

Milestone 5

Plants-Busters

Who:

- Erika Bailon
- Ruth Fernandes
- Khalid Aldawood
- Zhenqi Li
- Zhengwu Yuan
- Zijun Liu

Title:

- Plants Care

User Acceptance Test Template

Project Name: Plants Care

Project Manager: Erika Bailon

Document Date: April 22, 2018

Purpose: - Verify the basic features of our website

Prerequisites for this test: - None

Software used: - PHPUnit being used from terminal in the VM

Required Configuration: - We needed to install PHPUnit to be able to run the test files. Please refer to:

<https://phpunit.de/getting-started/phpunit-7.html>

1. UAT Scope (In Scope – Out of Scope)

UAT - In Scope	UAT - Out of Scope
<ul style="list-style-type: none">• Login• Sign up• Log out• Email about Q&A	<ul style="list-style-type: none">• Email verification• Display content• Email reminder• Adding a plant• Searching a plant

2. UAT Team Roles & Responsibilities

Name	Responsibilities
Erika Bailon	-Install PHPUnit in VM. -Run test php file
Khalid Aldawood	-Create test php files to test functions
Zhenqi Li	-Change current php files because it doesn't have functions

3. UAT Entry Criteria

ID	Criteria
3.1	<p>LOG-IN - Users will use this feature to access our website and their profile. Their profile contains all the plants that they have and users can check when they need to water certain plants. They can also add more plants to their profile once they log in.</p> <p>To test, we will create a test file and use <code>expectOutputString()</code> since we are using “echo” in our functions. If the expected output is not generated, then the test will be counted as a failure.</p>
3.2	<p>SING-UP - This page is for first time users. He/she will use this feature to sign-up for the website and its features. Users be prompted to enter their email, create and enter a username and password. Once he/she hits the signup button, their information will be stored in our user database.</p> <p>To test, we will create a test file and use <code>expectOutputString()</code> since we are using “echo” in our functions. If the expected output is not generated, then the test will be counted as a failure.</p>

3.3	<p>Q&A-EMAIL - This feature allows users to send website admin an email with his/her questions or comments. Users will enter their username, email, and comment/question and once he/she hits the send button, an email containing all of the information he/she entered will be sent to website admin.</p> <p>To test, we will create a file and use <code>expectOutputString()</code> since we are using “echo” in our functions. If the expected output is not generated, then the test will be counted as a failure.</p> <p>We were not able to complete this test since we found out about the type of file, and the type of software (PHPUnit) we needed to use to complete these tests.</p>
3.4	<p>LOG-OUT - This feature will allow users to log out of the website to make sure nobody access it and change their databases. The log-out will bring them back to the home page of the website which is the log-in.</p> <p>To test, we will create a file and use <code>expectOutputString()</code> since we are using “echo” in our functions. If the expected output is not generated, then the test will be counted as a failure.</p>

4. UAT Test Input/Results				
ID	Test Step/Input	Expected Output	Actual Output	Pass/Fail
3.1	In this function we have 4 parameters to check. In one we will give an email address to pass to the function and check if it is exactly equal to an address in the database. Other one will test if the “submit” button was	To check if the email is in the database, the output expected is true. To check that they click the “submit” button when both fields are filled(email and password) the expected output string is true. To check if the email is valid and if the email	True True False False	Pass

	<p>clicked. Also we have to check if both email and password were entered and if they were entered correctly, this counts as 2 tests. Please refer to the pictures below to see the code.</p>	<p>matched with the password, the output expected is false since we are setting it up to something that won't be valid so it will send a message that is not valid.</p>		
3.2	<p>The input for this step will be an email address, and two times the password. We will check 6 parameters in this test. First that the user fills up all the form. Second that the email is not in the database so it can be added. Third that everything is correct and the account was added. Fourth when the server is busy. Fifth, that both of the password in the fills to be filled out match. Sixth when it fails to be added. Pictures of code and test attached below.</p>	<p>The expected output is that some tests return true since different parameters are giving to test each of the outputs that can be given, and two return false within the results to make sure we will be able to sign them up.</p>	<p>True. True. True. False/True. False/False. True. True.</p>	<p>Pass</p>
3.3	<p>Not Done</p>	<p>Not Done</p>	<p>Not Done</p>	<p>Not Done</p>
3.4	<p>The input to test log-out is just checking if the log-out was clicked by the user. We need to check if it's false.</p>	<p>False</p>	<p>False</p>	<p>Pass</p>

	Pictures of code and test attached below.			
--	---	--	--	--

SCREENSHOTS

3.1

```

1  <?php
2  function login($submit,$email,$password,$num){
3      if($submit == "Sign in")
4      {
5          $user = $email;
6          $psw = $password;
7          if($user == "" || $psw == "")
8          {
9              echo "<script>alert('Enter your email and password! '); history.go(-1);</script>";
10         }
11         else
12         {
13             if($num)
14             {
15                 echo "<script>alert('Login successfully! ');</script>";
16                 echo"<script type=\"\".\"text/javascript\".\"\".\">\".\"window.location=\".\"\".\"content1.html\".\"\".\"</script>";
17             }
18             else
19             {
20                 echo "<script>alert('Incorrect email or password! ');history.go(-1);</script>";
21             }
22         }
23     }
24     else{
25         echo "<script>alert('cant submit!'); history.go(-1);</script>";
26     }
27 }
28
29 require_once 'PHPUnit/Autoload.php';
30
31 class OutputTest extends PHPUnit_framework_TestCase
32 {
33
34     public function testlogin1()
35     {
36         $this->expectOutputString(login("Sign in","", "",true));
37         print "<script>alert('Enter your email and password! '); history.go(-1);</script>";
38     }
39     public function testlogin2()
40     {

```

```

40     {
41         $this->expectOutputString(login("Sign in","123@123.com", "123", true));
42         print "<script>alert('Login successfully! ');</script><script type=\"text/javascript\">window.location='content1.html'</script><script>alert('Login successfully! ');</script><script type=\"text/javascript\">window.location='content1.html'</script>";
43     }
44     public function testlogin3()
45     {
46         $this->expectOutputString(login("test","123@123.com", "123", false));
47         print "<script>alert('cant submit!'); history.go(-1);</script>";
48     }
49     public function testlogin4()
50     {
51         $this->expectOutputString(login("Sign up","123@123.com", "123", false));
52         print "<script>alert('Incorrect email or password! ');history.go(-1);</script>";
53     }
54 }
55 }
56
57 ?>

```

```

user@cu-cs-vm:~/Desktop/unittest$ phpunit testlogin.php
PHPUnit 5.1.3 by Sebastian Bergmann and contributors.

```

```

.<script>alert('Enter your email and password! '); history.go(-1);</script><script>alert('Enter your email and password! '); history.go(-1);</script>.<script>alert('Login successfully! ');</script><script type="text/javascript">window.location="content1.html"</script><script>alert('Login successfully! ');</script><script type="text/javascript">window.location="content1.html"</script>.<script>alert('cant submit!'); history.go(-1);</script><script>alert('cant submit!'); history.go(-1);</script>.
4 / 4 (100%)<script>alert('cant submit!'); history.go(-1);</script><script>alert('Incorrect email or password! ');history.go(-1);</script>

```

```

Time: 14 ms, Memory: 4.00Mb

```

```

OK (4 tests, 0 assertions)

```

3.2

```
1  <?php
2  function signup($signup,$email,$password,$confirm,$exist,$res_insert){
3      if($signup == "Sign up")
4      {
5          $user = $email;
6          $psw = $password;
7          $psw_confirm = $confirm;
8          if($user == "" || $psw == "" || $psw_confirm == "")
9          {
10             echo "<script>alert('Enter your email and password! '); history.go(-1);</script>";
11         }
12         else
13         {
14             if($psw == $psw_confirm)
15             {
16                 if($exist)
17                 {
18                     echo "<script>alert('Already exist email'); history.go(-1);</script>";
19                 }
20                 else
21                 {
22                     if($res_insert)
23                     {
24                         echo "<script>alert('Sign up successfully! ');</script>";
25                     }
26                     else
27                     {
28                         echo "<script>alert('Server Busy! '); history.go(-1);</script>";
29                     }
30                 }
25         echo"<script type=\"\").\"text/javascript\".\"\".\">\".window.location=\"\").\"Login.html\".\"\".\"</script>";

```

```

31         }
32     }
33     else if($psw != $psw_confirm)
34     {
35         echo "<script>alert('Passwords have to be the same!'); history.go(-1);</script>";
36     }
37 }
38 }
39 else
40 {
41     echo "<script>alert('Fail! '); history.go(-1);</script>";
42 }
43 }
44
45 require_once 'PHPUnit/Autoload.php';
46
47 class OutputTest extends PHPUnit_framework_TestCase
48 {
49
50     public function testsignup1()
51     {
52         $this->expectOutputString(signup("", "", "123", "123", true, true));
53         print "<script>alert('Enter your email and password! '); history.go(-1);</script>";
54     }
55     public function testsignup2()
56     {
57         $this->expectOutputString(signup("Sign up", "123@123.com", "123", "123", true, true));
58         print "<script>alert('Already exist email'); history.go(-1);</script>";
59     }
60     public function testsignup3()
61     {
62         $this->expectOutputString(signup("Sign up", "123@123.com", "123", "123", false, true));
63         print "<script>alert('Sign up successfully! ');</script><script type=\"\text/javascript\".\"\".\">.\"window.location=
64     }
65     public function testsignup4()
66     {
67         $this->expectOutputString(signup("Sign up", "123@123.com", "123", "123", false, false));
68         print "<script>alert('Server Busy! '); history.go(-1);</script>";
69
70     }
71     public function testsignup5()
72     {
73         $this->expectOutputString(signup("Sign up", "123@123.com", "123", "321", true, true));
74         print "<script>alert('Passwords have to be the same!'); history.go(-1);</script>";
75     }
76     public function testsignup6()
77     {
78         $this->expectOutputString(signup("test", "123@123.com", "123", "321", true, true));
79         print "<script>alert('Fail! '); history.go(-1);</script>";
80     }
81 }
82 ?>

```



```

user@cu-cs-vm:~/Desktop/unittest$ phpunit testsignup.php
PHPUnit 5.1.3 by Sebastian Bergmann and contributors.

.<script>alert('Enter your email and password! '); history.go(-1);</script><scri
pt>alert('Enter your email and password! '); history.go(-1);</script><script>al
ert('Already exist email'); history.go(-1);</script><script>alert('Already exist
email'); history.go(-1);</script><script>alert('Sign up successfully! ');</scr
ipt><script type="text/javascript">window.location="Login.html"</script><script>
alert('Sign up successfully! ');</script><script type="text/javascript">window.l
ocation="Login.html"</script><script>alert('Server Busy! '); history.go(-1);</s
cript><script>alert('Server Busy! '); history.go(-1);</script><script>alert('Pa
sswords have to be the same!'); history.go(-1);</script><script>alert('Passwords
have to be the same!'); history.go(-1);</script>.
6 / 6 (100%)<script>alert('Fail! '); history.go(
-1);</script><script>alert('Fail! '); history.go(-1);</script>

Time: 15 ms, Memory: 4.00Mb

OK (6 tests, 0 assertions)

```

3.4

```

1  <?php
2  function logout($logout){
3  if ($logout == "Log out")
4  {
5      echo"<script>alert('logout!');</script>";
6      echo"<script type=\"",\"text/javascript\", \">\", \"window.location=\", \"\", \"Login.html\", \"\", \"</script>\";
7  }
8
9  else{
10
11      echo "false";
12  }
13  }
14  }
15
16  require_once 'PHPUnit/Autoload.php';
17
18  class OutputTest extends PHPUnit_Framework_TestCase
19  {
20
21      public function testlog()
22      {
23          $this->expectOutputString(logout("Log out"));
24          print "<script>alert('logout!');</script><script type=\", \"text/javascript\", \">\", \"window.location=\", \"\", \"Login.h
25      }
26      public function testlog2()
27      {
28          $this->expectOutputString(logout("test"));
29          print "flase";
30      }
31  }
32
33  ?>

```

```
user@cu-cs-vm:~/Desktop/unittest$ phpunit testlogout.php
PHPUnit 5.1.3 by Sebastian Bergmann and contributors.

.<script>alert('logout!');</script><script type="text/javascript">window.location="Login.html"</script><script>alert('logout!');</script><script type="text/javascript">window.location="Login.html"</script>.
2 / 2 (100%)falseflase

Time: 14 ms, Memory: 4.00Mb

OK (2 tests, 0 assertions)
user@cu-cs-vm:~/Desktop/unittest$
```