

Shablog (privates Weblog)

Shayan Shamseddin (2103763)

Betreuer: Prof. Dr. Andreas Plaß

08. Jan 2017

Projekt A

Studiengang Media Systems (B.Sc.)

Hochschule für angewandte Wissenschaften Hamburg /

Hamburg University of Applied Sciences

Department Medientechnik

Fakultät Design, Medien und Information

Inhalt

1. Projekt.....	3
1.1 Idee.....	3
1.2 Handlung	3
1.2.1 An-Abmeldung.....	3
1.2.2 Home.....	3
1.2.3 Einstellung.....	4
1.2.4 Photos/Videos	4
2. Installationsanleitung	4
3. Bedienungsanleitung	5
4. Systemarchitektur.....	7
5. Beschreibung eines technischen Teilaspektes	9
6. Fazit, Auswertung.....	10
7. Quellen.....	11
7.1 Links	11
7.2 Abbildung.....	11

1. Projekt

1.1 Idee

In meinem Projekt handelt es sich um ein privates Weblog. Ein Weblog ist ein Logbuch im Web – ein Web-Tagebuch, in das man Beiträge aller Art hineinschreiben kann. Ein Weblog (kurz Blog) besteht im Grunde aus ganz „normalen“ Webseiten, die im Gegensatz zu den gewohnten statischen HTML-Seiten bei jedem Aufruf dynamisch aus einer Datenbank erzeugt werden. Abhängig von den eingebauten dynamischen Elementen kann deshalb eine Blogseite um 12 Uhr 51 anders aussehen als um 12 Uhr 50, ohne dass der Blog-Autor (kurz: Blogger) irgendetwas gemacht hätte (wie z.B. einen neuen Beitrag zu schreiben).

Der Name „Shablog“ ist abgeleitet von meinem Vornamen „Shayan“ und das Wort „Blog“.

Um die Interesse während des Projekts nicht zu verlieren habe ich mir das Thema genommen um grob meinen eigenen Blog zu bauen und nach und nach ihn erweitern und weitere Funktionen hinzufügen.

1.2 Handlung

In meinem Weblog darf der User sich an bzw. abmelden, sich registrieren, Bilder hochladen bzw. anschauen, Tagesblog(Satz des Tages) lesen und sein Kommentar(Blog) aktualisieren.

Der Admin darf noch dazu Tagesblog aktualisieren, von den Users, Username, Kennwort und Emailadresse ändern.

Auf jeder Seite befindet sich die Navigationsbar, die dazu dient den User zu anderen Seiten zu verlinken.

Ebenfalls befinden sich Header und Footer, die als Microservice programmiert wurden, auf jeder Seite.

1.2.1 An-Abmeldung

Auf dieser Seite darf der User sich an bzw. abmelden. Sollte er kein Account besitzen wird er durch dem Registrierungslink zu der Registrierungsseite landen. Nach der korrekten Eingabe der verlangende Informationen wird sein Account auf der Datenbank erstellt und gleichzeitig seine Infos auf eine Datei namens Datensicherung.txt gesichert dann wird der User auf der Log-In-Seite weitergeleitet um sich mit seinem Account anzumelden. Ist ein User eingeloggt ändert sich die Log_In Taste zu Log_Out.

Sollte die Anmeldung fehlerfrei erfolgen wird der User mit einer Begrüßung zu index.php (Home) weitergeleitet.

1.2.2 Home

Wird die Seite zum ersten Mal aufgerufen, erscheint ein JavaScript-Alert zur Begrüßen zu der Seite. Es wird dann direkt durch „session“ ausgeschaltet damit der User nicht jedes Mal wenn er die Home-Seite aufruft eine Begrüßung-Nachricht bekommt.

Ist man angemeldet, erscheint dann eine Begrüßung zu der angemeldeten Person und eine Tabelle in dem der Username und der dazugehöriger Kommentar des Users.

Allerdings wird der Satz des Tages, der durch Admin aktualisiert wird in der Mitte der Seite dargestellt.

1.2.3 Einstellung

Nach Aufruf die Einstellung-Seite wird zuerst die Log-In-Status durch SESSION[„username“] überprüft. Sollte der User nicht angemeldet sein erscheint ein JavaScript-Alert, das dem User die Wahl zwischen Anmelden und zurück zum Home-Screen freilässt.

Ist der User hingegen angemeldet folgt der nächste Test um zu checken ob die angemeldete Person der Administrator oder ein normaler User entspricht.

Im Falle des Users, wird eine Tabelle mit allen Informationen über den User angezeigt und der User darf sein Kommentar aktualisieren.

Im Falle des Administrators, erscheint eine Tabelle mit allen registrierenden und der Admin ist bevollmächtigt von jedem User Username, Kennwort und die Emailadresse zu ändern.

1.2.4 Photos/Videos

Auf dieser Seite werden die hochgeladene Bilder in einer Tabelle dargestellt. Die Bilder können auf der Home-Seite hochgeladen werden. Sollte „upload.php“ keine Fehler melden landet das hochgeladene Bild in dem Upload-datei und mit Aufruf der photos.php werden die Bilder von diesem Datei geholt.

Zur Zeit, ist Videos-function nicht fertiggestellt. Es können jedoch in Zukunft noch mehrere Funktionen hinzugefügt werden, die den Users und Admin weitere wünsche erfüllen.

2. Installationsanleitung

.Ein Web host mit PHP und MySQL Unterstützung.

.Installieren eines Webserver auf dem eigenem PC, und dann installieren von PHP und MySQL.

Zur Nutzung des Localhostes reicht ein Webserver mit Unterstützung von PHP und MySQL aus. Ich habe „XAMPP“ benutzt. XAMPP ist die beliebteste PHP-Entwicklungsumgebung. Es ist eine vollständig kostenlose, leicht zu installierende Apache-Distribution, die MariaDB, PHP und Perl enthält. Das XAMPP Open-Source-Paket wurde für eine extrem einfache Installation und Nutzung eingerichtet

Nach der Installation von XAMPP samt Apache und MySQL öffnet man das XAMPP-Verzeichnis und dann htdocs-Verzeichnis. Fügt man den Code in dem Verzeichnis ein, kann man ihn in dem Browser mit dem Link „localhost/index.php“ aufrufen.

Damit man auf localhost zugreifen kann soll zuerst der Apache-Server auf'm Xampp gestartet werden.

Damit man auf die Datenbank zugreifen kann soll der MySQL-Server ebenfalls gestartet werden.

Mit dem Link „http://localhost/phpmyadmin/“ landet man auf die Datenbank.

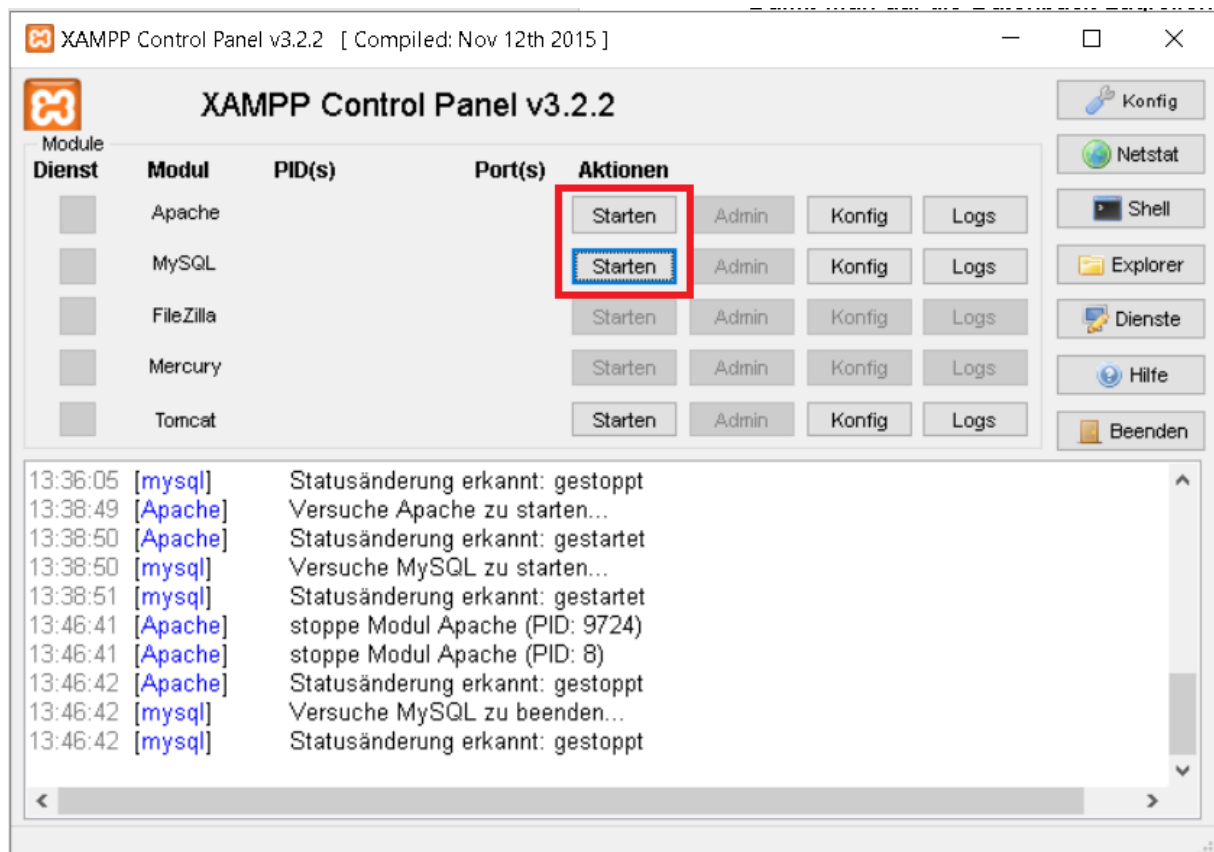


Abbildung 1 : XAMPP Control Panel v3.2.2

3. Bedienungsanleitung

Beim Aufruf der Seite wird man durch einem JavaScript-Alert begrüßt. Auf der Home-Seite befinden sich außer Header, Footer und die Navigationsbar, der Satz des Tages und eine Funktion um Bilder hochzuladen. Ist man angemeldet, erscheint ein Willkommen heißen für die Person und eine Tabelle mit dem Username und der Kommentar der angemeldeten Person.

Auf dem Header ist der Datum und die aktuelle Zeit zu sehen, Während auf dem Footer Copyright Zeichen mit dem Anfangsjahr bis zum aktuellen Jahr zu sehen ist.

Außerdem ist das Thema jeder Seite auf dem Header zu sehen, und sieht nicht für jede Seite gleich aus.

ShaBlog

Today is 2017/01/06
The time is 04:06

Home
An-Abmeldung
Einstellung
Photos/Videos

Satz des Tages

Wird schon werden!

Willkommen shayan !

username	kommentar
shayan	Wer nichts wird, wird Wirt!

Bilder Hochladen:

©2016-2017 -Projekt1

Abbildung 2 : Home

Um sich anzumelden muss man auf dem An-Abmeldungslink drücken. Besitzt man kein Account wird man durch dem Registrierungslink zu der Anmeldeformular weitergeleitet.

klick hier!'. The right screenshot is the 'Anmeldeformular' (registration form), showing fields for 'Username', 'E-mail', 'Kennwort', 'Kennwort(Wiederholung)', 'Website', 'Testkommentar', and 'Kommentar'. It also has radio buttons for 'geschlecht' (male/female) and an 'Absenden' button. A red arrow points from the 'klick hier!' link on the login page to the registration form."/>

Log In

Today is 2017/01/06
The time is 04:17

* Pflichtfeld.

Username: *

Kennwort: *

noch nicht registriert? [klick hier!](#)

©2016-2017 -Projekt1

Anmeldeformular

Today is 2017/01/06
The time is 04:15

* Pflichtfeld.

Username: *

E-mail: *

Kennwort: *

Kennwort(Wiederholung): *

Website:

Testkommentar:

Kommentar:

geschlecht: ☒ weiblich ☐ männlich *

Abbildung 3 : An / Abmeldung

Nach der erfolgreichen Registrierung landet man wieder auf die Log_In-Seite und darf sich einloggen.

Um auf die Einstellung zugreifen zu dürfen, muss man eingeloggt sein. Die Einstellung sieht für User und Admin unterschiedlich aus. Der User sieht alle seine Infos darf jedoch nur seinen Kommentar aktualisieren. Der Admin dagegen hat zugriff auf alle angemeldeten Personen und darf von jedem Username, Kennwort und die Emailadresse ändern.

Einstellung

Today is 2017/01/06
The time is 05:03

id	username	email	kennwort	website	kommentar	geschlecht	reg_date
1	Admin	admin@administrator.com	pass	admin-Shablog.com	administrator	männlich	2016-12-20 19:09:37
3	as	as	as		hey everybody:)	weiblich	2017-01-04 16:07:57
7	assss	ass	ssss		kkkkk	männlich	2016-12-20 19:42:04
9	assss	ass	ssss		kkkkk	männlich	2016-12-20 19:45:01
10	mei	nnne	nnn		neues Kommentar	weiblich	2017-01-03 23:51:40
11	shay	an	ij			männlich	2016-12-23 18:36:05
30	kigkh	klk	kkk		kiki	weiblich	2016-12-23 19:28:07
31	michael	fidfd	hhh			männlich	2017-01-02 15:12:09
34	gg	ff	hh		fg	männlich	2017-01-02 14:46:03
35	shayan	shayan2013@hotmail.de	keun	www.yahoo.com	Wer nichts wird, wird Wirt!	männlich	2017-01-06 16:06:42

id: *

username: email: kennwort:

[Home](#)
[An-Abmeldung](#)
[Einstellung](#)
[Photos/Videos](#)

©2016-2017 -Projekt1

Abbildung 4 : Einstellung - Admin-Ansicht

Darum sieht die Einstellung-Seite für Users und Admin ganz unterschiedlich aus.

Ein weiterer Aspekt ist der Photos/Videos-Link, der dem User ermöglicht Bilder hochzuladen und sich alle hochgeladene Bilder anzuschauen. Zurzeit ist die Video-Funktion ausgeschlossen.

Die Bilder können auf der Index-Seite durchsucht und hochgeladen werden. Ist das Bild kleiner als 500000 Bytes und eine von begrenzten Formaten (JPG, JPEG, PNG & GIF) landet das in dem Upload-Verzeichnis. Sollte das Bild schon existieren, oder ein Fehler währenddessen auftreten wird der User mit einer Warnmeldung informiert.

4. Systemarchitektur

In dem Hauptverzeichnis befinden sich :

Hauptklassen : index.php, einstellung.php, log_In.php, photos.php und register.php

Tagesblog (was durch Admin aktualisiert wird)

Und vier andere Verzeichnisse :

- **Design** besteht in : style.css
- **Dokumentation** besteht aus : ProjektADoku.pdf und Datensicherung
- **Uploads** : besteht aus hochgeladene Bilder
- **Microservices_de** besteht aus : dbConfig.php, fileCreator.php, headerFooter.php, imageConfig.php, meta.php, navigation.php, session.php und upload.php

Als Systemarchitektur nahm ich „Microservices“. Microservices sind ein Architekturmuster der Informationstechnik, bei dem komplexe Anwendungssoftware aus kleinen, unabhängigen

Prozessen komponiert wird, die untereinander mit sprachunabhängigen Programmierschnittstellen kommunizieren. Die Dienste sind klein, weitgehend entkoppelt und erledigen eine kleine Aufgabe. So ermöglichen sie einen modularen Aufbau von Anwendungssoftware.

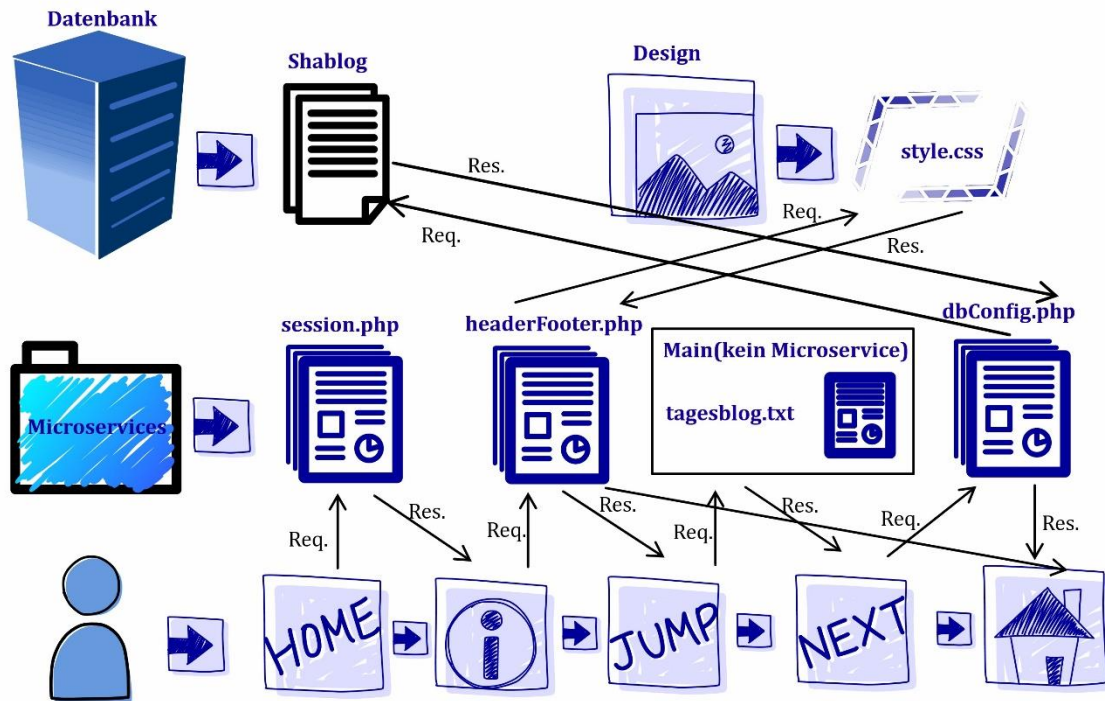


Abbildung 5 : Aufrufprozess von index.php

So funktioniert der Aufruf der Home-Seite zum ersten Mal.

Schritt 1 : session.php wird gerufen um zu wissen ob die Seite zum ersten Mal aufgerufen wird, oder ob der User angemeldet ist und wenn ja ist SESSION[„uname“] gesetzt.

Schritt 2 : headerFooter.php wird gerufen und damit der Header eingesetzt wird und später der Footer.

Schritt 3 : headerFooter.php ruft style.css von dem Designverzeichnis und holt sich vorgeschriebenes Style aus dieser Klasse.

Schritt 4 : Als nächstes wird der Satz des Tages von tagesblog.txt geholt und nach dem Header eingefügt.

Schritt 5 : Ist SESSION[„uname“] gesetzt wird dbConfig.php gerufen. Diese Klasse holt sich die Infos der angemeldeten Person von der Datenbank und gibt die an index.php weiter.

Schritt 6 : Zuletzt wird headerFooter.php nochmal aufgerufen um Footer einzusetzen.

Das ganze geschieht nur beim Aufruf von index.php

5. Beschreibung eines technischen Teilaspektes

Microservices_de/dbConfig.php>>updateData(\$uname, \$eMail, \$kenn, \$id) :

Durch die Funktion updateData darf der Admin Username, Email und Kennwort von einem User ändern.

Ich habe die Funktion so geschrieben dass der Admin freie Hand hat. Das heißt er darf Username, Email oder Kennwort einzeln ändern oder zwei davon gleichzeitig oder alle drei Variablen auf einmal einsetzen und aktualisieren lassen.

Das ganze funktioniert so :

.Mit der Aufruf von Update-Taste, wird erstmal geprüft ob die ID eingegeben ist.

.Dann werden die Variablen nach den Regeln geprüft.

.Ist der Username nicht leer

.Besteht er nur aus Buchstaben oder Leertaste => eingesetzt

.Ist das Email nicht leer

.Entspricht es das richtige Format => eingesetzt

.Ist das Kennwort leer => eingesetzt

Ist eine oder mehrere Variablen eingesetzt, wird die updateData-Funktion aufgerufen.

Für die nicht eingesetzte Variablen wird empty(„“) eingesetzt.

In der updateData-Funktion wird erstmal geprüft ob oder welche der Variablen leer sind.

Für die leeren Variablen wird die fetch_assoc()-Funktion gerufen und sie füllt die gefehlten Variablen aus.

```
$sql = "SELECT username, email, kennwort FROM users WHERE id = " . $id ;
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        if (empty($uname)){
            $uname = $row["username"];
        }
        if (empty($eMail)){
            $eMail = $row["email"];
        }
        if (empty($kenn)){
            $kenn = $row["kennwort"];
        }
    }
}
```

```
}
```

Erst danach wird der SQL-Befehl ersetzt und die Update-Funktion() ausgeführt.

6. Fazit, Auswertung

Das Bauen des Weblogs hat mir sehr viel Spaß gemacht. Da ich schon etwas Erfahrungen mit Microservices hatte, wusste ich direkt, welche Architektur ich für solch ein Projekt vornehme. Ich konnte vieles an technische Erfahrungen mitnehmen. Man hat eine Menge dazu gelernt, wie zum Beispiel den Umgang mit dem PhpMyAdmin und auch die Kenntnisse von verschiedenen Vorgehensmodelle des Projekts erweitert. Das Projekt hat mir auch gezeigt, wie wichtig fachliche Anforderungen, Zeitplan und Sprint Backlog ist. Das Schreiben und Korrigieren des Codes, sowie die Nachbearbeitung des Konzepts haben viel mehr Stunden in Anspruch genommen, als ursprünglich gedacht. Dadurch kam es zu zeitlichen Engpässen und ich bin vom Zeitplan abgekommen. In Zukunft muss ich das Zeitmanagement besser planen und darf den Umfang nicht unterschätzen

7. Quellen

7.1 Links

<https://de.wikipedia.org/wiki/Microservices> (Aufgerufen 06.01.2017)

<http://www.w3schools.com/>

<https://secure.php.net/>

<https://www.youtube.com/watch?v=f-ah0tIOW1o&t=851s>

<https://www.dasscrumteam.com/scrum>

7.2 Abbildung

Abbildung 6 : XAMPP Control Panel v3.2.2

Abbildung 7 : Home (index.php) : angemeldet als Shayan

Abbildung 8 : An / Abmeldung (rechts : log_In.php, links : register.php)

Abbildung 9 : Einstellung - Admin-Ansicht (eingeloggt als Admin)

Abbildung 10 : Aufrufprozess von index.php (Diagramm selbst erstellt mit „MetaMoji Note“)