# Seminararbeit 2014

# Homepage “hochsicherheit.ch”

****

|  |  |
| --- | --- |
| Verfasser | Severin Müller |
| Dozent | Matthias Bachmann |
| Modul | Programmieren |
| Seminar | “Webprojekt in PHP/MySQL” |
| Erscheinungsjahr | 2014 |

# 1. Themenwahl und Aufgabenstellung

## 1.1 Eigenmotivation

In der heutigen Zeit spielt das Internet eine zentrale Rolle im Privat- und Geschäftsleben. Schon früh habe ich mich mit dem Design und der Entwicklung von Homepages auseinandergesetzt weil ich diesen Trend erkannte und mich die Materie sehr interessierte. Nicht nur am Desktop, sondern auch unterwegs ist man heute meistens irgendwie mit dem Internet verbunden und man möchte Informationen jederzeit abrufen können.

Da aber längst nicht jeder sich mit dem Gestalten von Homepages auseinandersetzen benötigt man Fachleute, die sich dieser Thematik annehmen. Die Hochsicherheit GmbH ist ein junges Startup welches Sicherheits-Dienste für kleinere und mittelgrosse Veranstaltungen anbietet. Da dieses noch sehr junge Unternehmen einen Kundenstamm aufbauen möchte liegt die Vermutung nahe, dass ein ansprechender Internetauftritt dabei helfen kann. Da es sich beim Firmengründer um einen Freund von mir handelt, habe ich ihm angeboten, diesen Auftritt für ihn zu erstellen.

## 1.2 Abgrenzung der Aufgabenstellung

Das Erstellen einer Homepage ist technisch nicht besondern komplex. Einer der grössten Fehler die im Webdesign immer wieder gemacht werden ist das “drauflos Entwickeln” ohne mit dem Kunden Rücksprache oder ausreichend Rücksicht auf seine Anforderungen zu nehmen. Da der Kunde sich aber meist nicht bewusst ist, worauf es beim Gestalten ankommt ist die bilaterale Kommunikation von grosser Bedeutung. Daher möchte ich den Fokus in dieser Arbeit auf das Requirement Engineering legen.

Das Ziel dieser Arbeit ist es, eine Homepage zu erstellen die Funktionsfähig ist und den Anforderungen des Kunden entspricht. Inbesondere die folgenden drei Teilziele sollen erreicht werden:

- Sauberes Requirement Engineering

- Test Driven Development

- Erfolgreiche Abnahme durch den Kunden

# 2. Projektablauf

Die nachfolgenden Abschnitte sollen eine kurze Zusammenfassung darüber sein, wie das Projekt ablaufen soll.

## 2.1. Vorbesprechung mit dem Kunden

Um das Projekt überhaupt starten zu können war eine Vorbesprechung mit dem Kunden nötig. Er hat folgende Wünsche geäussert:

“Ich möchte einen Internetauftritt für meine neue Firma haben die für Interessenten Informationen über unsere Diensleistungen bereitstellt. Der Kunde soll Einsätze online buchen und mit uns über ein Formular Kontakt aufnehmen können. Ich habe eine Vorstellung davon, wie der Auftritt ungefährt aussehen soll und meine Partnerin hat bereit Logos kreiert.”

Es wurde vereinbart, dass ich die Logos per E-Mail erhalte und mit die farblichen Wünsche schriftlich mitgeteilt werden.

## 2.2 Erfassung inhaltliche Wünsche

In einer zweiten Vorbesprechung hat mir der Kunde Entwürfe der einzelnen Seiteninhalte vorgelegt. Dieser werden direkt so übernommen.

## 2.3. Erfassung gestalerische Wünsche

Wie der Kunde bereits in der Vorbesprechung erwähnte, hat er klare Vorstellungen darüber wie der Internetauftritt gestaltet werden soll. Er wurde vereinbart, dass ich einen ersten Entwurf kreieren und vorlegen werde**.**

## 2.4. Erfassung funktionale Wünsche

Der Kunde hat zunächst gewünscht, dass die Einsätze online gebucht werden können. Im Verlaufe der Projekts hat er aber den Wunsch geäussert, von dieser Funktion abzusehen, weil er seine Termine lieber selber koordinieren möchte und so keine Kundenerwartungen wecken muss, die dann nicht erfüllt werden können. Das Kontaktformular soll aber wie gewünscht erstellt werden.

## 2.5. Vorlegen von Design Beispielen

Wie bereits erwähnt, wurde mit dem Kunden vereinbart, dass vor der eigentlichen Entwicklung ein Designbeispiel vorgelegt wird, dass bei Bedarf angepasst wird.

## 2.6. Abschluss der Spezifikation

Hier findet das letzte Treffen mit dem Kunden vor der Implementierung statt. Das Design und die Inhalte werden abgenommen, kleinere Anpassung können bei Bedarf aber natürlich vorgenommen werden.

## 2.7. Implementierung des Projekts

In einer ersten Phase wird aufgrund der Spezifikation ein Requirement Engineering durchgeführt. Dessen Zweck ist es, den Problemraum zu erfassen und zu dokumentieren und eine Verbindung zwischen Problem- und Lösungsraum herzustellen1.

Sobald das Requirement Engineering abgeschlossen ist, kann die Implementierung beginnen.

## 2.8. Testen des Projekts

Wenn die Implementierung abgeschlossen ist, muss das Projekt getestet werden. Unit Tests ersetzen keinen finalen Test, in dem alle Funktionen ausreichend getestet werden.

## 2.9. Abnahme des Projekts durch den Kunden / Projektabschluss

Sobald das Projekt fertig implementiert und getestet wurde kann es dem Kunden zur Abnahme vorgelegt werden.

# 3. Spezifikation

## 3.1 Requirement Engineering - Einführung

Für die Abgrenzung der Anforderungen existieren verschiedene Modelle. Die Unterschiede sind aber gering. Gemäss IEEE wird das Requirement Engineering in folgende Schritte unterteilt2:

* Anforderungserhebung
* Anforderungsanalyse
* Anforderungsspezifikation
* Anforderungsbewertung

Wir wollen diese Schritte nun in unser Projekt einfliessen lassen. Die detaillierte Anforderungsbeschreibung finden Sie im Dokument “Anforderungsdokument”.

## 3.2 Entwurf

Nachdem ich die gesamten Anforderungen mit dem Kunden besprochen und abgeschlossen hatte machte ich mir Gedanken zum Design. Mir schwebte eine einfach wartbare Applikation vor, bei der die Daten und die Präsentation sauber getrennt werden. Deshalb habe ich mich für ein MVC Pattern entschieden, welches Good Practice in der Webentwicklung ist.

Die Umsetzung erfolgte mittels PHP, HTML und CSS.

## 3.3 Umgebung

Da für die gewünschte Homepage bestimmte Komponenten zur Verfügung stehen müssen, habe ich meinem Kunden angeboten, die Applikation auf dem Webserver bei meinem Hoster zu betreiben. Da weiss ich, dass ich sämtliche benötigten Teilsysteme verfügbar habe.

# 4. Lerninhalte

## 4.1 MVC

### 4.1.1 Einführung

Model View Controller (kurz MVC) ist ein Muster zur Strukturierung von Software-Entwicklung in die drei Einheiten Datenmodell (engl. model), Präsentation (engl. view) und Programmsteuerung (engl. controller)3

Das Ziel ist dabei die Trennung von der Programmlogik und der Darstellung. Dies hat den Vorteil, dass der Designer der Darstellung nicht mehr zwingend der Programmierer sein muss, sondern sich wirklich nur um das Design kümmern muss.

Wie MVC in diesem Projekt implementiert wurde ist im beiliegenden Entwurfsdokument nachzulesen.

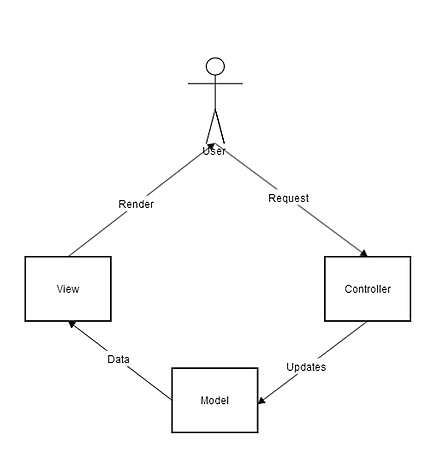


Abb. 1: MVC Modell

### 4.1.2 Model

Das Modell bearbeitet die Daten, die für die Darstellung notwendig sind. Sie werden vom Controller angefordert.

### 4.1.2 View

Die View ist das, was der User am Ende effektiv sieht. Sie erhält vom Model die Daten und wird typischerweise mit einem Template umgesetzt.

### 4.1.3 Controller

Der Controller ist für die Steuerung der Applikationen zuständig. Er nimmt Benutzeraktionen entgegen und leitet sie an die zuständige Stelle weiter.

## 4.2 PHP Interfaces und Klassen

Die Grundlagen der objektorientierten Programmierung dürfen wir für dieses Seminar als bekannt voraussetzen. Auf allgemeine Theorie gehe ich daher nicht näher ein.

Seit Version 3 unterstützt PHP eine objektorientierte Syntax4. Genau wie in Java kann man mittels interfaces eine Schnittstelle schaffen und Methoden definieren, welche von einer Klasse implementiert werden müssen, ohne dass ein Implementierungsdetail preisgegeben werden muss. Eine Klasse kann ein Interface implementieren und dessen Methoden müssen überschrieben werden. Eine weitere Abstraktionsstufe kann mittels abstrakten Klassen erreicht werden.

## 4.3. PHP und MySQL

PHP liefert eine vollständige Unterstützung für MySQL.

# Literaturverzeichnis

[1]: <http://www11.informatik.uni-erlangen.de/Lehre/SS2010/PR-SWE/Folien/fa1.pdf>, Folie 5, abgerufen am 29.03.2014

[2]: http://de.wikipedia.org/wiki/Anforderungsanalyse\_(Informatik), abgerufen am 29.03.2014

[3]: https://de.wikipedia.org/wiki/Model\_View\_Controller, abgerufen am 29.03.2014

[4]: <http://www.php.net/manual/de/history.php.php>, abgerufen am 12.04.2014

unit testing <http://www.sitepoint.com/tutorial-introduction-to-unit-testing-in-php-with-phpunit/>

database test phpunit: http://phpunit.de/manual/3.7/en/database.html