

STATE OF THE ART



REFUNDABLE

effiziente Reise- und Exkursions-
verwaltung für Schulen

Dehner Linus, Foster Ryan, Beier Michael

tgm
Die Schule der Technik

JUST DO IT
HÖHERE ABTEILUNG FÜR
INFORMATIONSTECHNOLOGIE

Version	Autor	QS	Datum	Status	Kommentare
0.1	Idehner	mbeier	2020-09-24	Draft	Create
1	mbeier	Idehner	2020-10-24	Draft	Backend - Überblick
2	mbeier	Idehner	2020-11-08	Draft	Layout finalisiert
3	mbeier	Idehner	2020-11-08	Draft	Backend - Docker
4	Idehner	mbeier	2020-11-09	Draft	Webdesign - Überblick

Inhaltsverzeichnis

1	Projektleitung & Frontend - responsives Webdesign	4
1.1	Überblick	4
1.2	HTML, CSS, JS	4
1.2.1	HTML	4
1.2.2	Vanilla CSS	6
1.2.3	JS	6
1.3	SASS	6
1.4	CSS Frameworks	6
1.4.1	Bootstrap	6
1.4.2	Materialize	6
1.4.3	ZURB Foundation	6
1.4.4	Tailwind CSS	6
1.5	Vergleich	6
1.6	Zielgruppenorientiertes Design	6
1.7	Fragestellungen	6
2	Frontend - Webapplikation als REST-Client	7
2.1	Überblick	7
2.2	Design-Patterns	7
2.2.1	MVVM	7
2.2.2	MVC	7
2.2.3	Vergleich	7
2.3	Datenformate	7
2.4	Umsetzungsmöglichkeiten	7
2.4.1	Vue	7
2.4.2	React	7
2.4.3	Angular	7
2.4.4	Ohne Framework	7
2.4.5	Vergleich	7
2.5	Aufbereitung der Daten	7
2.6	Fragestellung	7
3	Backend - REST-Schnittstelle und Infrastruktur	8
3.1	Überblick	8
3.2	Docker	9
3.2.1	Datenbank	9
3.2.2	Backend-Container	9
3.2.3	Webserver	9
3.3	Deployment	9
3.4	REST-Schnittstelle	9
3.4.1	Framework	9
3.4.2	Endpoints	9

3.5	Funktionalität	9
3.5.1	TGM-LDAP Schnittstelle	9
3.5.2	Datenbank Schnittstelle	9
3.5.3	Google Maps	9
3.5.4	WebUntis	9
3.5.5	E-Mails	9
3.5.6	PDF-Dateien	9
3.6	Kommunikation und Datenformate	9
4	Fazit	10
	Abbildungsverzeichnis	11

1 Projektleitung & Frontend - responsives Webdesign

1.1 Überblick

In diesem Kapitel werden die Anforderungen des Designs des **Frontends** geschildert. Da es mehrere Möglichkeiten gibt das Frontend zu realisieren werden hier drei wesentliche Methoden verglichen. Die Variante **HTML**, **CSS** und **JavaScript** zu verwenden. Die zweite Methode die zum Vergleich hergezogen wird ist statt **Vanilla CSS** **SASS** zu verwenden. Die dritte und auch letzte Methode in diesem Vergleich ist, dass die Arbeit durch die Verwendung eines **CSS Frameworks** vereinfacht wird. Anschließend werden die drei verschiedenen Methoden verglichen und die, die am Besten für unser Projekt Refundable passt ausgewählt. Zu guter Letzt wird das Design für die Zielgruppe analysiert, da Refundable explizit für Lehrer codiert wird und diese sich möglichst gut und schnell auf der Website zurecht finden sollen.

1.2 HTML, CSS, JS

Der eigentliche Standard HTML5 wird in der Praxis meist als Überbegriff für HTML, CSS und JS verwendet. In diesem Unterkapiteln beschäftigen wir uns genau damit. **[html5-css3-handbuch]** In den folgenden Kapiteln wird erklärt wozu HTML, CSS und JS da sind und welche Funktionalitäten sie bieten.

1.2.1 HTML

HTML ist eine **Auszeichnungssprache** steht für *Hypertext Markup Language* und wurde 1989 von dem britischen Informatiker Tim Burners-Kee veröffentlicht.

„Hypertext bezeichnet die Möglichkeit, Texte mit Hilfe von Hyperlinks oder kurz Links miteinander zu verbinden.“ **[html5-css3-def]**

Dies heißt, dass man mittels Hypertexts auf der Seite beliebig Herumspringen kann. Auszeichnungssprachen werden im Fachjargon auch als Markup Language (ML) bezeichnet. Markup Languages werden in zwei verschiedene Gruppen aufgeteilt, zum einen Procedural Markup Languages (PML), das sind jene Auszeichnungssprachen, die für die Verarbeitung von Daten optimiert sind. Zum anderen Descriptive Markup Languages (DML), diese sind für die logische Strukturierung von Daten da.

Bekannte Beispiele hierfür wären:

- PML
 - PDF
 - TeX
- DML
 - HTML
 - SVG

HTML ist hauptsächlich dafür da, um Texte, Grafiken und Hyperlinks (Links) darzustellen. Die Bearbeitung von HTML-Dokumenten ist relativ einfach und unkompliziert, da es eine reine textbasierte Sprache ist und mit jedem Texteditor bearbeitet werden kann. Allerdings ist HTML nicht mit einer Programmiersprache zu verwechseln, da nur **Tags** und keine Befehle oder Anweisungen verwendet werden. Solche Tags können wie folgt aussehen:

```
1      <tagname>Tag Inhalt</tagname>
2      <einzeltag attribut="123">
```

Das grundlegende Gerüst von HTML besteht aus einer Deklaration von HTML, einem *html*-, *head*- und *body*-Tag. In den *html*-Tag kommt ein *title*-Tag, in diesem wird der Titel der Website angegeben und ein *meta*-tag, in diesem werden Meta-Informationen angegeben. In den *Body*-Tag kommen wiederum Tags, die den Inhalt der Seite beinhalten.

```
1      <!DOCTYPE html>
2      <html lang="de">
3          <head>
4              <meta charset="UTF-8">
5              <meta name="viewport" content="width=device-width,
6                  ↪ initial-scale=1.0">
7              <title>Kurzes BSP</title>
8          </head>
9          <body>
10             <h1>Überschrift</h1>
11             <button>Drück mich!</button>
12         </body>
13     </html>
```

Wenn man die Datei nun im Browser öffnet schaut dies wie folgt aus:

Überschrift

Drück mich!

Abbildung 1: Beispiel einer HTML Seite mit einer Überschrift und einem Button

Da HTML nur für die Grundstruktur einer Website gedacht ist, also zum beschreiben der Struktur und des Inhalts schaut die Seite noch nicht sonderlich ansprechend aus. Um das zu verändern ist CSS benötigt. [\[auszeichnungssprachen\]](#) [\[html5-css3-handbuch\]](#) [\[html5-css3-def\]](#)

1.2.2 Vanilla CSS

[Vanilla CSS](#)

1.2.3 JS

1.3 SASS

1.4 CSS Frameworks

[CSS Frameworks](#)

1.4.1 Bootstrap

1.4.2 Materialize

1.4.3 ZURB Foundation

1.4.4 Tailwind CSS

Tailwind [CSS](#)

1.5 Vergleich

1.6 Zielgruppenorientiertes Design

1.7 Fragestellungen

2 Frontend - Webapplikation als REST-Client

2.1 Überblick

Das [Backend](#) muss mit dem [Frontend](#) verbunden werden. Es gibt unterschiedliche Möglichkeiten dies zu realisieren. Beim Realisieren, muss darauf geachtet werden, dass eine Struktur vorhanden ist. Es werden 2 verschiedene [Design-Pattern](#) angeschaut und verglichen. Umgesetzt wird dann ein [Design-Pattern](#) mithilfe von [JavaScript](#). Hier kann ein [JavaScript Framework](#) zum Einsatz kommen. Dazu werden hier verschiedene [JavaScript Frameworks](#) angeschaut und verglichen. Für die Verarbeitung der Daten ist es wichtig Datenformate festzulegen. Die Aufbereitung der Elemente fürs [Frontend](#) mit den Daten des [Backends](#) wird ebenfalls angeschaut.

2.2 Design-Patterns

2.2.1 MVVM

2.2.2 MVC

2.2.3 Vergleich

2.3 Datenformate

2.4 Umsetzungsmöglichkeiten

2.4.1 Vue

2.4.2 React

2.4.3 Angular

2.4.4 Ohne Framework

2.4.5 Vergleich

2.5 Aufbereitung der Daten

2.6 Fragestellung

3 Backend - REST-Schnittstelle und Infrastruktur

3.1 Überblick

Das Backend besteht aus mehreren Komponenten. Einerseits muss eine gewisse Software-Infrastruktur aufgebaut werden, um [Webinterface](#) und die REST-Schnittstelle bereitzustellen. Andererseits muss die Anwendung selbst auch entwickelt werden. Diese besteht wiederum auch aus mehreren Teilen. Darunter fällt die REST-Schnittstelle, inklusive der implementierten Endpoints, selbst, Schnittstellen zu diversen Diensten, wie dem TGM-LDAP Server, zur Datenbank, zu Google Maps und zu WebUntis, aber auch die weitere Funktionalität der Anwendung, unter anderem das Versenden von E-Mails oder Erstellen von PDF-Dateien.

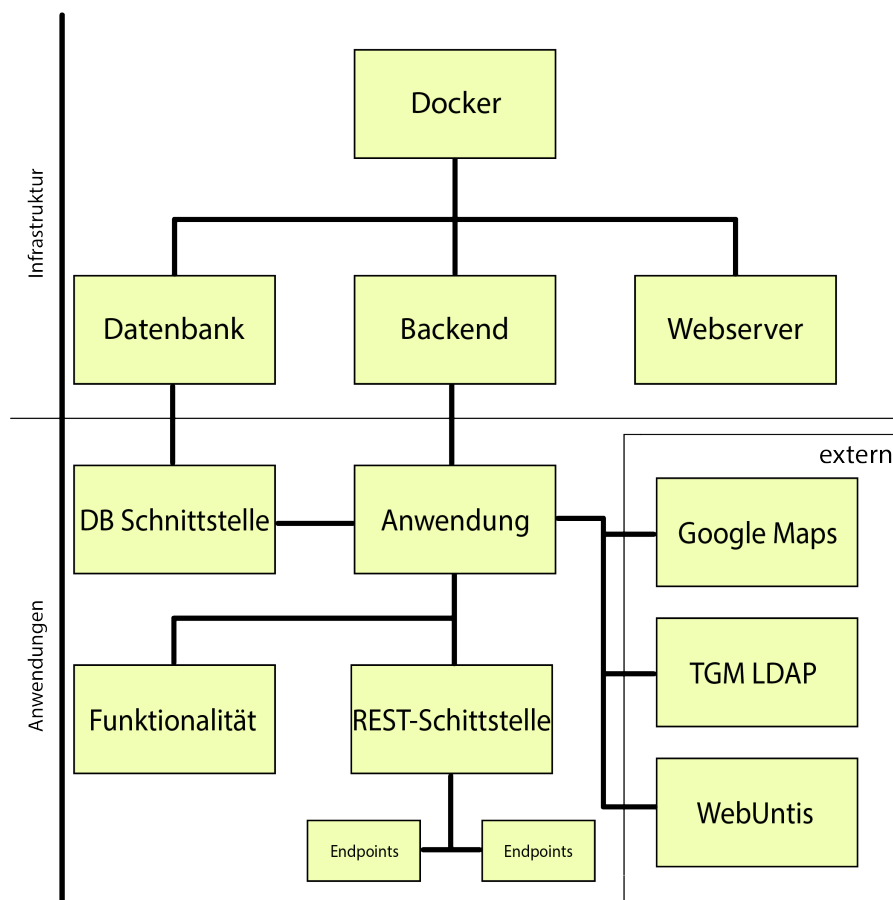


Abbildung 2: Übersicht über die verschiedenen Komponenten der Infrastruktur und der Anwendung

3.2 Docker

Um die Infrastruktur des Projektes einfach aufbauen zu können, wird [Docker](#) genutzt. Da es sich hier um eine komplex strukturierte Infrastruktur handelt wird zusätzlich das Werkzeug [Docker Compose](#) genutzt. Mit Docker Compose kann eine Infrastruktur aufgebaut die der [1. Abbildung \(Übersicht\)](#) entspricht. Für diese sind folgende Container vorgesehen, die in den nächsten Kapiteln noch ins Detail beschrieben werden.

3.2.1 Datenbank

3.2.2 Backend-Container

3.2.3 Webserver

3.3 Deployment

3.4 REST-Schnittstelle

3.4.1 Framework

3.4.2 Endpoints

3.5 Funktionalität

3.5.1 TGM-LDAP Schnittstelle

3.5.2 Datenbank Schnittstelle

3.5.3 Google Maps

3.5.4 WebUntis

3.5.5 E-Mails

3.5.6 PDF-Dateien

3.6 Kommunikation und Datenformate

4 Fazit

Abbildungsverzeichnis

1	HTML Beispielseite	5
2	Übersicht über die Komponenten	8