**Shablog (privates Weblog)**

Shayan Shamseddin (2103763)

Betreuer: Prof. Dr. Andreas Plaß

05. 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](#_Toc471571329)

[1.1 Idee 3](#_Toc471571330)

[1.2 Handlung 3](#_Toc471571331)

[1.2.1 An-Abmeldung 3](#_Toc471571332)

[1.2.2 Home 3](#_Toc471571333)

[1.2.3 Einstellung 3](#_Toc471571334)

[1.2.4 Photos/Videos 4](#_Toc471571335)

[2. Installationsanleitung 4](#_Toc471571336)

[3. Bedienungsanleitung 5](#_Toc471571337)

[4. Systemarchitektur 7](#_Toc471571338)

[5. Beschreibung eines technischen Teilaspektes 8](#_Toc471571339)

[6. Fazit, Auswertung 9](#_Toc471571340)

# 

# 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“.

## 1.2 Handlung

In meinem Weblog darf der User sich an/abmelden, sich regiestieren, Bilder hochladen bzw. anschauen, Tagesblog 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 dem User zu anderen Seiten zu verlinken.

### 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 erstellt und der User wird auf der Log-In-Seite weitergeleitet um sich mit seinem Account anzumelden.

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 ein Begrüßung 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 dargestellt.

### 1.2.3 Einstellung

Nach Aufruf die Einstellung-Seite wird zuerst die Log-In-Status 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 regiestierenden 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 hochgeladenes 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 Webservers 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.

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 localhost zugreifen kann soll zuerst der Apache-Server auf’m Xampp gestartet werden.

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

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

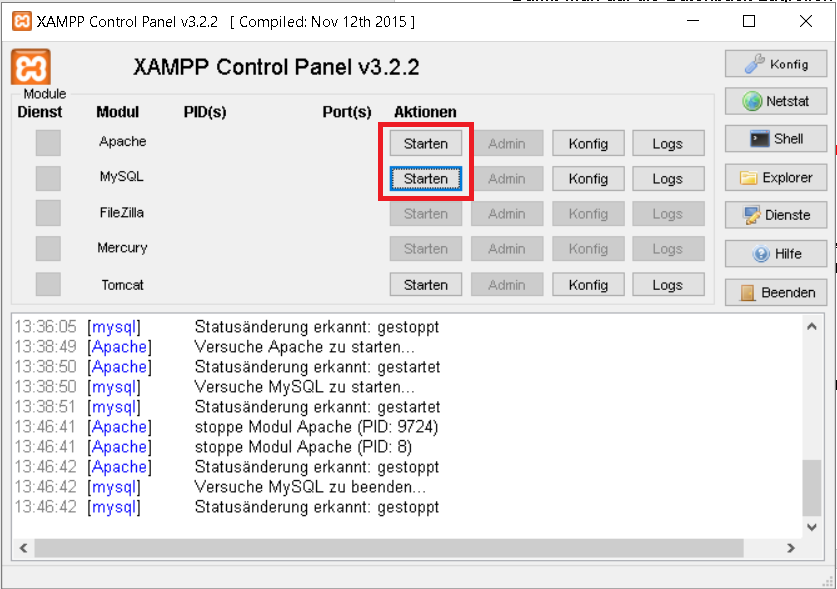


Abbildung 1 : XAMPP Control Panel v3.2.2

# 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.



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.

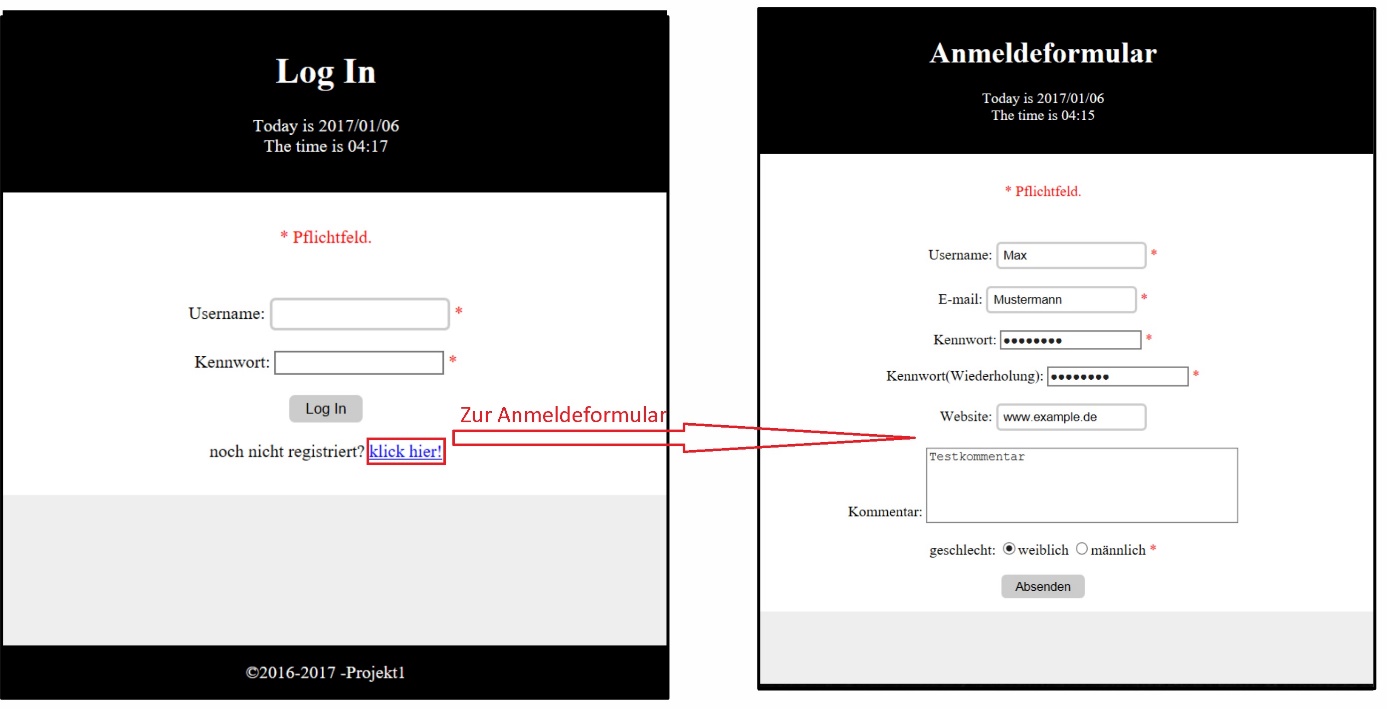


Abbildung 3 : An / Abmeldung

Nach der erfolgreichen Registrierung landet man auf die Log\_In-Seite und darf sich einloggen.

Um auf die Einstellung zuzugreifen 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.

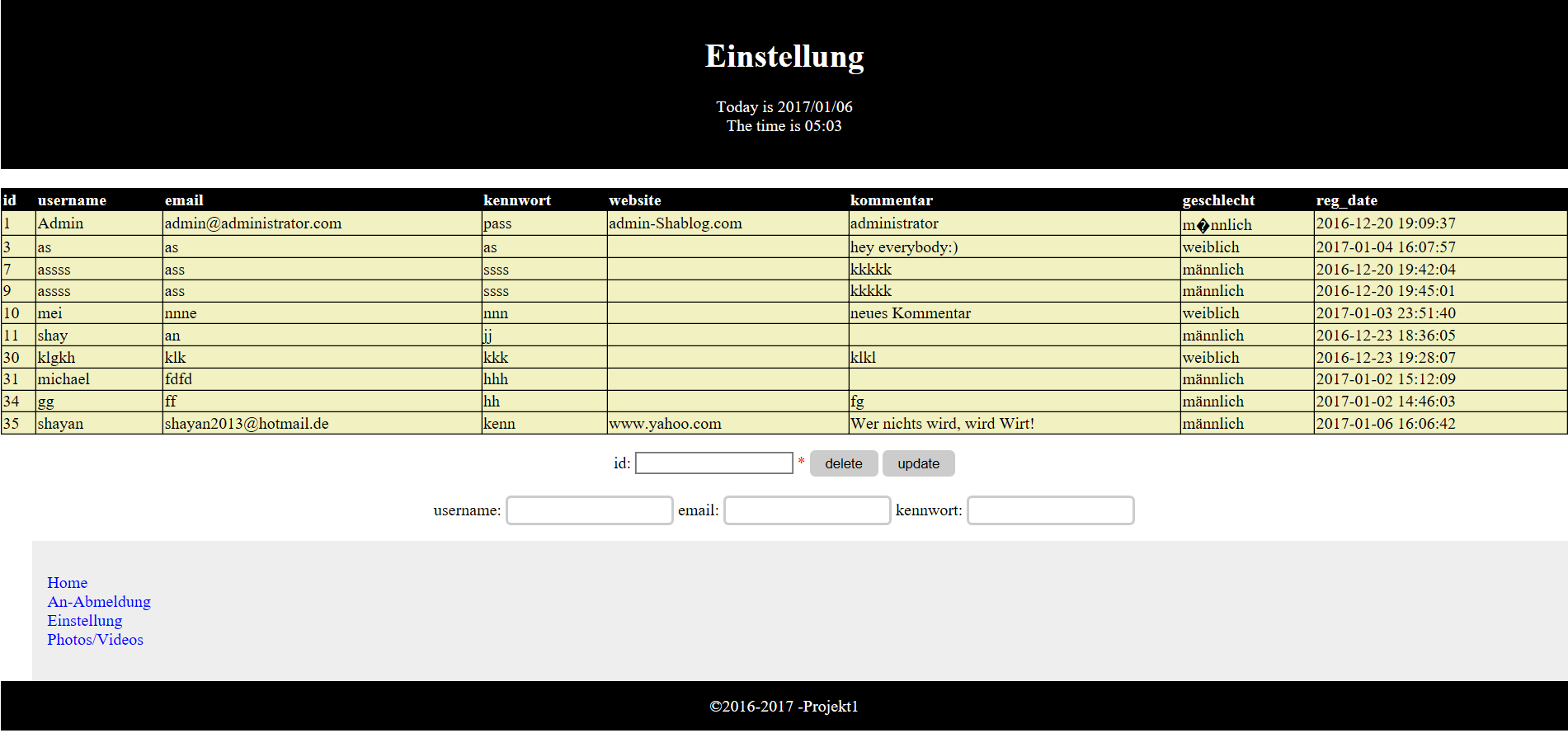


Abbildung 4 : Einstellung - Admin-Ansicht

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

Als Systemarchitektur nahm ich „Microservices“. Microservices sind ein [Architekturmuster](https://de.wikipedia.org/wiki/Architekturmuster) der [Informationstechnik](https://de.wikipedia.org/wiki/Informationstechnik), bei dem komplexe [Anwendungssoftware](https://de.wikipedia.org/wiki/Anwendungssoftware) aus kleinen, unabhängigen [Prozessen](https://de.wikipedia.org/wiki/Prozess_(Informatik)) komponiert wird, die untereinander mit sprachunabhängigen [Programmierschnittstellen](https://de.wikipedia.org/wiki/Programmierschnittstelle) kommunizieren. Die [Dienste](https://de.wikipedia.org/wiki/Dienst_(Informatik)) sind klein, weitgehend [entkoppelt](https://de.wikipedia.org/wiki/Kopplung_(Softwareentwicklung)) und erledigen eine kleine Aufgabe. So ermöglichen sie einen [modularen](https://de.wikipedia.org/wiki/Modularit%C3%A4t) Aufbau von Anwendungssoftware.

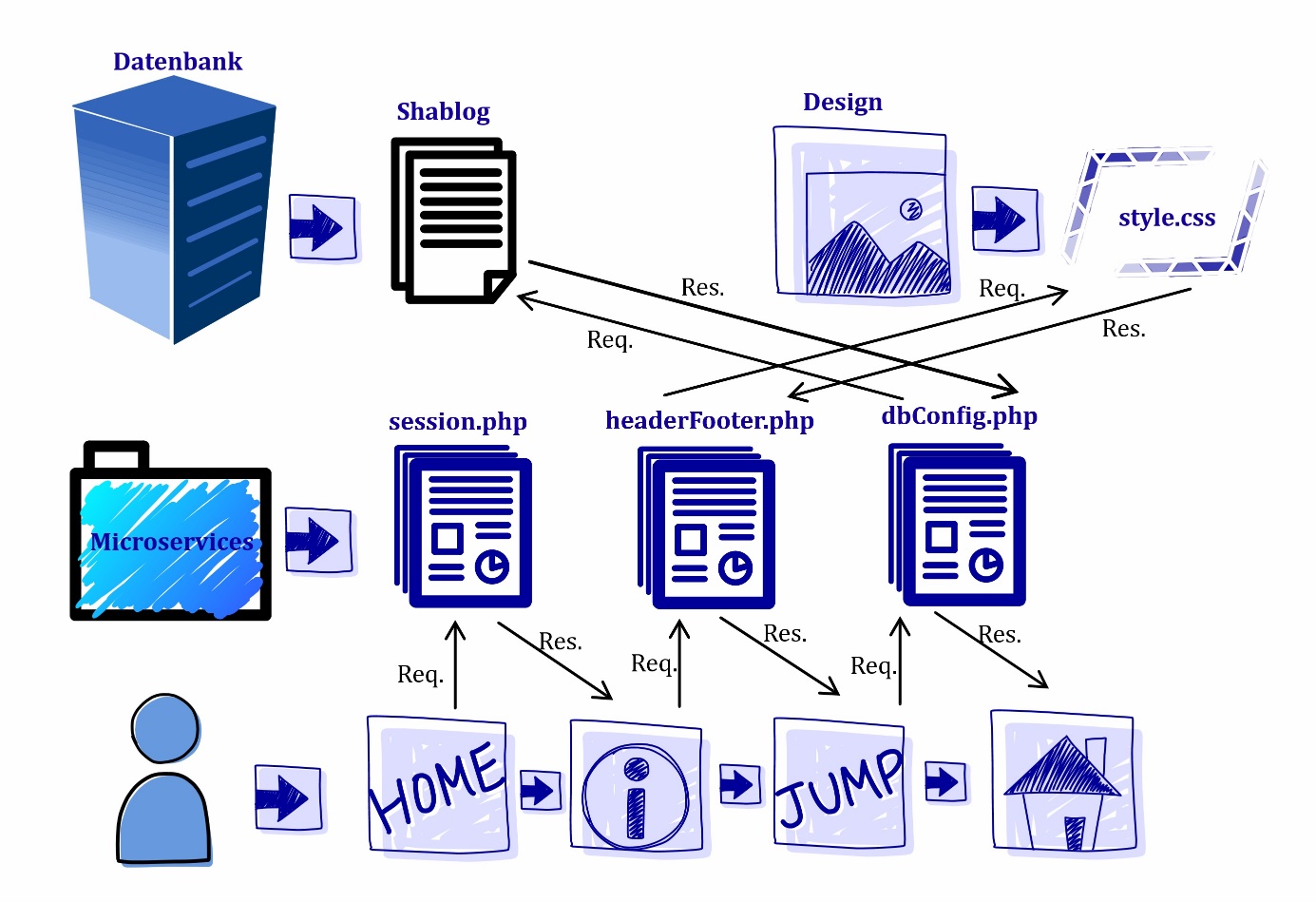


Abbildung 5 : Aufrufprozess von index.php

# 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