

Test Konzept
Homepage "hochsicherheit.ch"



Verfasser	Severin Müller
Dozent	Matthias Bachmann
Modul	Programmieren
Seminar	"Webprojekt in PHP/MySQL"
Erscheinungsjahr	2014

Inhalt

1. Einleitung	3
2. Unit Tests	4
2.1 Einleitung	4
2.2 Test Cases.....	4
2.2.1 MySQL Tests	4
2.2.2 Inhaltstest	4
3. Performance Test.....	5
4. Abhahmetest und Protokoll	8
4.1 Laden der Seiten.....	8
4.1.1 Hauptseite	8
4.1.2 Startseite	8
4.1.3 Über uns	8
4.1.4 Dienstleistungen.....	8
4.1.5 Allgemeine Geschäftsbedingungen	8
4.1.6 Referenzen	8
4.1.7 Links	8
4.1.8 Intern	9
4.1.9 Kontakt.....	9
4.1.10 Navigationsleiste	9
4.2 Kontaktformular	9
4.2.1 Absenden des Formulars mit allen Pflichtfelder ausgefüllt.....	9
4.2.2 Absenden des Formulars mit fehlenden Pflichtfeldern.....	9
4.2.3 Absenden des Formulars mit ungültiger CAPTCHA Eingabe.....	9
4.2.4 Feldvalidierung bei Eingabe	9
4.2.5 E-Mail Empfang	10
4.3 Interner Bereich	10
4.3.1 Login mit korrekten Daten	10
4.3.2 Login mit ungültigen Daten.....	10
4.3.3 Hochladen Dokumente	10
4.3.4 Generierung Löschlinks.....	10
4.3.5 Löschen von Dateien.....	10
4.3.6 Herunterladen von Dokument	11
5. Anhang.....	12
5.1 Anhang A: Bilderverzeichnis	12
5.2 Anhang B: Literaturverzeichnis	12

1. Einleitung

Für die Qualitätssicherung sind Tests notwendig. Ich habe mich entschieden, meine Tests auf mehrere Säulen zu stellen, um eine möglichst gute Abdeckung zu erhalten.

Bereits bei der Entwicklung wurden verschiedene Unit Tests geschrieben, um sicherzustellen, dass die kritischen Methoden funktionieren.

Da es nervenaufreibend ist, wenn man auf das Laden einzelner Seiten warten muss habe ich auch einen Performance Test durchgeführt.

Als letzte Stufe kommt der Abnahmetest dran. Der Entwurf wurde bereits für gut befunden, der finale Abnahmetest kommt dann beim Deployment im Juli zum Tragen. Dieser wird in erster Linie ein Klick-Test sein, bei dem der Kunde alle Unterseiten durchklicken wird und so allfällige Fehler noch finden kann.

2. Unit Tests

2.1 Einleitung

Bei der Implementierung wurde Test Driven, also Testgetrieben entwickelt. Dies hat sich in der Vergangenheit bewährt, weil zunächst die Methoden getestet werden, bevor man sie im effektiven Code verwendet. Sämtliche Unit Tests wurden mit PHPUnit durchgeführt.

2.2 Test Cases

Folgende Test Cases sind erstellt und durchgeführt worden:

2.2.1 MySQL Tests

Bei der Implementierung der Interfaces Database sind zwei Methoden zu überschreiben. Für die Implementierung dieser Methoden wurde jeweils ein Unit Test geschrieben:

```
public function testConnection()  
public function testLogin()
```

Da wir in der Klasse MySQL jedoch noch weitere Methoden bereitgestellt haben, haben wir für diese ebenfalls Unit Tests durchgeführt:

```
public function testAddUser()
```

2.2.2 Inhaltstest

Das Laden der Inhalte wird mit zwei PHP Funktionen realisiert. Daher sind hier eben Unit Tests vonnöten.

Die beiden Methoden für das Laden des Titels und der Seiteninhalts wurden mit jeweils einem Unit Test getestet:

```
public function testTitle()  
public function testContent()
```

3. Performance Test

Um sicherzustellen, dass die Performance im grünen Bereich ist, ist diese zu testen. Um einfach Messungen durchzuführen wird das Tool Xdebug installiert. Dieses erstellt Profile von Klicks auf der Homepage und stellt ein Output File bereit, welches durch das Tool Webgrind visualisiert dargestellt werden kann.

Hier sind die Resultate von unseren Testläufen:

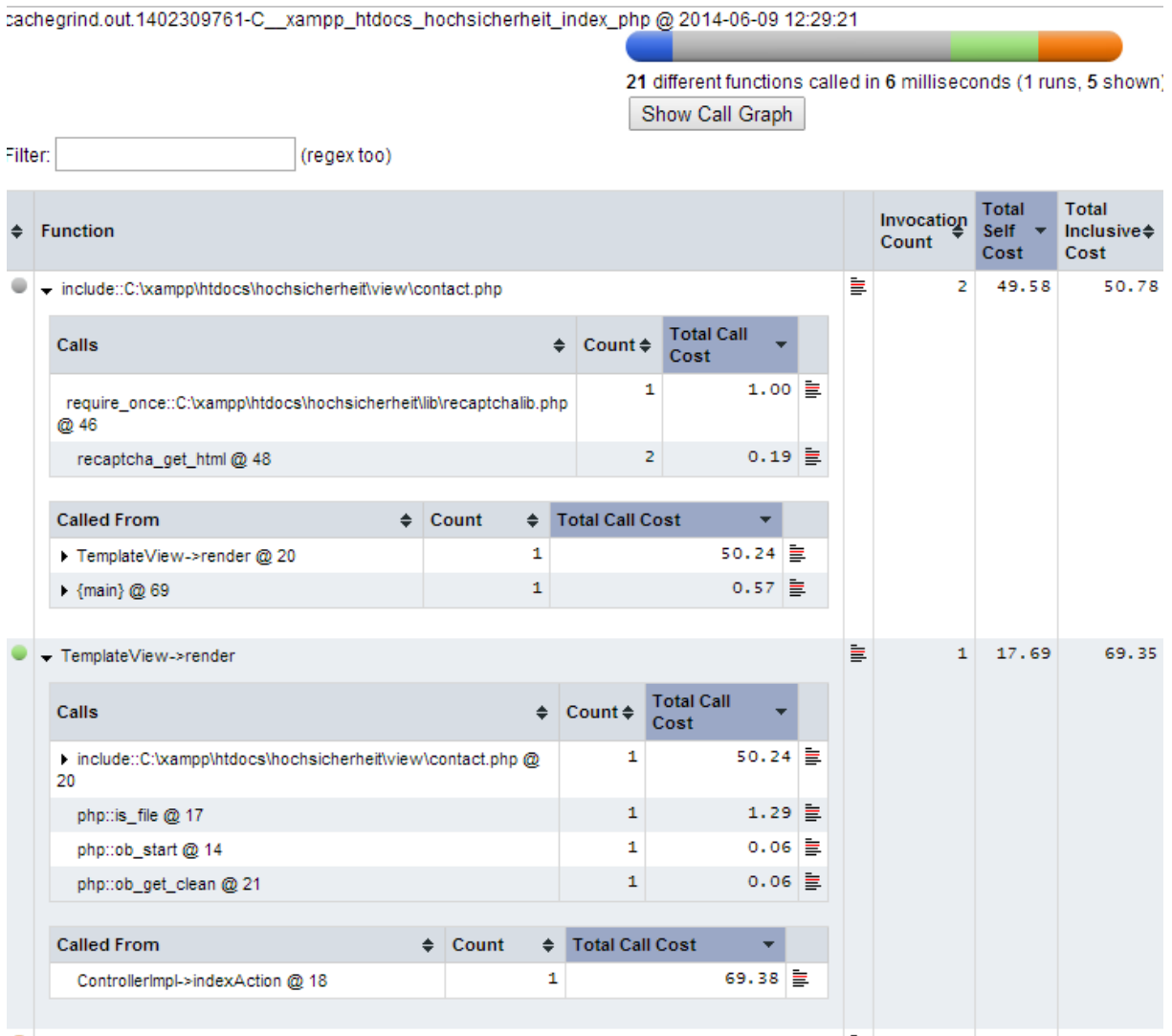


Abb. 1: Performance Test Index Teil 1

▼ {main}			1	14.33	99.95
Called from script host					
Calls	Count	Total Call Cost			
ControllerImpl->indexAction @ 69	1	69.89			
▶ require_once::C:\xampp\htdocs\hochsicherheit\controller\ControllerImpl.php @ 8	1	9.37			
▶ php::session_start @ 6	1	3.96			
php::realpath @ 7	1	0.72			
▶ include::C:\xampp\htdocs\hochsicherheit\view\contact.php @ 69	1	0.57			
ControllerImpl->__construct @ 16	1	0.46			
load_navigation @ 63	1	0.38			
php::ini_set @ 3	1	0.09			
php::error_reporting @ 4	1	0.05			
php::ini_set @ 5	1	0.05			
php::define @ 7	1	0.05			
require_once::C:\xampp\htdocs\hochsicherheit\inc\main.php.inc @ 9	1	0.05			
▼ require_once::C:\xampp\htdocs\hochsicherheit\controller\ControllerImpl.php			1	6.94	9.34
Calls	Count	Total Call Cost			
require::C:\xampp\htdocs\hochsicherheit\controller\Controller.php @ 6	1	1.65			
require::C:\xampp\htdocs\hochsicherheit\lib\class.TemplateView.php @ 7	1	0.64			
php::ini_set @ 3	1	0.08			
php::error_reporting @ 4	1	0.03			
Called From	Count	Total Call Cost			
▶ {main} @ 8	1	9.37			

Abb. 2: Performance Test Index Teil 2

▼ php::session_start			1	3.96	3.96
Called From	Count	Total Call Cost			
▶ {main} @ 6	1	3.96			

Abb. 3: Performance Test index Teil 3

Somit haben wir die Hauptseite mit ihren Unterseiten getestet und die Performance ist in Ordnung.

Als nächstes Testen wir einen Datei Upload

Filter: (regex too)

Function	Invocation Count	Total Self Cost	Total Inclusive Cost
php::fopen	1	53.29	53.29
php::PDOStatement->execute	1	17.14	17.14
php::PDO->__construct	1	11.06	11.06
{main}	1	7.20	99.96
MySQL->add_document	1	3.55	32.56

Abb. 4: Performance Dateiupload

Wir sehen hier, dass dieser Aufruf etwas länger geht, was aber auch logisch ist weil die Datei erst hochgeladen werden muss. Besonders erfreulich ist, dass die MySQL Funktion eine sehr gute Performance aufweist.

Als letztes haben wir uns das Kontaktformular unter die Lupe genommen:

Filter: (regex too)

Function	Invocation Count	Total Self Cost	Total Inclusive Cost															
include::C:\xampp\htdocs\hochsicherheit\view\contact.php	2	49.58	50.78															
<table><tr><th>Calls</th><th>Count</th><th>Total Call Cost</th></tr><tr><td>require_once::C:\xampp\htdocs\hochsicherheit\lib\recaptcha\lib.php @ 46</td><td>1</td><td>1.00</td></tr><tr><td>recaptcha_get_html @ 48</td><td>2</td><td>0.19</td></tr></table>	Calls	Count	Total Call Cost	require_once::C:\xampp\htdocs\hochsicherheit\lib\recaptcha\lib.php @ 46	1	1.00	recaptcha_get_html @ 48	2	0.19									
Calls	Count	Total Call Cost																
require_once::C:\xampp\htdocs\hochsicherheit\lib\recaptcha\lib.php @ 46	1	1.00																
recaptcha_get_html @ 48	2	0.19																
<table><tr><th>Called From</th><th>Count</th><th>Total Call Cost</th></tr><tr><td>TemplateView->render @ 20</td><td>1</td><td>50.24</td></tr><tr><td>{main} @ 69</td><td>1</td><td>0.57</td></tr></table>	Called From	Count	Total Call Cost	TemplateView->render @ 20	1	50.24	{main} @ 69	1	0.57									
Called From	Count	Total Call Cost																
TemplateView->render @ 20	1	50.24																
{main} @ 69	1	0.57																
TemplateView->render	1	17.69	69.35															
<table><tr><th>Calls</th><th>Count</th><th>Total Call Cost</th></tr><tr><td>include::C:\xampp\htdocs\hochsicherheit\view\contact.php @ 20</td><td>1</td><td>50.24</td></tr><tr><td>php::is_file @ 17</td><td>1</td><td>1.29</td></tr><tr><td>php::ob_start @ 14</td><td>1</td><td>0.06</td></tr><tr><td>php::ob_get_clean @ 21</td><td>1</td><td>0.06</td></tr></table>	Calls	Count	Total Call Cost	include::C:\xampp\htdocs\hochsicherheit\view\contact.php @ 20	1	50.24	php::is_file @ 17	1	1.29	php::ob_start @ 14	1	0.06	php::ob_get_clean @ 21	1	0.06			
Calls	Count	Total Call Cost																
include::C:\xampp\htdocs\hochsicherheit\view\contact.php @ 20	1	50.24																
php::is_file @ 17	1	1.29																
php::ob_start @ 14	1	0.06																
php::ob_get_clean @ 21	1	0.06																
<table><tr><th>Called From</th><th>Count</th><th>Total Call Cost</th></tr><tr><td>ControllerImpl->indexAction @ 18</td><td>1</td><td>69.38</td></tr></table>	Called From	Count	Total Call Cost	ControllerImpl->indexAction @ 18	1	69.38												
Called From	Count	Total Call Cost																
ControllerImpl->indexAction @ 18	1	69.38																
{main}	1	14.33	99.95															
require_once::C:\xampp\htdocs\hochsicherheit\controller\ControllerImpl.php	1	6.94	9.34															
php::session_start	1	3.96	3.96															

Abb. 5: Performance Test Kontaktformular

Auch hier sind unsere Funktionen gut dabei mit der Performance.

Abschliessend können wir sagen, dass unsere Applikationen gute Ausführungszeiten hat.

4. Abnahmetest und Protokoll

Für die Abnahme des Projekts werden einige Test Cases erstellt. Die Cases und deren Resultat sind nachfolgend beschrieben.

4.1 Laden der Seiten

4.1.1 Hauptseite

Erwartetes Resultat: Die Hauptseite lädt und die Startseite wird angezeigt

Durchführung: 06.06.2014

Resultat: **ok**

4.1.2 Startseite

Erwartetes Resultat: Die Seite wird korrekt angezeigt

Durchführung: 06.06.2014

Resultat: **ok**

4.1.3 Über uns

Erwartetes Resultat: Die Seite wird korrekt angezeigt

Durchführung: 06.06.2014

Resultat: **ok**

4.1.4 Dienstleistungen

Erwartetes Resultat: Die Seite wird korrekt angezeigt

Durchführung: 06.06.2014

Resultat: **ok**

4.1.5 Allgemeine Geschäftsbedingungen

Erwartetes Resultat: Die Seite wird korrekt angezeigt

Durchführung: 06.06.2014

Resultat: **ok**

4.1.6 Referenzen

Erwartetes Resultat: Die Seite wird korrekt angezeigt und die Links inkl. deren Weiterleitungen funktionieren korrekt.

Durchführung: 06.06.2014

Resultat: **ok**

4.1.7 Links

Erwartetes Resultat: Die Seite wird korrekt angezeigt und die Links inkl. deren Weiterleitungen funktionieren korrekt.

Durchführung: 06.06.2014

Resultat: **ok**

4.1.8 Intern

Erwartetes Resultat: Die Loginseite wird korrekt angezeigt. Das Login funktioniert korrekt und der interne Bereich wird korrekt angezeigt.

Durchführung: 06.06.2014

Resultat: **ok**

4.1.9 Kontakt

Erwartetes Resultat: Die Seite wird korrekt angezeigt. Das CAPTCHA Bild ist lesbar und sauber dargestellt.

Durchführung: 06.06.2014

Resultat: **ok**

4.1.10 Navigationsleiste

Erwartetes Resultat: Die Navigationsleiste wird korrekt angezeigt und die Links funktionieren korrekt. Die aktuelle Seite soll in der Navigationsleiste eingefärbt sein.

Durchführung: 07.06.2014

Resultat: **ok**

4.2 Kontaktformular

4.2.1 Absenden des Formulars mit allen Pflichtfelder ausgefüllt

Erwartetes Resultat: Das Kontaktformular kann abgeschickt werden, wenn alle Pflichtfelder korrekt ausgefüllt sind.

Durchführung: 06.06.2014

Resultat: **ok**

4.2.2 Absenden des Formulars mit fehlenden Pflichtfeldern

Erwartetes Resultat: Der Benutzer kriegt eine Fehlermeldung mit dem Hinweis, dass nicht alle Felder korrekt ausgefüllt worden sind.

Durchführung: 06.06.2014

Resultat: **ok**

4.2.3 Absenden des Formulars mit ungültiger CAPTCHA Eingabe

Erwartetes Resultat: Das Kontaktformular zeigt eine Fehlermeldung an wenn das CAPTCHA Feld nicht korrekt ausgefüllt wurde.

Durchführung: 06.06.2014

Resultat: **ok**

4.2.4 Feldvalidierung bei Eingabe

Erwartetes Resultat: Die roten Daumen nach unten werden grün, wenn ein Pflichtfeld ausgefüllt wurde. Bei Feld E-Mail wird zudem eine gültige E-Mail erwartet. Werden die Felder geleert werden die grünen Daumen wieder rot.

Durchführung: 06.06.2014

Resultat: **ok**

4.2.5 E-Mail Empfang

Erwartetes Resultat: Das gesendete Kontaktformular kommt als korrekt dargestellte E-Mail beim Kunden an.

Durchführung: 06.06.2014

Resultat: **ok**

4.3 Interner Bereich

4.3.1 Login mit korrekten Daten

Erwartetes Resultat: Werden gültige Zugangsdaten eingegeben ist der Login erfolgreich und der interne Bereich wird korrekt angezeigt.

Durchführung: 07.06.2014

Resultat: **ok**

4.3.2 Login mit ungültigen Daten

Erwartetes Resultat: Werden ungültige Daten eingegeben wird der Benutzer zurück zur Loginseite geführt mit entsprechendem Hinweis.

Durchführung: 07.06.2014

Resultat: **ok**

4.3.3 Hochladen Dokumente

Erwartetes Resultat: Ein Dokument kann ausgewählt und hochgeladen werden. Nach dem Hochladen wird eine entsprechende Meldung ausgegeben und das Dokument ist auf der Seite sichtbar. Das Dokument wird in der korrekten Kategorie angezeigt.

Durchführung: 07.06.2014

Resultat: **ok**

4.3.4 Generierung Löschlinks

Erwartetes Resultat: Wurde ein Dokument hochgeladen muss der Link „Löschen“ angezeigt werden.

Durchführung: 07.06.2014

Resultat: **ok**

4.3.5 Löschen von Dateien

Erwartetes Resultat: Wird der Link „Löschen“ bei einem Dokument gedrückt wird dieses gelöscht und ist nicht mehr sichtbar.

Durchführung: 07.06.2014

Resultat: **ok**

4.3.6 Herunterladen von Dokument

Erwartetes Resultat: Wir auf ein hochgeladenes Dokument geklickt wird dieses heruntergeladen und kann korrekt geöffnet werden.

Durchführung: 07.06.2014

Resultat: **ok**

5. Anhang

5.1 Anhang A: Bilderverzeichnis

Abb. 1: Performance Test index Teil 1	5
Abb. 2: Performance Test index Teil 2	6
Abb. 3: Performance Test index Teil 3	6
Abb. 4: Performance Dateiupload	7
Abb. 5: Performance Test Kontaktformular	7

5.2 Anhang B: Literaturverzeichnis

Folgende Quellen wurden bei der Bearbeitung dieses Dokuments hinzugezogen:

<https://code.google.com/p/webgrind/>

<http://xdebug.org/>

<http://phpunit.de/>