Werbedisplay Steuercenter

Brainstorming

Joël julien Müller

2023

Inhalt

[Vorwort 1](#_Toc131077158)

[Anforderungen 1](#_Toc131077159)

[Voraussetzungen 1](#_Toc131077160)

[Brainstorming 1](#_Toc131077161)

[Funktionen und aussehen 2](#_Toc131077162)

[Backend Programmiersprache 2](#_Toc131077163)

[**Advantages of PHP** 2](#_Toc131077164)

[**Disadvantages of PHP** 2](#_Toc131077165)

[**Advantages of Node.js** 3](#_Toc131077166)

[**Disadvantages of Node.js** 3](#_Toc131077167)

[Frontend Programmiersprache 3](#_Toc131077168)

[Webserver 4](#_Toc131077169)

# Vorwort

In der letzten Besprechung hat mir Adrian Bähler die Anforderungen für eine Website, wo man die Werbedisplays steuern kann, gezeigt. Meine Aufgabe ist es nun, ein Brainstorming zu machen und das dann mit Adrian zu besprechen.

# Anforderungen

Auf der Website sollte man Filialen erstellen können, bei der Filiale einstellen, wie viele Monitore dort vorhanden sind und auswählen, auf welchem Monitor welche Bilder, in Welcher Reihenfolge dargestellt werden sollten. Die Website sollte Multi User fähig sein, einfach zu bedienen und ansprechend aussehen.

# Voraussetzungen

Optimalerweise sollte die Webapplikation auf XAMPP lauffähig sein. XAMPP stellt bereits Apache und MySQL (bzw. MariaDB seit Oktober 2015) zur Verfügung. Daher ist es naheliegend PHP als Programmiersprache zu verwenden. Adrian und Christan haben damit auch schon Erfahrung.

Die Website sollte durch die Konfiguration des Benutzers eine Konfigurationsdatei erstellen, die von den Raspberry Pis über FTP (Cronjob) heruntergeladen wird. Mit der Information, der Konfigurationsdatei, weiss der PC dann, welche Bilder er herunterladen muss. Diese werden dann in ein Verzeichnis kopiert und mit der Anzeigesoftware dargestellt.

# Brainstorming

Um eine bessere Vorstellung zu bekommen, wie die Website aussehen sollte, habe ich eine Zeichnung gemacht (Siehe blueprint.drawio).

## Funktionen und aussehen

Die Website sollte ein Dashboard haben, wo man Files uploaden kann (per drag and drop oder mit dem Explorer). Des Weiteren sollte sie eine Matrix haben, mit den Filialen als Spalten und Monitoren als unterspalten.

Als Zeilen sollten die Dateien, mit dem Dateinamen und einem kleinen Bild als Vorschau angezeigt werden.

Wenn man neue Dateien hochgeladen hat, sollten diese zuoberst angezeigt werden und markiert (z.b. mit einem blauen Punkt).

In der Matrix kann man dann Zahlen eingeben, in welcher Reihenfolge die Bilder angezeigt werden sollte.

Bei den Bildern sollte es eine kleines Suchfeld haben, zum Suchen eines Bildes und eine Filterfunktion (z.b. Nach Upload Datum, A-Z ).

Die Website sollte des Weiteren einen Menu punkt Settings haben, wo man die Anzahl Monitore einer Filiale ändern kann und eine neue Filiale erfassen, sowie eine bereits bestehende Filiale löschen kann.

Eine Login Funktion würde ebenfalls Sinn machen. Je nach dem könnte man das nur mit Microsofts Website Login machen und dann eine Gruppe erstellen, deren Mitglieder zugriff haben, oder selbst ein Login bauen, mit einer «Forgot Pasword» Funktion.

## Backend Programmiersprache

Es macht Sinn die Website mit PHP zu bauen, da Adrian und Christian Erfahrung mit dieser Programmiersprache haben.

Man könnte die Website aber auch mit Node.JS (Server Side JavaScript) machen.

Hier eine kleine Auflistung der vor und Nachteile von PHP und Node.JS:

### **Advantages of PHP**

* Several built-in security features
* Regularly updated with new features and improvements
* Robust support for CMSes
* Strong support for web development frameworks, web servers, operating systems, and hosting platforms
* A wide range of web development tools and libraries

### **Disadvantages of PHP**

* Less flexible and dynamic than other languages, such as Java or C++
* Slower performance for certain types of applications
* Not ideal for real-time applications or parallel programming
* Global namespace makes it more challenging to manage and organize large PHP codebases
* Relatively simple object-oriented model, which can make it less powerful and flexible than other languages like Java and C++

### **Advantages of Node.js**

* Efficient performance
* Dynamic and flexible for concurrent and parallel programming
* Front-end JavaScript developers can easily pick up Node.js to build their back ends, making them more productive in a minimal amount of time
* Built-in support for websockets for building real-time applications
* Well suited for building microservices and serverless functions

### **Disadvantages of Node.js**

* Not ideal for web applications that require the rendering of sophisticated graphics
* Fewer built-in security features
* Less compatibility with older or legacy systems
* Lacks robust support for CMSes

Quelle : [https://blog.hubspot.com/website/php-vs-node-js#:~:text=PHP%20is%20a%20general%2Dpurpose,and%20efficient%20back%2Dend%20applications](https://blog.hubspot.com/website/php-vs-node-js" \l ":~:text=PHP%20is%20a%20general%2Dpurpose,and%20efficient%20back%2Dend%20applications).

Ich habe beide Programmiersprachen schon einmal ausprobiert. Auf freeCodeCamp habe ich das JavaScript tutorial gemacht und eine kleine Website mit Node.JS gemacht, wo man in real time einen Knopf drücken kann, der eine Zahl erhöht.

Mit PHP habe ich ein änliches tutorial gemacht und dabei eine Login funktion mit SQL und LDAP gemacht.

Node.JS wäre sehr spannend, jedoch bietet PHP in diesem Fall mehr Vorteile.

Je nach dem, wie stabil die Anwendung sein muss, würde es vielleicht noch Sinn machen ein Framework wie Laravel zu verwenden. Ich habe das Framework einmal kurz angeschaut.

## Frontend Programmiersprache

Für das Frontend würde ich Bootstrap5 oder Talwind CSS verwenden. Mit beiden Frontend Frameworks kann man responsive Frontends machen. Tailwind CSS bietet mehr Freiheit im Design als Bootstrap.

Mit Bootstrap habe ich jedoch mehr Erfahrung als mit Tailwind CSS. Bootstrap5 kann man relativ einfach über CDN in das Projekt miteinbinden.

## Webserver

Für den Webserver würde das bereits vorhandene XAMPP oder Docker verwenden. XAMPP haben wir schon. Jedoch sollte man XAMPP eigentlich nie als Produktiv Server verwenden.

Eine Docker Umgebung (Apache, PHP, PHPmyAdmin, MariaDB) könnte man mit Docker-compose machen. Ich habe das zum Ausprobieren schon einmal zuhause gemacht.