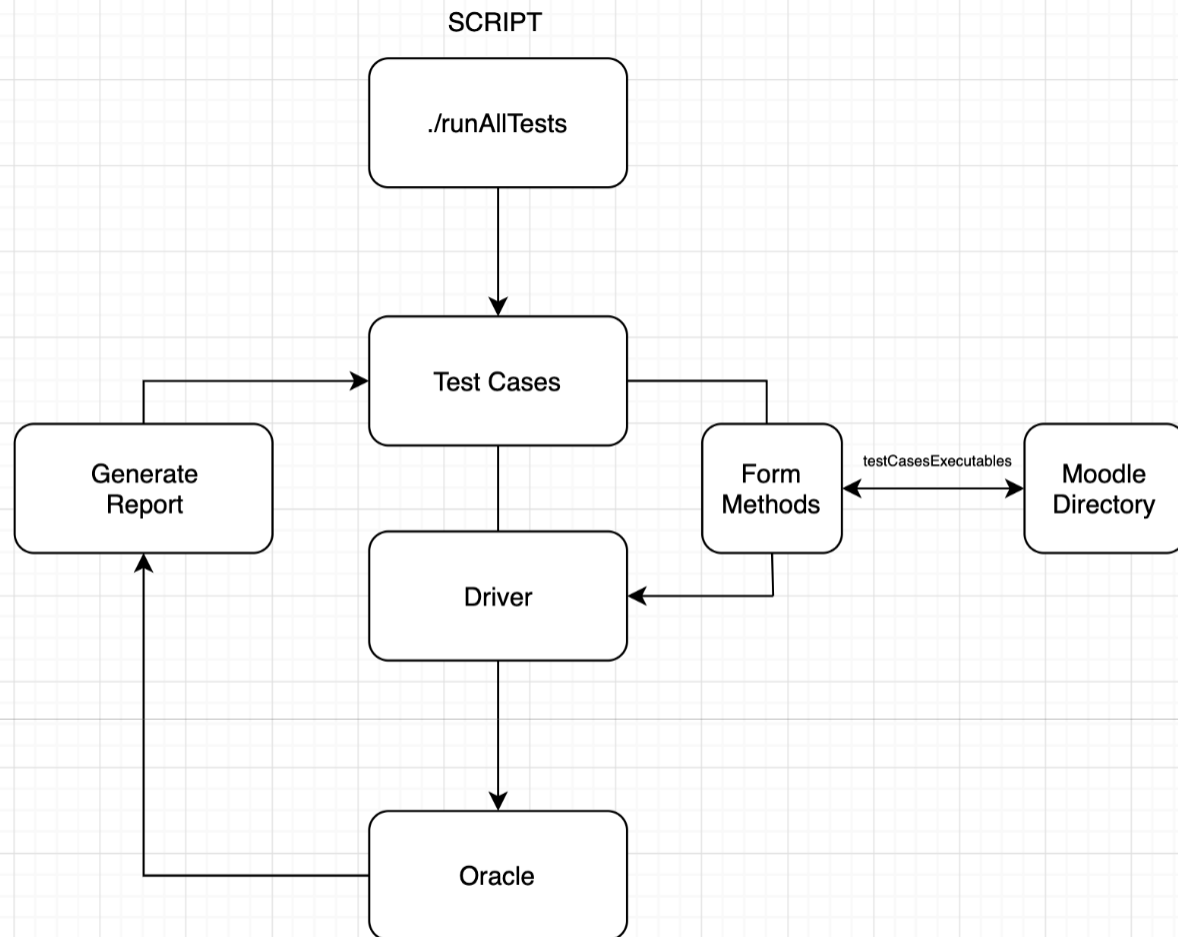
## *Overview:*

The architecture of the automatic testing framework has changed since that of deliverable 3. In deliverable 3 we expressed that we were in need of a way to instantiate our classes in order to properly test the majority of classes available in the massive project that is Moodle.

## *Previous Architecture:*



## Current Architecture:



# TEST CASES

---

001

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

@ @ @

Array ( )

002

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

" "

Array ( )

003

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

""

Array ( )

004

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

Hey! :) Whats up!

Array ( [0] => Hey [1] => Whats [2] => up )

005

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

:():@!Hey!\*

Array ( [0] => Hey )

006

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

Welcome to the framework!

Array ( [0] => Welcome [1] => to [2] => the [3] => framework )

007

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

Moodle is... fun!

Array ( [0] => Moodle [1] => is [2] => fun )

008

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

0101100

Array ( [0] => 0101100 )

009

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

"0101100"

Array ( [0] => 0101100 )

010

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

"The" "College" "of" "Charleston"

Array ( [0] => The [1] => College [2] => of [3] => Charleston )

011

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

(3,3),(5,9)

Array ( )



012

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

1000 \* 2 = 2000

Array ( [0] => 1000 [1] => 2000 )

013

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

"200 / 2 = 200"

Array ( [0] => 200 / 2 = 200 )

014

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

www.moodle.com

Array ( [0] => www [1] => moodle [2] => com )

015

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

tokenize(\$needle)

Array ( [0] => tokenize [1] => needle )

016

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

0.0021 is a decimal

Array ( [0] => 0021 [1] => is [2] => decimal )

017

The tokenize method takes in a string and returns an array with the words indexed

projects/moodle/grade/grading/lib.php

tokenize

"00.0021" is a string!

```
Array ( [0] => 00 [1] => 0021 [2] => is [3] => string )
```

018

The tokenize method takes in a string and returns an array with the words indexed

```
projects/moodle/grade/grading/lib.php
```

```
tokenize
```

```
~/$HOME/Desktop/CSCI362
```

```
Array ( [0] => ~/$HOME/Desktop/CSCI362 )
```

019

The tokenize method takes in a string and returns an array with the words indexed

```
projects/moodle/grade/grading/lib.php
```

```
tokenize
```

```
moodle -> Framework
```

```
Array ( [0] => moodle [1] => Framework )
```

020

The tokenize method takes in a string and returns an array with the words indexed

```
projects/moodle/grade/grading/lib.php
```

```
tokenize
```

```
"-----"
```

```
Array ( [0] => ----- )
```

021

The tokenize method takes in a string and returns an array with the words indexed

```
projects/moodle/grade/grading/lib.php
```

```
tokenize
```

```
"      "
```

```
Array ( )
```

022

The tokenize method takes in a string and returns an array with the words indexed

```
projects/moodle/grade/grading/lib.php
```

```
tokenize
```

```
Adding y0u t0 the class
```

```
Array ( [0] => Adding [1] => y0u [2] => t0 [3] => the [4] => class )
```

023

The tokenize method takes in a string and returns an array with the words indexed

```
projects/moodle/grade/grading/lib.php  
tokenize  
Hello world  
Array ( [0] => Hello [1] => world )
```

024

The tokenize method takes in a string and returns an array with the words indexed

```
projects/moodle/grade/grading/lib.php  
tokenize  
Test case  
Array ( [0] => Test [1] => case )
```

025

The tokenize method takes in a string and returns an array with the words indexed

```
projects/moodle/grade/grading/lib.php  
tokenize  
Nothing  
Array ( [0] => Nothing )
```

# CONCLUSION

---

Moodle is an open source software designed to assist in education. Most of the methods in the Moodle code rely on objects such as students, grades, courses, etc. This requires our test cases to have elaborate set up and break down methods. These set up methods are in place to instantiate the necessary objects to test the method, and the tear down objects reset to a good state so the tests can be run again without conflicting information. Programming these set-up methods require an intimate understanding of the software and the method being tested, as all

of the necessary objects must be created with specific parameters to allow the method to run. This was initially challenging to our team because Moodle was not a familiar software at the beginning of this project. After spending more time than we would like to admit familiarizing ourselves with the Moodle code, we now understand the methods well enough to program the set-up and tear-down methods.