



## IT-HTL YBBS AN DER DONAU

HÖHERE TECHNISCHE LEHRANSTALT  
FÜR INFORMATIONSTECHNOLOGIE  
AUSBILDUNGSSCHWERPUNKT NETZWERKTECHNIK



# DIPLOMARBEIT

## Echtzeit Visualisierung von Energiesystemen

Ausgeführt im Schuljahr 2021/22 von:

Marcel Entner 5AHITN  
Tobias Kronsteiner 5AHITN  
David Pöchacker 5AHITN

Ybbs an der Donau, am 04.04.2022

Betreuer/Betreuerin:

Dipl.-Ing. Johann Burgstaller

Projektpartner:

Bioenergy and Sustainable Technologies  
GmbH

---

Abgabevermerk:

Datum:

Betreuer:

# Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

## Unterschriften der Projektmitglieder

Ybbs an der Donau, am 04.04.2022

---

Marcel Entner

---

Tobias Kronsteiner

---

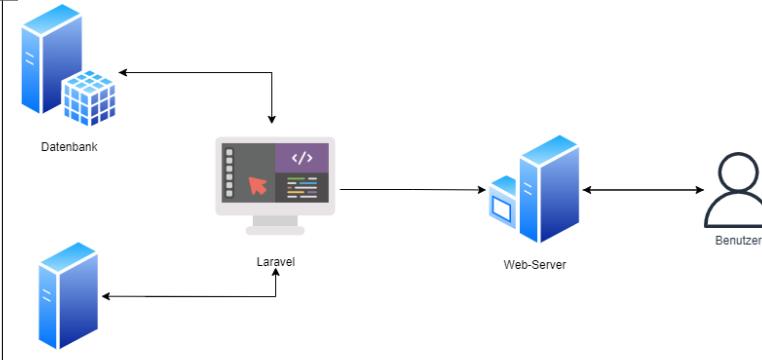
David Pöchacker

# Kurzfassung der Diplomarbeit/Abstract

<b>htl</b> <small>bildung mit zukunft</small>	<b>HÖHERE TECHNISCHE LEHRANSTALT YBBS AN DER DONAU</b>
Fachrichtung:	<b>Informationstechnologie</b>
Ausbildungsschwerpunkte:	<b>Netzwerktechnik</b>

## DIPLOMARBEIT DOKUMENTATION

Namen der Verfasser/innen	David Pöchacker, Marcel Entner, Tobias Kronsteiner
Jahrgang Schuljahr	5AHITN 2021/22
Thema der Diplomarbeit	Echtzeit Visualisierung von Energiesystemen
Kooperationspartner	Bioenergy and Sustainable Technologies GmbH
Aufgabenstellung	Das Ziel der Diplomarbeit "Echtzeit Visualisierung von Energiesystemen," ist, dem Unternehmen Best GmbH eine zentrale Verwaltung von Energiesystemen bereitzustellen. Zusätzlich zur Verwaltung soll es möglich sein, Echtzeitdaten von einer ausgewählten Energietechnologie in Form von Statistiken zu visualisieren.
Realisierung	Die Weboberfläche wurde mit Laravel umgesetzt. Eingegebene Daten werden in einer Datenbank erfasst und mittels Grafana auf der Weboberfläche visualisiert. Abrufbar ist das Produkt über eine vom Auftraggeber bereitgestellte Domain mit dazugehörigem Webserver.
Ergebnisse	Mithilfe des Produkts können Energiesysteme sowie Energietechnologien erstellt, bearbeitet und gelöscht werden. Ein rollenbasiertes Benutzersystem regelt den Zugriff auf die Verwaltung der einzelnen Energiesysteme. Der Administrator hat als Einziger die Möglichkeit neue Benutzer hinzuzufügen oder bestehende zu löschen. Jedem Benutzer ist es möglich, die Grafana Statistiken seiner selbst erstellen Energietechnologien anzeigen zu lassen.

		
Architektur		
Teilnahme an Wettbewerben, Auszeichnungen	Mostviertler Schulinovationspreis noch keine	
Möglichkeiten der Einsichtnahme in die Arbeit	Bibliothek des SZ-Ybbs	
Approbation Prüfer (Datum / Unterschrift)	Prüfer/Prüferin	Direktor bzw. Abteilungsvorstand

<b>htl</b> bildung mit zukunft	<b>HÖHERE TECHNISCHE LEHRANSTALT YBBS AN DER DONAU COLLEGE of ENGINEERING</b>
Department:	Information Technology
Educational focus:	Network and Media Technology

## DIPLOMA THESIS

### Documentation

Author(s)	David Pöchacker, Marcel Entner, Tobias Kronsteiner	
Form		
Academic year	5AHITN 2021/22	
Topic	Echtzeit Visualisierung von Energiesystemen	
Co-operation Partners	Bioenergy and Sustainable Technologies GmbH	
Assignment of Tasks	The goal of the diploma thesis "Echtzeit Visualisierung von Energiesystemen," is to provide the company Best GmbH with a central management of energy systems. In addition to the administration it should be possible to visualize real-time data from a selected energy technology in the form of statistics.	
Realisation	The web interface was implemented with Laravel. Data is entered into a database and visualized on the web interface using Grafana. The product can be accessed via a domain provided by the Best GmbH with an associated web server.	
Results	With the product, energy systems as well as energy technologies can be created, edited and deleted. A role-based user system regulates access to the administration of the individual energy systems. The administrator is the only one who has the possibility to add new users or delete existing ones. Each user is able to view the Grafana statistics of his self-created energy technologies.	
Architecture	<pre> graph LR     DB[Datensatz] --&gt; Laravel[Laravel]     Grafana[Grafana-Server] --&gt; Laravel     Laravel --&gt; WebServer[Web-Server]     WebServer &lt;--&gt; Benutzer[Benutzer]   </pre> <p>The diagram illustrates the system architecture. It consists of four main components: a Database (represented by a blue cube icon), a Grafana-Server (represented by a blue cube icon), a Laravel application (represented by a monitor icon with code symbols), and a Web-Server (represented by a blue cube icon). The Database and Grafana-Server are connected to the Laravel application. The Laravel application is connected to the Web-Server. Finally, the Web-Server is connected to the User (represented by a person icon).</p>	
Participation in Competitions Awards	Mostviertler Schulinovationspreis not yet awarded	
Accessibility of Diploma Thesis	Library of the SZ-Ybbs	
Approval (Date / Sign)	Examiner	Head of College / Department

# Danksagung

Wir möchten uns bei allen bedanken, die uns bei der Erstellung dieser Arbeit geholfen haben.

Insbesondere möchten wir uns bei unserem Diplomarbeitsbetreuer Herrn Dipl.-Ing. Johann Burgstaller bedanken. Er unterstützte uns vor allem bei fachlichen und inhaltlichen Fragen jeglicher Art und stand uns bei jeder Besprechung mit dem Auftraggeber bei. Zusätzlich zu den Besprechungen war er auch in der Freizeit jederzeit für uns zur Verfügung, um diverse Fragen zu beantworten. Bei unserem Kooperationspartner Best GmbH möchten wir uns ebenso für die Zusammenarbeit bedanken. Mit Herrn Stefan Aigenbauer, Armin Cosic, Michael Zellinger und Jürgen Mitterlehner hatten wir während der gesamten Projektphase ständigen Kontakt, um Änderungen sowie Vorschläge des Projektes gemeinsam abzustimmen. Außerdem stellten sie uns diverse Hardware zur Verfügung, wodurch die Umsetzung der Diplomarbeit drastisch erleichtert wurde.

Abschließend möchten wir uns ebenso bei unseren Familien, Freunden sowie Klassenkollegen bedanken, die uns während dieser Phase immer zur Seite standen und uns immer wieder aufs Neue motivierten.

Vielen Dank!

# Inhaltsverzeichnis

<b>Eidesstattliche Erklärung</b>	<b>A</b>
<b>Kurzfassung der Diplomarbeit/Abstract</b>	<b>B</b>
<b>Danksagung</b>	<b>E</b>
<b>Inhaltsverzeichnis</b>	<b>i</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Zielsetzung und Aufgabenstellung . . . . .	1
1.1.1 Beschreibung der Diplomarbeit . . . . .	1
1.1.2 Motivation zur Diplomarbeit . . . . .	2
1.1.3 Ziel der Arbeit . . . . .	2
1.1.4 Ergebnis . . . . .	2
1.2 Rollen und individuelle Zielsetzung der Teammitglieder . . . . .	3
1.2.1 David Pöchacker . . . . .	3
1.2.2 Marcel Entner . . . . .	3
1.2.3 Tobias Kronsteiner . . . . .	4
<b>2 Grundlagen und Methoden</b>	<b>5</b>
2.1 Analyse des vorhandenen Systems . . . . .	5
2.1.1 Begriffe . . . . .	6
2.2 Anforderungen an das Produkt . . . . .	8
2.2.1 Schutz von vertraulichen Informationen . . . . .	8
2.2.2 Statistische Auswertung . . . . .	8
2.3 Architektur des Zielsystems . . . . .	9
2.3.1 Endgeräte . . . . .	9

2.3.2	Serverseitig . . . . .	10
2.3.3	Clientseitige Interaktion des Benutzers . . . . .	10
2.3.4	Framework . . . . .	11
2.3.5	Front-End Templates . . . . .	15
2.3.6	Verbindung der Datenbank mit Laravel . . . . .	18
2.4	Visuelle Darstellung der Energiesysteme und Energietechnologien . . . . .	20
2.4.1	Kartendienste . . . . .	20
2.4.2	Geoinformationssystem . . . . .	22
2.4.3	CSS-System . . . . .	22
2.4.4	Auswahl des Anbieters . . . . .	23
2.5	Berechtigungssystem Benutzer . . . . .	24
2.5.1	Benutzerrollen . . . . .	24
2.5.2	Berechtigungen in Laravel . . . . .	25
2.6	Ui/Ux Design . . . . .	25
2.6.1	Wireframe . . . . .	26
2.6.2	Persona . . . . .	26
2.7	Template Layout . . . . .	26
2.7.1	Platzhalter Yield . . . . .	26
2.7.2	Sections . . . . .	27
2.7.3	Einbindung der definierten Sections . . . . .	27
2.8	Laravel Befehle . . . . .	28
2.8.1	Migration Befehle . . . . .	28
2.8.2	Seeder und Factory Befehle . . . . .	28
2.8.3	Model und Controller Befehle . . . . .	29
2.8.4	Starten des Laravel Entwicklungs Servers . . . . .	30
2.8.5	Befehle nach dem Git Pull . . . . .	30
2.9	Routen in Laravel . . . . .	31
2.9.1	Ressource Routen . . . . .	31
2.9.2	GET Routen . . . . .	31
2.9.3	Auth Routen . . . . .	32
2.10	MVC . . . . .	32
2.10.1	Model . . . . .	33
2.10.2	View . . . . .	33
2.10.3	Controller . . . . .	33

<b>3 Ergebnisdokumentation</b>	<b>34</b>
3.1 Laravel . . . . .	34
3.1.1 Installation . . . . .	34
3.1.2 Bootstrap Einbindung . . . . .	34
3.1.3 Grafana Einbindung . . . . .	35
3.2 Datenbankanbindung in Laravel . . . . .	36
3.2.1 Datenbank Anmeldeinformationen . . . . .	36
3.2.2 Mail Server Konfigurationen . . . . .	37
3.2.3 Migrations . . . . .	38
3.3 Datenbankdesign . . . . .	38
3.3.1 Erstellen eines neuen Schemas . . . . .	38
3.4 Corporate Design . . . . .	40
3.4.1 Vorschläge . . . . .	40
3.4.2 Änderungsvorschläge . . . . .	42
3.4.3 Finales Design . . . . .	42
3.4.4 Definierte Farben . . . . .	43
3.4.5 Überschriften . . . . .	44
3.4.6 Interaktionsfarben . . . . .	46
3.4.7 Schriftarten . . . . .	47
3.4.8 Schriftgrade . . . . .	48
3.4.9 Logo . . . . .	48
3.4.10 Verwendete Icons und deren Bedeutungen . . . . .	49
3.4.11 Map Icons . . . . .	49
3.4.12 Icons in Formularen . . . . .	50
3.4.13 Icons im DataTable . . . . .	51
3.4.14 Buttons . . . . .	52
3.4.15 Tabelle mit generellen Informationen über einzelne HTML Elemente . . . . .	53
3.4.16 Datenformate . . . . .	53
3.5 Webanwendung . . . . .	54
3.5.1 Back-End . . . . .	54
3.5.2 Front-End . . . . .	64
3.5.3 Login . . . . .	68
3.5.4 Registrierung . . . . .	69

3.5.5	Kartendienst Funktionalitäten . . . . .	69
3.5.6	Anzeige von Energiesystemen und Energietechnologien auf der Karte . . . . .	71
3.5.7	Layoutvorlage der Website . . . . .	72
3.6	DataTable . . . . .	72
3.6.1	Individueller DataTable . . . . .	73
3.6.2	Sortierfunktion . . . . .	73
3.6.3	Suchfunktion . . . . .	74
3.6.4	Seitenanzahl . . . . .	74
3.6.5	Icons . . . . .	74
3.6.6	MoveToMarker . . . . .	74
3.7	Galerie Funktionen . . . . .	75
3.7.1	Auswahl eines Energiesystems . . . . .	75
3.7.2	Energietechnologien des Energiesystems anzeigen . . . . .	76
3.8	Grafana . . . . .	77
3.8.1	Automatisches Erstellen der Dashboards . . . . .	77
3.8.2	Automatisches Erstellen der Panels . . . . .	77
3.8.3	Energietechnologien Statistiken anzeigen . . . . .	78
3.9	Einbindung von Google Maps . . . . .	79
3.9.1	Google Cloud . . . . .	79
3.9.2	Google Cloud Plattform Account erstellen . . . . .	79
3.9.3	API's aktivieren und einbinden . . . . .	79
3.9.4	Individuelle Map erstellen und einbinden . . . . .	80
<b>4</b>	<b>Resümee und Ausblick</b>	<b>81</b>
<b>5</b>	<b>Quellen und Literatur</b>	<b>82</b>
<b>6</b>	<b>Verzeichnisse</b>	<b>83</b>
	<b>Abbildungsverzeichnis</b>	<b>84</b>
	<b>Tabellenverzeichnis</b>	<b>87</b>
<b>7</b>	<b>Begleitprotokoll gem. § 9 Abs. 2 PrO-BHS</b>	<b>89</b>
7.1	Begleitprotokoll David Pöchacker . . . . .	90
7.2	Begleitprotokoll Tobias Kronsteiner . . . . .	93
7.3	Begleitprotokoll Marcel Entner . . . . .	95

<b>8 Anhang</b>	<b>98</b>
8.1 Verfasser der Kapitel . . . . .	98
8.1.1 David Pöchacker . . . . .	98
8.1.2 Marcel Entner . . . . .	99
8.1.3 Tobias Kronsteiner . . . . .	99
8.2 Verwendete Software . . . . .	100
8.2.1 Visual Studio Code . . . . .	100
8.2.2 Apache WebServer . . . . .	100
8.2.3 Composer . . . . .	100
8.2.4 Windows Eingabeaufforderung (CMD) . . . . .	100
8.2.5 Github VCS und Github Desktop GUI . . . . .	100
8.2.6 phpMyAdmin . . . . .	101
8.2.7 Adobe XD . . . . .	101
8.2.8 Adobe Photoshop . . . . .	101
8.2.9 LaTex . . . . .	101
8.3 Projektplanung . . . . .	102
8.3.1 Projektkommunikation . . . . .	102
8.3.2 Projektstrukturplan . . . . .	102
8.3.3 Verantwortungsmatrix . . . . .	103
8.3.4 Meilensteinplan . . . . .	104
8.3.5 Terminplan . . . . .	105
8.4 Inhalt von GitHub . . . . .	105

# Kapitel 1

## Einleitung

In diesem ersten Kapitel wird auf die Aufgabenstellung und die Ziele dieser Diplomarbeit eingegangen.

### 1.1 Zielsetzung und Aufgabenstellung

Eine allgemeine Beschreibung, die Veranlassung, das Ziel und das Ergebnis dieser Diplomarbeit ist der Inhalt der folgenden Abschnitten.

#### 1.1.1 Beschreibung der Diplomarbeit

Im Rahmen der Diplomarbeit „Echtzeit-Visualisierung von Energiesystemen“ soll ein Produkt zur Verwaltung von Energiesystemen erstellt werden. Dabei soll auf die bereits bestehenden Komponenten wie Webserver, Datenbank sowie Grafana Server aufgebaut werden. Der Webserver präsentiert die Verwaltungsanwendung, welche öffentlich im Internet für jeden zugänglich ist. Auf dieser Anwendung werden die vorhandenen Energiesysteme sowie Energietechnologien dargestellt. Durch den Kooperationspartner „Best GmbH“ ist es möglich, gemeinsam wichtige Entscheidungen wie die Auswahl der anzuzeigenden Daten zu treffen, da gewisse Informationen über ein Energiesystem<sup>1</sup> sowie eine Energietechnologie<sup>2</sup> aus Datenschutzgründen nicht angezeigt werden sollen. Der Anwender soll die Möglichkeit haben, sein eigenes Energiesystem mit den dazugehörigen Energietechnologien zu erstellen und anschließend deren Echtzeitdaten<sup>3</sup> zu veranschaulichen.

---

<sup>1</sup>Eine genaue Begriffserklärung befindet sich in 2.1.1.3

<sup>2</sup>Eine genaue Begriffserklärung befindet sich in 2.1.1.2

<sup>3</sup>Eine genaue Begriffserklärung befindet sich in 2.1.1.1

### **1.1.2 Motivation zur Diplomarbeit**

Die Diplomarbeit ist dadurch entstanden, dass Mitarbeiter der Best GmbH angefragt haben, ob das SZ-Ybbs mit ihnen eine Diplomarbeit durchführen möchte. Das Ziel des Projektes ist es, eine zentrale Verwaltung von Energiesystemen zu ermöglichen, da eine solche zentrale Verwaltung bei dem Auftraggeber noch nicht vorhanden ist. Da es sich dabei um ein Thema handelt, das im Lehrplan der IT-HTL Ybbs/Donau breiten Raum einnimmt, war die Kooperation mit der Schule für dieses Projekt eine gute Wahl. Daraufhin wurde das Projektteam auf diesen Vorschlag für eine Diplomarbeit aufmerksam, und entschloss sich kurze Zeit danach, dieses Produkt umzusetzen. Es sind zwar bereits eine Vielzahl solcher Produkte vorhanden, jedoch noch keine ideale Lösung, die den Anforderungen des Auftraggebers entspricht. Für das Projektteam war das ein Ansporn, genau dieses Produkt zu entwickeln und es soll mit wenig Aufwand und Vorwissen für einen Benutzer möglich sein, das Produkt zu bedienen.

### **1.1.3 Ziel der Arbeit**

Das Ziel dieses Projektes ist eine Weboberfläche, welche es ermöglicht, Energiesysteme einfach und intuitiv zentral zu verwalten. Die Energiesysteme sollen auf einer Karte angezeigt werden, um einen Überblick aller vorhandenen Energiesysteme zu ermöglichen. Man soll Energiesysteme erstellen können, und es soll möglich sein, einem Energiesystem mehrere Energietechnologien hinzuzufügen. Bei den Energietechnologien soll es möglich sein, deren Echtzeitdaten in Form von Statistiken anzeigen zu lassen. Ebenso soll zu einer Energietechnologie ein Foto hinzugefügt werden können, welches in einer Bildergalerie dargestellt werden soll. Über die gesamte Anwendung ist zudem ein Design Handbuch sowie ein Benutzerhandbuch zu verfassen. Zusätzlich ist ein neues Datenbankschema zu entwickeln, da das bereits vorhandene Datenbankschema nicht dem heutigen Stand der Technik entspricht. Eine rollenbasierte Benutzerverwaltung soll zur Steuerung des Zugriffs auf die Website implementiert werden. Somit ist jeder Benutzer auf der Website nur dazu berechtigt, seine eigenen Energiesysteme zu verwalten, mit Ausnahme des Administrator-Benutzers.

### **1.1.4 Ergebnis**

Das Ergebnis der Diplomarbeit ist eine lauffähige Weboberfläche, die das Verwalten von Energiesystemen mit dazugehörigen Energietechnologien ermöglicht. Die Verwaltung ist mit einem rollenbasierten Benutzerauthentifizierungs-System begrenzt, um unbefugten Besuchern die Verwaltung zu verwehren. Dafür hat der Administrator eine eigene Registrierungsseite, um neue Benutzer anzulegen und bestehende zu löschen. Die Funktionen „Energiesystem“ sowie „Energietechnologie Erstellen, Bearbeiten, Löschung“ und die dazugehörigen Statistiken anzeigen zu lassen erfüllen alle ihre Funktionalitäten. Die vorgesehene Bildergalerie wurde erfolgreich umgesetzt, und zeigt zu jeder Energietechnologie das dazugehörige Bild an. Unter folgendem Link ist das Ergebnis der Diplomarbeit ersichtlich: <https://visu.microgrid-lab.eu/>

## 1.2 Rollen und individuelle Zielsetzung der Teammitglieder

In Abbildung 1.1 sind alle Mitglieder, die bei dieser Diplomarbeit eine wichtige Rolle spielen, ersichtlich.

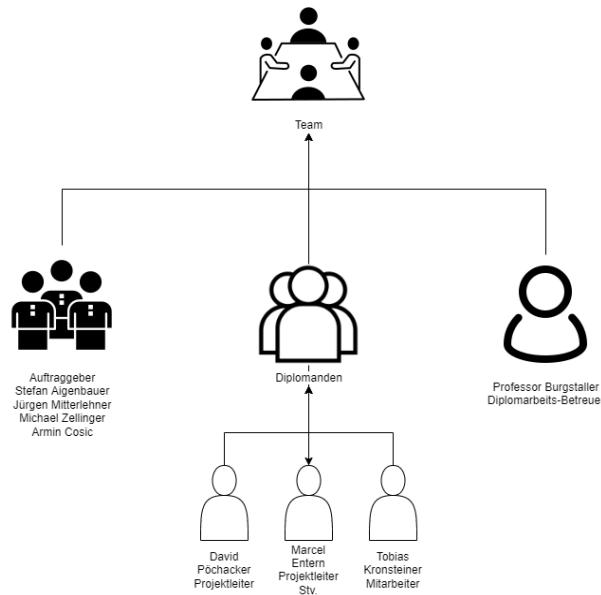


Abbildung 1.1: Team

### 1.2.1 David Pöchacker

Die implementierte Karte muss so konfiguriert werden, dass das Hinzufügen von Energiesystemen sowie Energietechnologien für den Benutzer möglich ist. Zusätzlich zum Erstellen eines Energiesystems oder einer Energietechnologie sollen die Funktionen, diese zu bearbeiten und zu löschen, ebenfalls gegeben sein. Um die Verwaltung der Energiesysteme unter Kontrolle zu haben und gegen unerwünschte Zugriffe zu schützen, ist ein rollenbasiertes Benutzersystem notwendig. Um diese Anforderungen zu erfüllen, müssen folgende Aufgaben umgesetzt werden:

- Kartendienst Funktionen implementieren
- Verwaltungsfunktionen der Energiesysteme sowie Energietechnologien erstellen
- Benutzerverwaltungssystem einrichten

### 1.2.2 Marcel Entner

Hauptaufgabe ist die generelle Konzeption und Umsetzung des Front-End Designs. Er ist dafür verantwortlich, Front-End Vorlagen zu entwerfen und diese mit dem Auftraggeber abzustimmen. Das Teammitglied ist auch für die Umsetzung des Vorschlages auf der Website verantwortlich und verfasst dazu ein Designhandbuch. Eine weitere Aufgabe ist es, eine Bildergalerie, die zu jeder Energietechnologie ein dazugehöriges Bild anzeigt, zu entwerfen. Das Teammitglied war auch dafür verantwortlich, dass eine Impressums- und eine Datenschutzseite vorhanden ist.

### 1.2.3 Tobias Kronsteiner

Das Teammitglied setzt sich mit der Analyse des vorhandenen Systems sowie der Datenbank auseinander. Weiters befasst es sich mit der Analyse, Konzeption und Implementierung von Visualisierungsmöglichkeiten der anwendungsspezifischen Statistiken. Der Zugriff sowie der Datenverkehr zwischen dem Produkt und der neu konzipierten Datenbank gehört ebenfalls zum Aufgabenbereich. Zum größten Teil befasst sich das Teammitglied mit dem Visualisierungs Tool Grafana und dessen HTTP API<sup>4</sup>. Eine besondere Herausforderung stellt hier die Implementierung eines Algorithmus zum erweitern des Datenmodells eines Dashboards<sup>5</sup>. sowie das dynamische Teilen von Panels<sup>6</sup>. auf der Benutzeroberfläche des Produktes dar.

---

<sup>4</sup>Anwendungsspezifische Programmierbar Schnittstelle eines Programmes

<sup>5</sup>Genauere Erklärung erfolgt im Unterabschnitt 3.8.1

<sup>6</sup>Genauere Erklärung erfolgt im Unterabschnitt 3.8.2

# Kapitel 2

## Grundlagen und Methoden

In diesem Abschnitt werden die verwendeten Grundlagen und Methoden, die für die Umsetzung dieses Diplomarbeitsprojekts notwendig sind, dargestellt. Falls bei der Umsetzung mehrere Möglichkeiten zur Wahl standen, werden die einzelnen Möglichkeiten miteinander verglichen und nach einem Vergleich wird die besser geeignete Variante ausgewählt.

### 2.1 Analyse des vorhandenen Systems

Das bestehende System des Auftraggebers umfasst die Komponenten Grafana Server, Webserver und eine Datenbank mitsamt den notwendigen Algorithmen, um diese mit Echtzeitdaten zu befüllen. Bei der vorgegebenen Datenbank handelt es sich um einen MariaDB SQL-Server in der Version 10.1.48. Bei MySQL handelt es sich um eine quelloffene Implementierung des SQL Standards in der Version 5.0.12. In der nachfolgenden Abbildung 2.1 ist das vorhandene System des Auftraggebers ersichtlich.



Abbildung 2.1: Vorhandene System des Auftraggebers

Der Grafana Server wird übernommen, um Dashboards für die Energiesysteme und Panels für die Energietechnologien zu erstellen. Dabei können mehrere Statistiken, auch Panels genannt, zu einem Dashboard zusammengefasst werden, um somit eine Ordnerstruktur am Grafana Server zu erlangen. Der Webserver wird als Produktivserver benutzt, um die fertige Website zu präsentieren. Die vorhandene Datenbank wird aufgrund deren Schemas nicht übernommen, stattdessen wird ein neues besseres Datenbankschema entwickelt. Die dafür notwendigen Algorithmen, um die neue Datenbank mit Echtzeitdaten zu befüllen, werden vom Auftraggeber entwickelt.

## 2.1.1 Begriffe

In diesem Abschnitt werden notwendige Begriffe, die in dieser Diplomarbeit eine wichtige Rolle spielen und zu Missverständnissen führen könnten, erklärt.

### 2.1.1.1 Echtzeitdaten

Der Begriff Echtzeitdaten ist definiert durch die regelmäßige und in gleichen Zeitabständen erfolgende Erfassung sowie Verarbeitung von Daten einer Energietechnologie. Der Zeitabstand der Datenerfassung bei den Energietechnologien beträgt 30 Sekunden. Aufgrund des vorgegebenen Zeitabstandes werden die Echtzeitdaten als weiche Echtzeitanforderung definiert, da das Produkt alle einkommenden Daten schnellstmöglich mit einem konstanten Zeitabstand von 30 Sekunden bearbeitet. Ein Überschreiten dieser Zeitgrenze wird nicht als Versagen definiert, solange sich die Zeit noch in einem akzeptablen Toleranzbereich von wenigen Sekunden befindet. Eine Unterschreitung der Zeitangabe ist sehr selten möglich.

### 2.1.1.2 Energietechnologie

Als Energietechnologie wird ein Stromerzeuger, Stromverbraucher oder ein Energiespeicher bezeichnet. Stromerzeuger sind PV-Anlagen oder Windkraftanlagen. Stromverbraucher sind E-Ladestationen oder Hausanschlusszähler. Batteriespeicher oder Wärmespeicher sind Speicher-Energietechnologien. Die von diesen Technologien erfassten Echtzeitdaten werden in einer Datenbank gespeichert und anschließend grafisch in Form von Statistiken visualisiert.

Dabei werden folgende Echtzeitdaten für die Statistiken verwendet:

- Erzeuger/Verbraucher - Leistung [kW]
- Erzeuger/Verbraucher - Energie [kW/h]
- Speicher – Kapazität/Temperatur [kW/h]/[°]

Die folgende Tabelle bietet eine Übersicht aller vorhandenen Energietechnologien. In der ersten Spalte der Tabelle befinden sich reine Erzeuger-Energietechnologien[PM1]. Verbraucher-Energietechnologien sind in der zweiten Spalte ersichtlich. Speicher-Energietechnologien in der dritten Spalte und in der vierten und zugleich auch letzten Spalte befinden sich Energietechnologien, die sowohl als Verbraucher-Energietechnologie und als Erzeuger-Energietechnologie definiert sind.

<b>Erzeuger</b>	<b>Verbraucher</b>	<b>Speicher</b>	<b>Verbraucher &amp; Erzeuger</b>
PV-Anlage	Wasserstoff- Elektrolyse	Batteriespeicher	Biomasseheizkraftwerk
Stromnetzbezug	E-Ladestation	Wasserstoff-speicher	Biomasseheizwerk
Wasserstoff Brennstoffzelle	Hausanschlusszähler	Wärmespeicher	Biomassekessel
Windkraftanlage	Gebäude- Wärmebedarfsszähler	Kältespeicher	Kompressionskältemaschine
Wärmenetzbezug	Gebäude- Kältebedarfsszähler		Ab- oder Adsorptionskältemaschine
Solarthermieanlage			
Wärmepumpe			

Tabelle 2.1: Energietechnologien

### 2.1.1.3 Energiesystem

Der Begriff Energiesystem fasst mehrere Energietechnologien in einem bestimmten Gebiet, die genau einem Energiesystem zugeordnet sind, logisch zusammen. Ein Energiesystem kann somit eine Gemeinde, ein Gebäude oder ein einziger Haushalt mit mehreren Energietechnologien sein.

### 2.1.1.4 Front-End

Der Begriff Front-End beschreibt die Weboberfläche auf welcher der Benutzer verschiedene Interaktionen durchführen kann. Dabei umfasst dieser Begriff alle Unterseiten der Weboberfläche welche von einem Benutzer verwendet werden können.

### 2.1.1.5 Back-End

Mit dem Begriff Back-End ist das Laravel-Projekt, welches für die Ereignisse der Benutzer-Interaktionen zuständig ist, gemeint. Zusätzlich sind in diesem Begriff die Zugriffe auf die Datenbank sowie auf den Grafana Server mit eingebunden, welche durch das Laravel-Projekt durchgeführt werden.

## 2.2 Anforderungen an das Produkt

Die Datenbank des Auftraggebers ist die größte Schwachstelle des vorhanden Systems. Aufgrund dessen wird ein neues Datenbankschema entwickelt. Neben der Entwicklung eines neuen Datenbankschemas sind folgende Anforderungen an das Produkt gegeben:

- Übersichtliche Darstellung von Energiesystemen und Energietechnologien auf einem Kartendienst
- Grafana-Statistiken mit Echtzeitdaten der Energietechnologien anzeigen
- Bildergalerie um Energietechnologien eines ausgewählten Energiesystems zu präsentieren
- Managementfunktion mit Hilfe von verschiedenen Benutzern für unterschiedliche Berechtigungen

### 2.2.1 Schutz von vertraulichen Informationen

Sämtliche Daten wie Adressen, Standorte (in Form von Koordinaten) oder die Namen der Ersteller von Energiesystemen oder Energietechnologien sind vertrauliche Informationen und sollten somit nicht für jeden einsehbar sein. Darum sind diese Daten nur in der Datenbank gespeichert, welche nur für Benutzer mit entsprechender Berechtigung zugänglich ist. Die Daten, die in den Grafana-Statistiken dargestellt werden, sind ebenso vertrauliche Daten. Die Informationen, die man einer solchen Statistik entnehmen kann, könnten in die falschen Hände gelangen und zu unerwünschten Tätigkeiten führen. Um dieses zu verhindern, sieht jeder angemeldete Benutzer nur seine eigenen Statistiken und keine anderen. Ein nicht angemeldeter Besucher sieht keine Statistiken.

### 2.2.2 Statistische Auswertung

Zur statistischen Auswertung wird das Daten visualisierungs Framework Grafana verwendet. In Grafana werden einzelne Energiesysteme als “Dashboards” abgebildet, in diesen Dashboards werden die Energietechnologien als “Panels” erstellt. In der Konfiguration der Panels werden die SQL Abfragen definiert aus welchen sich die Statistiken zusammensetzen. Die Statistiken bilden sowohl Echtzeit Daten als auch Daten der letzten 24 Stunden ab. Die Statistiken werden hier als Zeitreihen<sup>1</sup>. Diagramme abgebildet.

---

<sup>1</sup>In diesen Diagrammen werden Zahlenwerte in Abhängigkeit von Zeitwerten angezeigt.

## 2.3 Architektur des Zielsystems

Die Abbildung 2.2 repräsentiert die Architektur des Zielsystems. Der Benutzer greift über die Weboberfläche auf den Webserver zu, wo sich das Laravel-Projekt befindet. Auf dieser Website kann der Benutzer verschiedene Interaktionen durchführen. Dabei ist die Weboberfläche ständig in Verbindung mit der Datenbank sowie dem Grafana Server. Sobald der Benutzer auf die Energiesysteme-Seite der Weboberfläche wechselt, wird eine Datenbankabfrage aller vorhandenen Energiesysteme durchgeführt. Zusätzlich dazu wird mittels API-Abfrage auf den Grafana Server zugegriffen, um Statistiken einer Energietechnologie anzuzeigen, falls dies vom Benutzer angefordert wird.

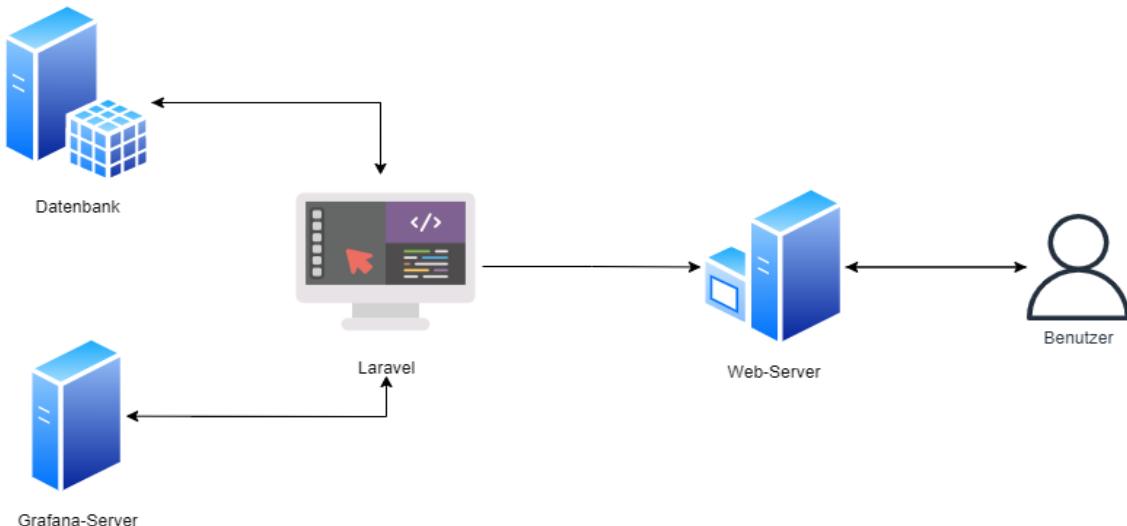


Abbildung 2.2: Architektur des Zielsystems

### 2.3.1 Endgeräte

Das Produkt wurde als Browser Anwendung entwickelt, um überall lauffähig zu sein. Es ist in Chrome, Firefox und Edge Browser getestet worden und ist dort voll funktionsfähig. Die Oberfläche wurde für einen 13 bis 27 Zoll Monitor entwickelt und funktioniert dort in jeder Zoomstufe einwandfrei. Klares Ziel war es, das Produkt nicht für den Mobiltelefongebrauch zu programmieren. Das Team hat sich bewusst gegen eine native Anwendungsentwicklung entschieden, um so die Verwendbarkeit auf allen Plattformen zu garantieren. Darüber hinaus würde eine parallele Anwendungsentwicklung auf den drei gängigen Betriebssystemen (Windows, MacOS, Linux) den zeitlichen Rahmen der Diplomarbeit sprengen. In welchen Browser Versionen das Produkt getestet wurde ist in der Tabelle 2.2 ersichtlich.

Browser	Version
Firefox	99.0beta
Chrome	99.0.4844.51
Edge	99.0

Tabelle 2.2: Browser und Versionen in denen das Produkt getestet wurde

### 2.3.2 Serverseitig

Bei PHP handelt es sich um einen serverseitige zur Laufzeit interpretierte Skriptsprache, mit welcher sich Daten dynamisch auf Website anzeigen lassen. Da es sich bei Laravel um ein PHP Framework handelt, wird der Code am Webserver beim Laden der Website interpretiert. Laravel kommuniziert serverseitig mithilfe eines sogenannten ORM (Object Relational Mapper) mit der Datenbank. Hiermit kann der Programmierer auf vorgefertigte Funktionen des ORM zurückgreifen und muss Datenbankoperationen wie Abfragen oder das Erstellen von Tabelleneinträgen nicht als SQL Abfragen Formatieren.

### 2.3.3 Clientseitige Interaktion des Benutzers

Bei dem Punkt Clientseitig befindet man sich in der Architektur des Zielsystems bei dem Element JavaScript und mit den damit verbundenen Interaktionen des Benutzers. Map-Interaktionen sind Interaktionen mit einer Karte, dazu gehört das Erstellen, Bearbeiten und Löschen von Energiesystemen sowie Energietechnologien. Eine weitere Interaktion wäre die Verwendung des Adresssuchfeldes, welche mit einem Klick auf den Button „Suche“ oder durch Drücken der Enter-Taste durchgeführt wird. Tabellen-Interaktionen sind eine zusätzliche clientseitige Aktion, da die vom DataTable<sup>2</sup> bereitgestellten Funktionen wie die Suchfunktion, Sortierungsfunktion oder Seitennummerierung allesamt clientseitig stattfinden und somit keinen Einfluss auf das Back-End haben. Folgende Abbildung 2.3 zeigt den Aufbau der einzelnen Elemente, die zusammenarbeiten, um die Website zu erstellen.

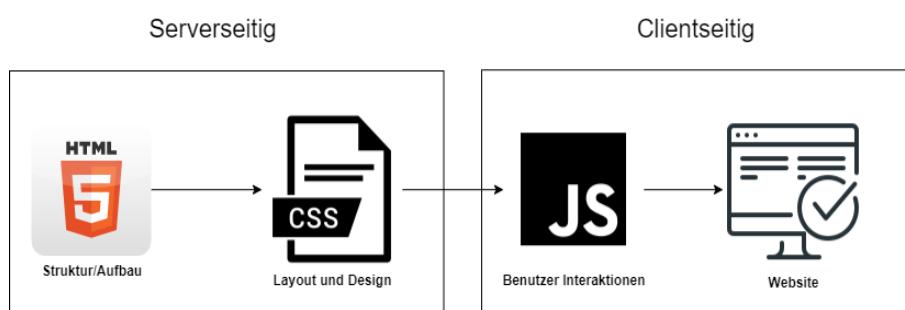


Abbildung 2.3: Clientseitiger Aufbau der einzelnen Elemente

<sup>2</sup>Eine genaue Begriffserklärung befindet sich in 3.6

## 2.3.4 Framework

Die Wahl des Frameworks ist für die Umsetzung des Projekts entscheidend. Unter dem Begriff Framework versteht man eine Art Programmiergerüst, welches es dem Programmierer deutlich erleichtert, ein Produkt zu erstellen. Jedes Framework bietet spezielle Lösungen und Lösungsansätze an und damit hat jedes Einzelne sein eigenes Einsatzgebiet. Zusätzlich wurde noch die Software-Bibliothek React ergänzt, da auch diese eine mögliche Umsatzungsvariante ist. In den nachfolgenden Unterkapiteln werden mögliche Frameworks für die Umsetzung des Projekts kurz erklärt. In 2.3.4.5 wird genauer darauf eingegangen, welches Framework gewählt wurde und aus welchem Grund.

### 2.3.4.1 Laravel

Das PHP-Framework Laravel, welches 2011 entwickelt wurde, basiert auf dem MVC Muster<sup>3</sup> und bietet damit eine sehr gute Strukturierung und Übersicht beim Arbeiten. Die meisten Projekte werden mit Laravel im Back-End in Kombination mit Vue.js im Front-End umgesetzt. Genaue Informationen zu Vue.js sind in 2.3.5.3 nachzulesen. Laravel bietet jedoch die Möglichkeit, im Back- als auch im Front-End verwendet zu werden. Es lässt sich in folgende Einzelteile strukturieren:

- Migrations
- Views
- Controller
- Models
- Routen

Migrations sind die Abbildung der Tabellenstruktur und ermöglichen es, bei richtiger .env Datei Konfiguration, mit artisan<sup>4</sup> Befehlen, ganz einfach eine Datenbank und die dazugehörigen Tabellen zu erstellen und diese auch genauso einfach wieder zu löschen oder zu leeren. Views fungieren als visueller Gliederungspunkt zwischen den Controllern und den Benutzern. In ihnen wird alles, was auf der Oberfläche ersichtlich ist, ausprogrammiert und gestaltet. Controller bilden die Verbindung zwischen den Views und den Models und ermöglichen es dem Benutzer, Daten mithilfe eines Zugriffs auf das Model zu verändern. Models bilden die Datenstruktur ab und ermöglichen den Zugriff auf die Daten und die Änderung dieser. Routen vermitteln die Benutzerabfrage von der View mit dem dazugehörigen Controller und werden in der Datei “web.php” definiert. Der ganze Prozess ist in Abbildung 2.4 visuell dargestellt.

---

<sup>3</sup>wird in Abschnitt 2.10 genauer erläutert

<sup>4</sup>artisan ist die in Laravel enthaltene Befehlszeilschnittstelle

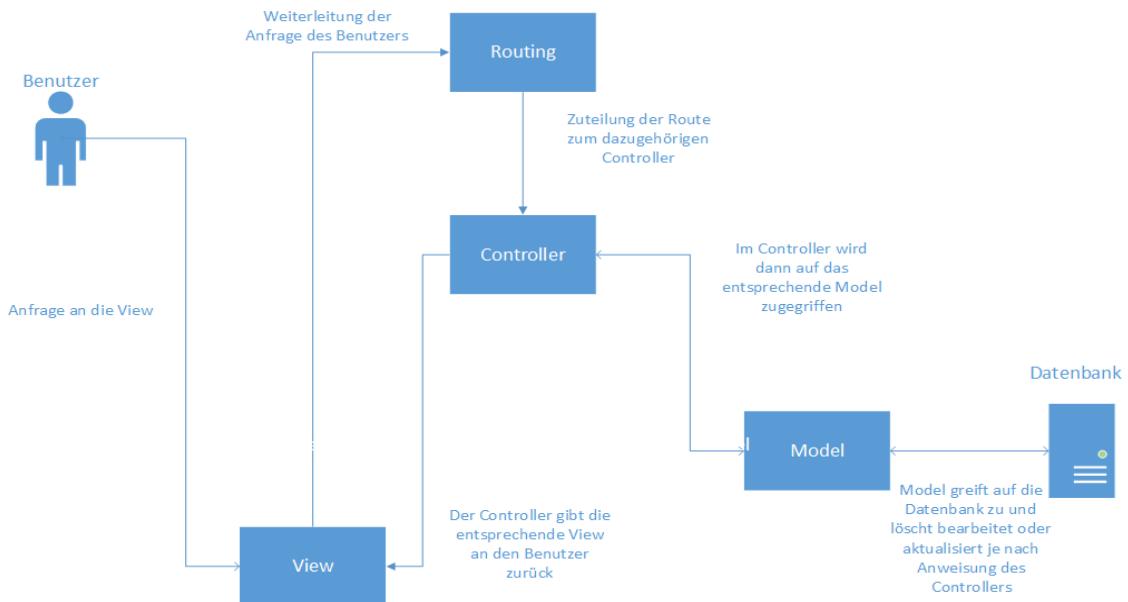


Abbildung 2.4: Laravel MVC

#### 2.3.4.2 Angular

Angular ist ein clientseitiges JavaScript-Web-Framework und wird meistens zum Erstellen von sogenannten Single Page Applications<sup>5</sup>, wie beispielsweise ein einfaches Bedienelement zur Steuerung einer Maschine, verwendet. Angular ist speziell für Web-, Desktop- und Mobile-Anwendungen entworfen. Ein weiterer Aspekt des Frameworks ist es, dass es von Google entwickelt wurde und so ein langfristiger Support gewährleistet ist. Angular macht es mit der Kombination aus HTML und TypeScript möglich, so ziemlich jede mögliche Aufgabenstellung zu bewältigen. Es bietet auch besondere Features wie eine bidirektionale Datenbindung an und ist somit gut für Echtzeitanwendungen geeignet. Alle Daten zu Angular wurden der Quelle [?] entnommen.

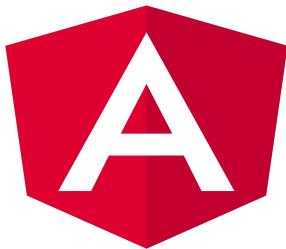


Abbildung 2.5: Angular Logo

<sup>5</sup>Anwendung die nur auf einer Seite zu bedienen ist

### 2.3.4.3 ASP.NET

ASP.NET (Active Server Pages) ist ein Web Application Framework, welches 2002 von der Firma Microsoft veröffentlicht wurde. Es ist der Nachfolger des ASP Frameworks und bietet eine perfekte Grundlage, um dynamische Websites, Webanwendungen und Webservices zu entwickeln. Projekte werden hier in der Regel in der Sprache C# programmiert. Jedoch besteht auch die Möglichkeit, andere Sprachen wie beispielsweise Perl, Python oder Cobol zu verwenden. ASP.NET bietet nicht nur die Möglichkeit, dynamische Websites zu erstellen, sondern auch Desktop Anwendungen. Durch die vordefinierte Datenstruktur, die ASP.NET bereitstellt, hilft es dem Programmierer Programmiersprachen nicht zu vermischen und einen übersichtlichen Programmierstil beizubehalten. Des Weiteren können viele Teile automatisch generiert werden und ersparen dem Programmierer damit einen immensen Aufwand.



Abbildung 2.6: ASP.NET Logo

Alle Informationen und Daten über ASP.NET können unter der Quelle xy (<https://asp.mvc-tutorial.com/de/421/einführung-ist-asp-net-mvc/>) nachgelesen werden.

### 2.3.4.4 React

React ist eine JavaScript-Softwarebibliothek, die es erleichtert, Benutzeroberflächen in Form von Web Applikationen zu erstellen. Es ist komponentenbasiert, was bedeutet, dass jedes Element in Blöcken aufgebaut ist und durch Zusammenfügen dieser einzelnen Codebits kann dann eine sogenannte View erstellt werden. Dies bietet die Möglichkeit, bereits erstellte Codebits auch in anderen Views zu verwenden. React wird oft zusammen mit ASP.NET verwendet.

Genaueres zu React kann unter der Quelle xy (<https://t3n.de/news/react-facebook-623999/>) nachgelesen werden.

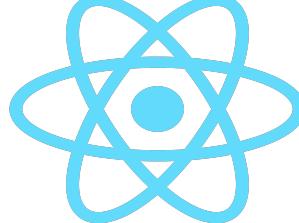


Abbildung 2.7: React Logo

### 2.3.4.5 Entscheidung des Frameworks

Positive und negative Aspekte jedes Frameworks wurden vom Projektteam, mithilfe der Tabelle 2.3 abgewogen. Daten für diese Tabelle wurden aus den Quellen xy entnommen.

Framework oder Bibliothek	Vorteile	Nachteile
React	Software Skalierbarkeit; hohe Flexibilität; Codebits ermöglichen es, die Redundanz zu erhöhen; schnelles Rendering; leichter Einstieg	Es gibt nicht genug Möglichkeiten, eine Webapp zu erstellen und benötigt Zusatzbibliotheken; schweres debuggen; komplexe Benutzeroberfläche
ASP.NET	hohe Flexibilität; MVC Architektur; gute Skalierbarkeit; einfache Benutzung	ressourcenaufwendig; teuer, aufgrund des Lizenskaufes; keine aktuelle Dokumentation; alte Versionen sind nicht mehr kompatibel
Angular	frühzeitige Fehlererkennung; konsistent	relativ starr und unflexibel; schwer zu erlernen;
Laravel	Unit Tests werden angeboten; einfache Erweiterbarkeit; Routing in der App; nutzt die neuesten Funktion von PHP; sehr gute Dokumentation; integrierte Benutzerverwaltung; integrierte Mailverwaltung; performant; einfache Handhabung; Bootstrap Einbindung ist einfach möglich	unterstützt keine Zahlungsfunktion; keine Mobil App Entwicklung; oft Probleme bei Updates

Tabelle 2.3: Entscheidung des Frameworks

Nach Begutachtung der Tabelle 2.3 und aufgrund der überwiegenden positiven Aspekte, entschied sich das Projektteam für das Framework Laravel. Entscheidend war beispielsweise die einfache Handhabung und die Möglichkeit, Bootstrap in Kombination zu verwenden. Da dieses Framework auch im Schulfach Softwareentwicklung genauer bearbeitet wurde, besitzt das Projektteam bereits ein gutes Grundwissen. Es wurde darüber hinaus überlegt, ob auch das Front-End Framework Vue.js zusätzlich verwendet werden soll. Da Vue.js nur zusätzliche Komplexität ohne einen wirklichen Nutzen mit sich bringen würde, wurde diese Option vernachlässigt. Das Framework Laravel bietet alle nötigen Front- und Back-End Möglichkeiten, die es ermöglichen, Laravel im Front wie auch im Back-End zu verwenden. Entscheidender Faktor war auch, dass das Projektteam noch nie mit einem anderen Framework gearbeitet hat, und so eigentlich nur Laravel in Frage gekommen ist.

### 2.3.5 Front-End Templates

Ein Front-End Template dient als Vorlage, um die Web Oberflächengestaltung zu erleichtern. Deswegen ist die Auswahl des Front-End Templates essenziell für die Umsetzung der Weboberfläche des Produkts. Jedes Front-End Template bringt seine Vor- und Nachteile mit sich, wobei das Projektteam die Anforderungen an das Front-End Template mit den Vor- und Nachteilen jeden einzelnen Front-End Templates verglichen hat, um sich schlussendlich für eines zu entscheiden. Dabei sind die Anforderungen des Projektteams an das Front-End Template, dass es unkompliziert in das Projekt einzubinden ist und das Projektteam bereits Erfahrung damit hat, um das Arbeiten damit zu erleichtern. Weitere Anforderungen sind, dass es Responsive-Layouts ermöglicht, sowie Vorlagen von Komponenten, die eingebunden werden können, zur Verfügung stellt. Die drei Front-End Templates, die das Projektteam zur Auswahl stellt, sind Bootstrap, Tailwind und Vue.js. Diese Auswahl ist dadurch entstanden, da das Projektteam bereits erste Erfahrungen damit sammeln konnten.

#### 2.3.5.1 Bootstrap

Bootstrap ist ein Open Source Front-End Template. Es basiert auf den Programmiersprachen HTML, CSS sowie JavaScript und stellt Gestaltungsvorlagen wie Formulare, Tabellen, Buttons und weitere Oberflächengestaltungen bereit. Bootstrap ist weltweit eines der bekanntesten und beliebtesten Front-End Template. Ebenso bietet Bootstrap eine Open Source-SVG-Icon-Bibliothek an, womit die Kompatibilität zwischen Komponenten und Icons bestenfalls gegeben ist. Vorteil dieses Templates ist, dass es leichtgewichtig [PM1] ist, weil nur Front-End-Daten geladen werden müssen und keine Backend-Funktionalitäten. Zusätzlich ist es problemlos in das Projekt einzubinden und stellt mehrere Vorlagen zur Verfügung, die ebenso eingebunden werden können. Bootstrap ist kostenfrei zu verwenden, und das Projektteam bringt bereits Erfahrung mit diesem Template mit. Die Nachteile sind, dass Bootstrap wenige Back-End-Funktionen bereitstellt, und es ist weniger gut geeignet für sehr große Applikationen. In Abbildung 2.8 ist das Bootstrap Logo ersichtlich.



Abbildung 2.8: Bootstrap Logo

### 2.3.5.2 Tailwind.css

Tailwind ist ein Utility-First CSS-Template, welches derzeit weltweit sehr beliebt ist. Diese Art von Template soll mehr Flexibilität bieten als die traditionellen Vorgänger. Tailwind.css stellt Utility-Klassen zur Verfügung, mit denen man selbst Klassen zum Stylen der Komponenten definieren kann. Ein Inlinestyles ist bei diesem Template ebenso möglich, womit externe CSS-Dateien überflüssig sind. Tailwind stellt Hilfsklassen zur Verfügung, welche das Arbeiten um einiges erleichtern. Zusätzlich bietet dieses Template eine hohe Flexibilität sowie mehrere Funktionen, um ein Responsive Design zu erreichen. Jedoch hat dieses Template keine vorgefertigten Komponenten, auf welche zurückgegriffen werden kann, und der Benutzer dieses Templates braucht sehr viel Zeit zum Erlernen der richtigen Verwendung. Die Größe der zu installierenden CSS-Datei ist sehr groß, da dieses Template eine breite Palette von Klassen bereitstellt, wobei die meisten oft in einem Projekt nicht verwendet werden. Das Projektteam hat mit diesem Template wenig Erfahrung, und die Dokumentation im Internet ist sehr mangelhaft. In Abbildung 2.9 ist das Tailwind.css Logo ersichtlich.



Abbildung 2.9: Tailwind Logo

### 2.3.5.3 Vue.js

Das Template Vue.js, welches für die Front-End-Entwicklung eingesetzt werden kann, wird immer populärer. Die Einfachheit und Benutzerfreundlichkeit ist ein Grund für die immer größer werdende Beliebtheit. Vue.js basiert auf dem virtuellen DOM<sup>6</sup>, was bedeutet, dass das Abarbeiten von Befehlen schneller funktioniert als beim herkömmlichen bekannten DOM<sup>7</sup>. Vue.js basiert auf den Programmiersprachen HTML, CSS und JavaScript, womit die wichtigsten verwendeten Sprachen abgedeckt sind. Vue.js stellt Funktionen wie Unitests, End-to-End-Tests sowie Routing-Systeme bereit. Die Lesbarkeit bei diesem Template ist sehr gut, da alle Komponenten sich in einer Datei befinden. Das Projektteam hat nur wenig Erfahrung mit diesem Template und es herrschen generelle Sprachbarrieren, da vieles nur in der Sprache Chinesisch vorhanden ist. In Abbildung 2.10 ist das Vue.js Logo ersichtlich.



Abbildung 2.10: Vue.js Logo

<sup>6</sup>Virtuelles Document Object Model, beim Aktualisieren schneller als das herkömmliche

<sup>7</sup>Document Object Model, stellt XML oder HTML Dokumente als Baumstruktur dar

### 2.3.5.4 Entscheidung des Front-End Templates

Das Projektteam hat sich für das Front-End Template Bootstrap CSS entschieden aufgrund der Vorteile, die es mit sich bringt. Diese Entscheidung wurde unter anderem deswegen so getroffen, da das Projektteam bereits Erfahrung mit diesem Framework sammeln konnte und dieses Framework alle Anforderungen, die an das Front-End Template gestellt wurden, erfüllt. In der nachfolgenden Tabelle ist eine Gegenüberstellung der Vor- und Nachteile der Front-End Templates ersichtlich:

Front-End Template	Vorteile	Nachteile
<b>Bootstrap</b>	basiert auf HTML, CSS und JavaScript stellt Gestaltungsvorlagen bereit weltweit sehr bekannt und beliebt leicht in das Projekt einzubinden Projektteam hat bereits Erfahrungen damit	wenige Back-End Funktionen nicht geeignet für große Applikationen
<b>Tailwind.css</b>	weltweit sehr beliebt bietet mehr Flexibilität stellt Utility-Klassen zur Verfügung	keine vorgefertigten Komponenten Projektteam hat sehr wenig Erfahrung Online-Dokumentation ist mangelhaft erfahrungsloser Benutzer benötigt viel Einarbeitungszeit Größe der CSS-Datei ist enorm
<b>Vue.js</b>	einfach und benutzerfreundlich basiert auf HTML, CSS und JavaScript stellt Funktionen wie UniTests, End-to-End Tests und Routing-Systeme bereit hohe Lesbarkeit des Codes	Projektteam hat wenig Erfahrung Sprachbarrieren, vieles nur auf Chinesisch

Tabelle 2.4: Vor- und Nachteile der Front-End Templates

## 2.3.6 Verbindung der Datenbank mit Laravel

In Laravel ist die Verbindung der Datenbank mithilfe der .env Datei möglich. Tabellen werden in sogenannten Migrations definiert. Auf die einzelnen Komponenten und deren Individualisierung wird in den Kapiteln 2.3.6.1 und 2.3.6.2 noch genauer eingegangen.

### 2.3.6.1 Laravel .env Datei

In Laravel bietet die .env Datei, welche sich direkt im Projektordner befindet, die Möglichkeit, alle nötigen Umgebungsvariablen zu initialisieren. Ein Beispiel für Umgebungsvariablen sind Datenbank Anmeldeinformationen, Cache-Treiber, Mail-Server-Domain oder auch die App URL. Wichtig ist es, dass die .env Datei immer gerätespezifisch konfiguriert wird. Das heißtt, es ist wichtig beim Zusammenarbeiten, speziell mit Git, sehr gut aufzupassen. Das Projekt Team hatte Problem, wie beispielsweise ungleiche PHP Versionen, die zu Fehlern und Nichtausführbarkeit des Produkts führten. Auf dem Produktivsystem gab es dann Probleme mit der Benennung der Ordner. Folge dessen ist es empfehlenswert, die .env Datei mit einem Gitignore<sup>8</sup> zu versehen, um Komplikationen zu vermeiden. Es besteht auch die Möglichkeit, Änderungen in der .env Datei direkt aus dem Programm vorzunehmen.

### 2.3.6.2 Migrations

Migrations bilden in Laravel eine Möglichkeit, die Tabellenstruktur im Programm selbst fest zu legen. Sie sind sozusagen der Blueprint<sup>9</sup>, in welchem die gesamte Tabellenstruktur vordefiniert ist. Für jede Tabelle, die in der Datenbank, welche vorher in der .env Datei definiert wurde, erstellt werden soll, gibt es eine eigene Migration. Migrations sind unterteilt in eine up() und in eine down() Methode<sup>10</sup>. In der up() Methode wird der Name der Tabelle, der Name der einzelnen Tupel und deren Datentyp definiert. Der Datentyp id() ist standardmäßig immer der primary Key. In der down() Methode wird die Tabelle von der Datenbank gelöscht. Die beiden Methoden werden durch artisan Befehle, welche im Kapitel 2.8.1 ersichtlich sind, angesteuert. Erstellt werden Migrations mithilfe eines weiteren Laravel Befehls, welcher im Kapitel 2.8.1 beschrieben wird. Wenn die Migration erfolgreich erstellt wurde, ist diese folgendermaßen benannt: JJJJ.MM.DD\_tempname.php und befindet sich im Ordner Projektname\database \migrations.

Alle Informationen liegen der Quelle (<https://laravel.com/docs/9.x/migrations>) zugrunde.

---

<sup>8</sup>Gitignore markiert Daten, die von Git ignoriert werden sollen

<sup>9</sup>ein Blueprint ist sozusagen der Bauplan oder eine Vorlage wie etwas auszusehen hat

<sup>10</sup>Methoden sind eine Art Container, für den in ihnen auszuführenden Code

### 2.3.6.3 Seeder und Factories

Factories ermöglichen es, Fake Models<sup>11</sup> zu erstellen und mithilfe dieser auf die Datensätze in der Datenbank zuzugreifen und diese zu verändern. Daten werden mithilfe der Open Source-Bibliothek Faker generiert. Die Faker Bibliothek kann folgende Daten generieren:

- E-Mail
- einen Namen
- eine Telefonnummer
- Wörter
- Sätze
- Absätze
- Zufallszahlen

Datenbank-Factories erlauben es, Tabellen programmgesteuert mit Daten anzufüllen.

Sie ermöglichen es, innerhalb von einigen Minuten tausende Daten in die Datenbank zu schreiben.

Die Hauptaufgabe eines Seeders ist es jedoch nur die Factories aufzurufen.

### 2.3.6.4 Datenübergabe in Laravel

Ein Controller hat die Aufgabe mithilfe eines Models die Daten zu verändern und danach eine View zurückzugeben. Möchte man jedoch Daten wie Attribute, welche im Formular mitgeschickt werden, auf der zurückgegebenen View verwenden, müssen diese mitgegeben werden. Es gibt die Möglichkeit, Daten mithilfe von compact in das Return Statement<sup>12</sup> anzuhängen. Dies würde im Code folgendermaßen aussehen:

Quellcode 2.1: Return Statement mit Compact

```
1 return view('Viewname', compact('data'));
```

In der View kann dann mit folgendem Code auf die Daten zugegriffen werden:

Quellcode 2.2: Zugriff mittels Foreach Schleife

```
1 @foreach ($data as $d)
2 $d->Beispiel_Attribut
3 @endforeach
```

Die zweite Variante Daten zu übergeben schaut folgendermaßen aus:

Quellcode 2.3: Attributübermittlung in einem return View Statement

```
1 return view('Viewname', ['AttributName' => 'AttributWert']);
```

<sup>11</sup>Fake Models sind Models, die nur temporär angelegt werden um auf die Daten zugreifen zu können

<sup>12</sup>Rückgabewert einer Funktion

Diese Variante bietet den Vorteil, dass im Code gleich direkt auf das Attribut zugegriffen werden kann und keine Foreach Schleife nötig ist. Ein solcher Zugriff würde folgendermaßen aussehen :

Quellcode 2.4: Zugriff im Programm

```
1 {{ $AttributName }}
```

Alle Informationen und Arten der Datenübergabe wurden aus dem Video aus der Quelle xy () entnommen.

## 2.4 Visuelle Darstellung der Energiesysteme und Energietechnologien

Von Seiten des Auftraggebers wurde von Anfang an der Wunsch geäußert, Energiesysteme und Energietechnologien, mithilfe eines Kartendienstes visuell darzustellen. Um andere Möglichkeiten, wie eine Karte auszuschließen, wurde auch zu Geoinformationssystem und CSS-System recherchiert. Die Ergebnisse dieser Recherchen werden in den nachfolgenden Kapiteln behandelt.

### 2.4.1 Kartendienste

Das Produkt stellt eine zentrale Verwaltungsmöglichkeit in Form einer Karte bereit. Um diese Karte auf dem Produkt zu verwenden, wird ein Kartenanbieter benötigt. Die zwei Optionen Google Maps und OpenStreetMap wurden vom Projektteam analysiert. Auf diese zwei Anbieter wird in den Kapitel 2.4.1.1 und 2.4.1.2 genauer eingegangen. Von Seiten des Auftraggebers wurde, aufgrund einer gewissen Abneigung zu Diensten, die von Google bereitgestellt werden, eine Präferenz zu OpenStreetMap geäußert.

#### 2.4.1.1 Google Maps

Google Maps ist der klare Marktführer in der Sparte der Kartendienst Anbieter. Der Großteil aller Webanwendungen, welche Maps verwenden, werden mit Google Maps umgesetzt. Google Maps überzeugt nicht nur mit einer sehr hohen Individualisierungsmöglichkeit, sondern auch mit einer sehr hohen Ausfallsicherheit. Der größte Vorteil ist jedoch, das große Angebot an vordefinierten API Diensten. Diese können einfach über die Google Cloud Platform überwacht, bedient oder aktiviert werden. Genauere Infos zur Google Maps Plattform stehen im Kapitel 3.9.1. Ein weiterer Vorteil von Google Maps ist es, dass es wirklich weitflächig alles abdeckt, ganz im Gegensatz zu OpenStreetMap. Der einzige Nachteil ist, dass die Verwendung einer Google Maps Karte auf der Website kostenpflichtig ist. Es gibt jedoch auf der Google Cloud Seite eine genaue Kostenaufstellung. Bezahl wird bei Verwendung der API Dienste. Google verrechnet hier je nach API Dienst einen gewissen Betrag pro Verwendung. Diese Kosten waren jedoch bei der Entwicklung kein Problem, da Google Cloud für die ersten 30 Tage ein Startkapital von 300 Dollar anbietet.



Abbildung 2.11: Google Maps Logo

#### 2.4.1.2 OpenStreetMap

OpenStreetMap ist ein 2004 gegründetes Projekt, welches das Ziel hat, eine frei verfügbare Weltkarte zu erstellen. Gegründet wurde dieses Projekt von Steve Coast und zählt mittlerweile um die 7.8 Millionen Nutzer. Das Hauptargument OpenStreetMap eher als Google Maps zu verwenden ist es, dass dieses kostenlos und lizenfrei ist. Es gibt jedoch gravierende Nachteile gegenüber Google Maps. Openstreetmap bietet keine API Dienste an. Es gibt zwar die Möglichkeit, mithilfe von Leaflet, welches eine JavaScript-Bibliothek ist, eigenen API Dienste zu programmieren, dies ist jedoch mit einem sehr hohen Aufwand und einer sehr hohen Komplexität verbunden. Ein weiterer Nachteil von OpenStreetMap ist es, dass es auf Benutzerdaten aufbaut. Das bedeutet, Nutzer sind dafür verantwortlich, die Karte immer auf dem aktuellsten Stand zu halten. Somit ist dieser Kartendienst wenig zuverlässig. Ein weiterer Nachteil wäre, dass Daten für alle Benutzer zugänglich sind. Informationen über OpenStreetMap wurden aus der Quelle xy () entnommen.



Abbildung 2.12: Open Street Map Logo

## 2.4.2 Geoinformationssystem

Die aufwendigste Option ist es, die Energiesysteme mittels GPS (Global Positioning System) zu orten und diese dann auf der Seite visuell darzustellen. In diesem Falle müsste das Galileo, welches das Europäische GPS System ist, genutzt werden. Galileo ist ein offenes, kostenloses Signal, welches es ermöglicht, den Sender auf bis zu zwei Meter genau zu orten. Ein solcher GPS Sender würde auf circa zehn bis zwanzig Euro pro System oder Technologie kosten. Die gesendeten Signale können mithilfe eines Empfängers ausgewertet werden.

## 2.4.3 CSS-System

Eine weitere Option ist es, die Energiesysteme und Technologien mittels CSS auf der Website darzustellen. Eine Möglichkeit wäre es, die Koordinaten der Systeme in die Datenbank zu schreiben, diese dann in Laravel auszulesen und auf einem selbst erstellten Hintergrund einzuziehen. Dieses System könnte in Kombination mit einem Geoinformationssystem verwendet werden. Dies könnte dann folgendermaßen aussehen:

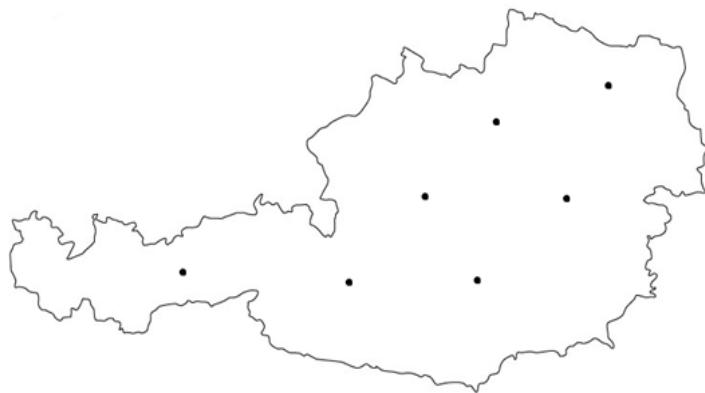


Abbildung 2.13: Beispiel eines CSS-Systems

## 2.4.4 Auswahl des Anbieters

Bei der Wahl des Anbieters, musste der Auftraggeber Best GmbH berücksichtigt werden.

Folgende Punkte muss der Anbieter erfüllen:

- leichte Implementierung
- es sollen alle gewünschten Funktionen umsetzbar sein
- es soll möglichst ohne viel extra Aufwand möglich sein
- es soll nicht kostspielig sein

Um die Anbieter auf die gewünschten Anforderungen zu überprüfen, wurde die Tabelle 2.5 angefertigt. Daten für diese Tabelle wurden den Quellen ... entnommen.

Anbieter	Kosten	API's vorhanden	Umsetzbarkeit(Zeit)	Vorteile und Nachteile
Google Maps	gering	ja	ja	viele API's, die die Arbeit erleichtern; einfache Einbindung auf der Website; performant; großflächig aufgeschlossen; Daten nur für den Betreiber selbst einsehbar
Open Street Map	keine	nein	nein	keine API's vorhanden; viel mehr Aufwand, um das Gleiche wie Google Maps zu bieten; unperformant; keine großflächige Aufschließung; Daten sind für Jeden einsehbar
Geoinformationssystem	sehr hohe	nein	nein	genaue Ortung der Systeme; sehr preisintensiv; sehr zeitintensiv und komplex
CSS-System	keine	nein	ja	keine API's; sehr viel Aufwand; erfüllt den gleichen Zweck wie eine Karte

Tabelle 2.5: Vor- und Nachteile der Anbieter

Auf Basis der Tabelle 2.5, einigten sich der Auftraggeber und das Projektteam darauf, einen Kartendienst zu verwenden, da ein Geoinformationssystem zu anspruchsvoll gewesen wäre und ein CSS-System genau das Gleiche wie eine Karte wäre, nur mit mehr Aufwand verbunden ist. Daraufhin wurde vom Projektteam das Dokument “Google Maps vs Open Street Map” erstellt. Dieses Dokument ist auf der DVD der Diplomarbeit ersichtlich. Nach gemeinsamer Begutachtung dieses Dokuments und dem Abwegen der Vor- und Nachteile jedes Anbieters, haben sich der Auftraggeber und das Projektteam auf die Verwendung von Google Maps geeinigt. Google Map überwiegt klar in den Vorteilen gegenüber Open Street Map. Es ist zwar nicht kostenfrei, jedoch wird in der Regel bei normaler Benutzung der Website nie das Pensum von 300 Euro überschritten.

## 2.5 Berechtigungssystem Benutzer

Im folgenden Abschnitt wird näher auf das Benutzerverwaltungssystem eingegangen sowie auf die vorhandenen Benutzerrollen, die einem Benutzer zugeteilt werden können.

### 2.5.1 Benutzerrollen

Die Weboberfläche bietet für deren Besucher ein rollenbasiertes Benutzersystem an, um die Funktionen für jeden Benutzer zu deklarieren. Das rollenbasierte Benutzersystem unterscheidet zwischen folgenden Rollen:

- Administrator
- Mitarbeiter
- Öffentlicher Benutzer

#### 2.5.1.1 Administrator

Der Administrator-Benutzer ist der höchste Benutzer von allen. Er darf Energiesysteme sowie Energietechnologien erstellen und alle anderen vorhandenen bearbeiten und löschen. Zusätzlich dazu hat er Einsicht in sämtliche Grafana-Statistiken. Diese Benutzerrolle hat das Recht, neue Benutzer auf der Weboberfläche zu registrieren und vorhandene zu löschen, was für alle anderen Benutzer nicht möglich ist.

#### 2.5.1.2 Mitarbeiter

Die Rolle „Mitarbeiter“ darf auf der Weboberfläche neue Energiesysteme sowie Energietechnologien erstellen. Die von ihm erstellten Energiesysteme sowie Energietechnologien kann er bearbeiten oder löschen, und er hat auf diesen ebenso die Berechtigung auf Einsicht der Statistiken. Andere Mitarbeiter dürfen seine Energiesysteme und Energietechnologien nicht bearbeiten oder löschen und haben keinen Zugriff auf dessen Statistiken mit Ausnahme des Administrators. Ein Mitarbeiter-Benutzer darf somit nur seine selbst erstellten Energiesysteme und Energietechnologien verwalten. Der Mitarbeiter-Benutzer hat nicht die Berechtigung, neue Benutzer zu registrieren oder vorhandene zu löschen.

### 2.5.1.3 öffentlicher Benutzer

Dieser Benutzer hat die geringste Berechtigung und tritt in Kraft, wenn man nicht auf der Weboberfläche angemeldet ist. Dieser Benutzer darf keine Energiesysteme sowie Energietechnologien erstellen, bearbeiten oder löschen. Zusätzlich hat der öffentliche Benutzer keine Einsicht auf sämtliche Grafana-Statistiken. Der Zugang zu der Benutzerverwaltung ist ebenso nicht erreichbar. Dieser Benutzer sieht lediglich die öffentlichen Daten, die für jedes Energiesystem und jede Energietechnologie preisgegeben werden.

## 2.5.2 Berechtigungen in Laravel

Im folgenden Abschnitt wird erklärt, wie in Laravel die zuvor genannten Benutzerrollen unterschieden werden. Dafür bietet Laravel Authentifizierungsrichtlinien. Dabei kann mit den Befehlen @auth und @guest überprüft werden, ob der aktuelle Benutzer authentifiziert oder ein Guest ist. Je nach Authentifizierung und Rolle hat der Benutzer unterschiedliche Rechte sowie Funktionen zur Verfügung. Für genauere Informationen, wie eine Berechtigungsüberprüfung in Laravel umgesetzt werden kann, siehe Quelle x.y.

## 2.6 Ui/Ux Design

Im Wort Ui/Ux Design steht das Ui für User Interface und Ux für User Experience. Es handelt davon, wie eine Website von einem Nutzer aufgefasst und wahrgenommen wird. Es beschreibt die Wirkung, die das Design auf einen Nutzer hat. Genauer beschreibt die User Experience wie sich ein Nutzer auf der Website fühlt oder wie er auf ihr navigieren kann. Ziel ist es, dem Benutzer ein möglichst entspanntes Gefühl zu geben und ihm bei Allem zu helfen. Das User Interface beschäftigt sich vielmehr mit der Website an sich. Man macht sich hierbei Gedanken, ob der Benutzer das Design ansprechend finden könnte, oder ob überall die idealen Elemente oder Elementgrößen gewählt wurden. Eine weitere Frage, die man sich im User Interface überlegt, ist was wirkt auf den Benutzer positiv und was negativ. Einige Methoden das Ui/Ux Design umzusetzen sind:

- erstellen eines Wireframes
- Umfragen an Usern durchführen
- Fragebögen auf der Seite selbst
- erstellen einer Übersicht mit den Anforderungen der Benutzer

Nützliche Programme zum Erstellen der besagten Dokumente sind folgende:

- Kissmetrics
- Figma
- SimilarWeb

Informationen über Ui und Ux Design wurden der Quelle xy entnommen.

## 2.6.1 Wireframe

Ein Wireframe hilft dem Design Team eine grobe Struktur der Website vorzudefinieren. Bei der Erstellung eines Wireframes wird der visuelle Design Part komplett vernachlässigt. Bei einem Wireframe wird sich auf den Aufbau und die Position der Elemente konzentriert. Es ist meist in schwarz und weiß gehalten, da Farben bei diesem Aufbauprozess nur behindern würden.

## 2.6.2 Persona

Eine Persona dient dazu, die Bedürfnisse und Ziele der Zielgruppe zu erfassen. Mithilfe der Persona kann auf Wünsche und Bedürfnisse der Benutzer während der Entwicklung eingegangen werden. Empfohlen wird pro Zielgruppe um die fünf bis sechs Personen die das Produkt testen. Erstellt werden sie dann auf Grundlage von:

- Interviews mit dem Benutzer
- Benutzertests
- Umfragen

## 2.7 Template Layout

Laravel bietet die Möglichkeit, mithilfe einer Layout Datei, Codeabschnitte auf mehreren Seiten zu verwenden. Hauptsächlich wird diese Funktion genutzt, um statische Elemente, die auf jeder Unterseite gleich sind, nur einmal zu programmieren. Ein gutes Beispiel wäre der Header oder auch der Footer, da diese Elemente auf den Unterseiten nicht variieren.

### 2.7.1 Platzhalter Yield

In der Layout Datei kann @yield verwendet werden, um Platz zu halten. Dies ermöglicht es, beispielsweise den Titel der Website individuell für jede Unterseite zu setzen. Der Platzhalter wird in der Layout Datei folgendermaßen konfiguriert:

Quellcode 2.5: Platzhalter in der Layout Datei Konfigurieren

```
1 <title>@yield('title')</title>
```

Möchte man diesen Platzhalter dann auf einer der Unterseiten, welche die Layout Datei verwenden, setzen, ist das folgendermaßen möglich:

Quellcode 2.6: Platzhalter auf einen Wert setzen

```
1 @section('title', 'titlename')
```

Das Projektteam verwendet dieses Statement auf jeder Unterseite, um den Titel zu setzen.

## 2.7.2 Sections

Sections ermöglichen es, die Layout Datei in Abschnitte zu unterteilen. Der Beginn einer Section wird mit folgendem Befehl gekennzeichnet:

Quellcode 2.7: Beginn einer Section kennzeichnen

```
1 @section('sectionname')
```

Um die Section wieder zu schließen, muss folgendes Statement benutzt werden:

Quellcode 2.8: Ende einer Section kennzeichnen

```
1 @endsection
```

Die Einteilung des Codes in Section erlaubt es, dieses Codestück auf einer anderen Seite mit nur einem Stichwort einzubinden.

## 2.7.3 Einbindung der definierten Sections

Um diese im Layout File definierten Sections im Code einbinden zu können, müssen einige Einstellungen vorgenommen werden. Am Anfang einer neuen Seite muss dieser Code eingefügt werden:

Quellcode 2.9: Einbindung der Layout Datei

```
1 @extends('layoutfilename')
```

Damit wird der Seite mitgeteilt, dass die angegebene Seite als Layout Vorlage verwenden werden soll. Wenn diese Zeile vorhanden ist, können mittels `section('sectionname')` die im Layout File definierten Sections eingebunden werden. Wichtig ist, jedesmal nach dem `@section` auch ein `@endsection` einzugeben, sonst wird die Section nicht angezeigt. Zur Ergänzung der Section kann man den zu ergänzenden Code einfach zwischen dem `@section` und dem `@endsection` platzieren. Dieser wird dann zum bereits vordefinierten Code ergänzt.

## 2.8 Laravel Befehle

Laravel bietet die Möglichkeit, viele Dinge mithilfe der Konsole Models, Controller oder Views, automatisch generieren zu lassen. In den folgenden Unterkapiteln werden die Wichtigsten dieser Befehle aufgelistet und kurz erklärt.

### 2.8.1 Migration Befehle

Um eine neue Migration zu erstellen, muss folgendes Kommando in der Commandline ausgeführt werden:

Quellcode 2.10: Erstellen einer neuen Migration

```
1 php artisan make:migration migration_name
```

Wenn die Migration erfolgreich erstellt und initialisiert wurde, kann man mit folgendem Kommando die up() Methode aller Migrations aufrufen:

Quellcode 2.11: Aufruf der up() Methode in der Migration

```
1 php artisan migrate
```

Nach Ausführung dieses Kommandos sind alle Tabellen in der Datenbank so wie gewünscht erstellt. Um alle ausgeführten Migrations anzeigen zu lassen, muss dieses Kommando verwendet werden:

Quellcode 2.12: Befehl um den Status der Migrations zu bekommen

```
1 php artisan migrate:status
```

Wenn die zuletzt erstellten Tabellen rückgängig gemacht werden sollen, gibt es die Möglichkeit, folgendes Kommando auszuführen:

Quellcode 2.13: Migrations rückgängig machen

```
1 php artisan migrate:rollback
```

Weitere nützliche Befehle sind unter der Quelle [xy \(\)](#) zu finden.

### 2.8.2 Seeder und Factory Befehle

Um einen Datenbank Seeder zu erstellen, ist folgendes Kommando notwendig:

Quellcode 2.14: Datenbank Seeder erstellen

```
1 php artisan make:seeder seeder_name
```

Wenn eine Factory erstellt werden soll, gibt es folgende Möglichkeit:

Quellcode 2.15: Datenbank Factory erstellen

```
1 php artisan make:factory factory_name
```

Um den Seeder nach Initialisierung auszuführen, gibt es folgende Möglichkeit:

Quellcode 2.16: Datenbank Seeder ausführen

```
1  php artisan db:seed
```

Weitere nützliche Befehle und Individualizations Möglichkeiten sind unter der Quelle xy() verzeichnet.

### 2.8.3 Model und Controller Befehle

Um ein neues Model zu erstellen, verwendet man dieses Kommando:

Quellcode 2.17: Erstellen eines neuen Models

```
1  php artisan make:model model_name
```

Um einen normalen Controller anzulegen, gibt es folgendes Kommando:

Quellcode 2.18: Erstellen eines neuen Controllers

```
1  php artisan make:controller controller_name
```

Es gibt ergänzend die Möglichkeit, einen Resource Controller anzulegen. Ein Resource Controller hat den Vorteil, dass er vorgenerierte Methoden besitzt, wie zum Beispiel:

- index()
- create()
- store()
- show()
- edit()
- update()
- destroy()

Erstellt wird ein Resource Controller folgendermaßen:

Quellcode 2.19: Erstellen eines neuen Resource Controllers

```
1  php artisan make:controller controller_name --resource
```

Genauere Informationen zu der Erstellung von Models und Controllern finden Sie unter der Quelle xy().

## 2.8.4 Starten des Laravel Entwicklungs Servers

Laravel bietet die Möglichkeit, wenn PHP auf dem Entwicklungssystem installiert ist, einen PHP Development Server zu hosten und das Projekt über den Localhost anzuzeigen. Um den Server zu starten, muss folgende Zeile eingeben werden:

Quellcode 2.20: Start des Laravel Entwicklungs Servers

```
1  php artisan serve
```

Nach Ausführen dieses Kommandos kann man das Projekt unter <http://localhost:8000> aufgerufen.

## 2.8.5 Befehle nach dem Git Pull

Bei Team Projekten in Laravel ist Git die beste Lösung. Jedoch kommt es oft zu Komplikationen beim Herunterladen des Projekts durch Git. Es gibt aber einige Befehle, die es ermöglichen, das Laravel Projekt komplikationsfrei zu starten. Dieser Kommand sorgt dafür, dass der APP\_KEY in der .env Datei richtig gesetzt wird:

Quellcode 2.21: APP\_Key generieren

```
1  php artisan key:generate
```

Um sicher zu stellen, dass alle nötigen Pakete installiert wurden, kann folgender Befehl verwendet werden:

Quellcode 2.22: Alle nötigen Pakete installieren

```
1  composer update
```

Nach Durchführung dieser Schritte sollte sich das Programm problemlos starten und verändern lassen.

## 2.9 Routen in Laravel

Laravel verfügt über ein ausgedehntes Routing System über welches Views aber auch ganze Gruppen von Elementen angesprochen und angezeigt werden können. Diese Routen werden in der Datei “web.php” angelegt und verwaltet.

### 2.9.1 Ressource Routen

Mit der Ressource Route wird ein ganzer Controller angesprochen, so können über diese Route alle Methoden in diesem Controller angesprochen und verwendet werden. Im Produkt wird jeder Controller, welcher wiederum für jede Tabelle vorhanden ist, durch eine Ressource Route adressiert. Dieser Vorgang wird in Abbildung 2.14 veranschaulicht.

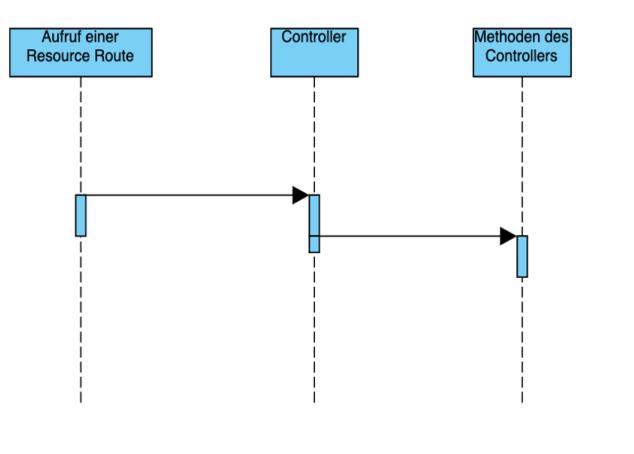


Abbildung 2.14: Aufbau einer Ressource Route

### 2.9.2 GET Routen

Mit diesem Typ von Routen lässt sich eine bestimmte Methode in einem ausgewählten Controller adressieren. Außerdem ist es möglich in einer GET Route einen Parameter mitzugeben welcher dann in der ausgewählten Methode im Controller verarbeitet wird. Im Produkt wird dieser Typ von Methoden verwendet, um beispielsweise die Startseite anzuzeigen. Weiters werden sie verwendet, um bestimmte Methoden in Controllern, zB. löschen von Energiesystemen, mit der Id des ausgewählten Objektes auszuführen. Am meisten kommt dieser Typ von Routen beim Erstellen, Editieren oder Löschen von Datenbankeinträgen zum Einsatz, da hier ein gewisser Eintrag mit dem mitgelieferten Parameter einer Route adressiert werden kann. Der Aufbau dieser Route wird in Abbildung 2.15 verdeutlicht.

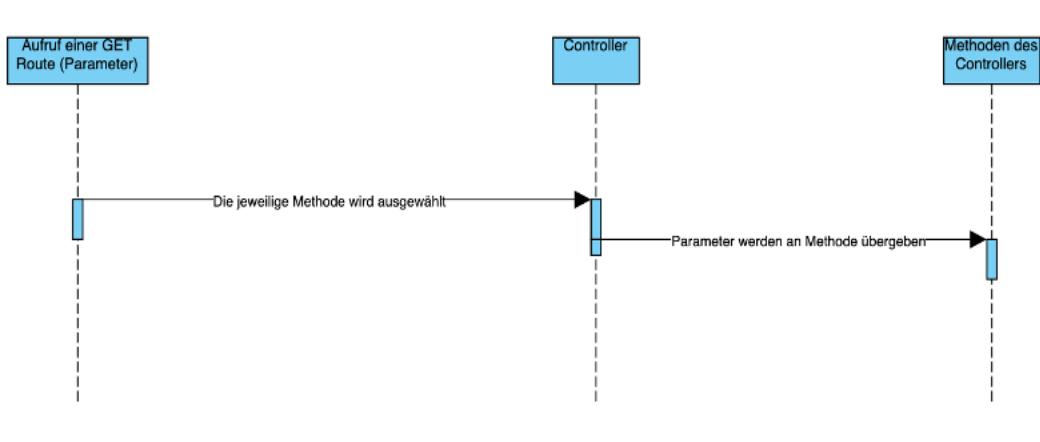


Abbildung 2.15: Aufbau einer GET Route

### 2.9.3 Auth Routen

Sobald in Laravel das Authentifizierungssystem aktiviert wird, werden diese Routen automatisch erstellt. Diese Routen adressieren, die ebenfalls durch Laravel erstellen, auth Controller in der die Benutzeranmeldung sowie Registrierung bearbeitet wird.

## 2.10 MVC

Das Produkt verwendet das von Laravel bereitgestellte Design Pattern MVC (Model, View, Controller), welches in der Abbildung 2.16 dargestellt wird.

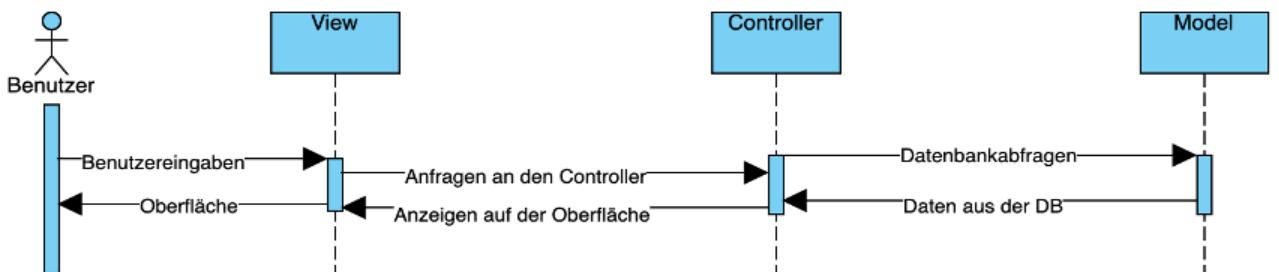


Abbildung 2.16: MVC Design Pattern

### 2.10.1 Model

Für jede durch eine Migration (Kapitel 3.2.2) erstellte Tabelle gibt es ein Model, in diesem wird wie der Name suggeriert, das Datenmodell abgebildet. Durch diesen Ansatz kann von jedem Model eine Instanz erstellt werden und so die entsprechende Tabelle in einem beliebigen Programmabschnitt verwendet werden. Weiters werden in den Models die Relationen zwischen den einzelnen Tabellen beschrieben.

### 2.10.2 View

Das Produkt baut auf acht Views auf. In diesen Views werden die Daten, welche von den Controllern übergeben werden mithilfe von DataTables (Kapitel 3.6) dem Benutzer angezeigt. Weiters gibt es in den Views verschiedene Möglichkeiten der Benutzerinteraktion.

### 2.10.3 Controller

In den Controllern wird der Datenweg zwischen den Views, also den Benutzerinteraktionen und der Datenbank gesteuert. Für jede, durch eine Migration in der Datenbank erstellte Tabelle, gibt es einen Controller, welcher das Speichern, die Ausgabe, das Editieren sowie das Löschen von einzelnen Tupeln aus der Datenbank steuert. Weiters werden in den "store()" und "destroy()" Methoden des EnTechControllers sowie des EnSysControllers die API Aufrufe an die Grafana HTTP API mit den vom Benutzer eingegebenen Daten ausgeführt.

# Kapitel 3

## Ergebnisdokumentation

Dieser Abschnitt beinhaltet eine Dokumentation der verwendeten Werkzeuge zur Umsetzung der einzelnen Funktionen. Es werden alle behandelten Teilbereiche erläutert.

### 3.1 Laravel

In diesem Kapitel werden die Ergebnisse und Erfahrungen dokumentiert, welche sich im Laufe der Produktentwicklung mit dem Laravel Framework heraus kristallisiert haben.

#### 3.1.1 Installation

Die Installation von Laravel erfolgt mithilfe des PHP Installations Tools Composer. Dieser installiert hierzu die Paketquellen Lokal auf dem System. Beim Erstellen eines neuen Laravel Projekts werden die von Composer installierten Quelldateien in das Verzeichnis des neu zu erstellenden Projekts kopiert und installiert. Im nachfolgenden Link ist mehr über die Installation von Laravel zu lesen:

Unter der Quelle xy () ist mehr über Laravel zu lesen.

#### 3.1.2 Bootstrap Einbindung

Für das Design des Front-Ends wird die CSS Bibliothek “Bootstrap” (Kapitel 2.3.6.1) verwendet. Hierzu werden die Quelldateien von Bootstrap durch Composer lokal abgespeichert und durch den Laravel Befehl "php artisan ui bootstrap" dem Laravel Projekt hinzugefügt. Laravel bietet die Möglichkeit beim Hinzufügen von Bootstrap mit der Option “– auth” direkt die Beispieleiten (Login, Registrieren) für den Authentifizierungsvorgang zu erstellen. Im nachfolgenden Link ist mehr über die Installation von Bootstrap zu lesen: Dokumentation der Installation

### 3.1.3 Grafana Einbindung

Das Produkt kommuniziert mit dem Grafana Server über die von Grafana bereitgestellte HTTP API. Laravel verfügt hierzu über das Paket “guzzle-http”, welches es ermöglicht, mithilfe von wenigen Zeilen PHP Code einen HTTP Aufruf an eine bestimmte Quelle abzusenden.

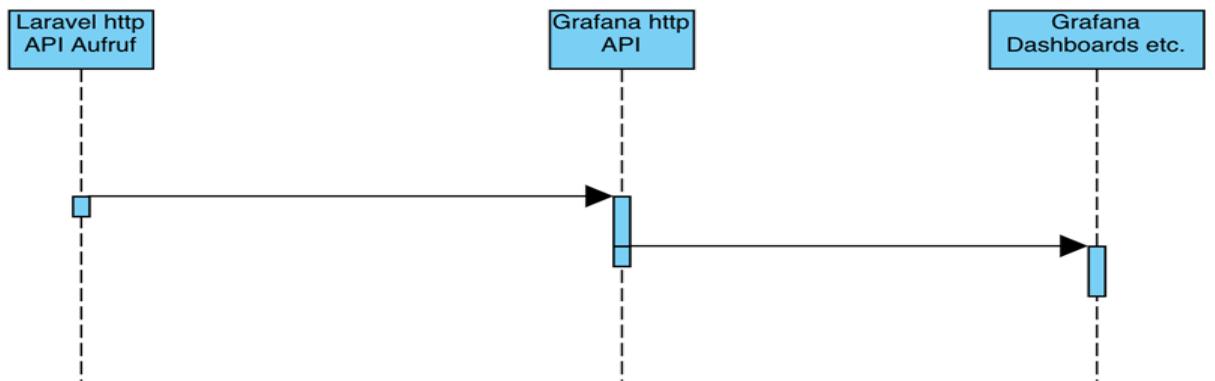


Abbildung 3.1: Grafana Einbindung

## 3.2 Datenbankanbindung in Laravel

Um die Datenbank an das Programm anzubinden, müssen einige Änderungen in der .env Datei vorgenommen werden. Genauere Information zur .env Datei befinden sich im Kapitel 2.3.6.1. Welche Konfigurationen genau getätigter werden, können in den nachfolgenden Kapiteln nachgelesen werden.

### 3.2.1 Datenbank Anmeldeinformationen

Eine Änderung war es, die Datenbank Anmeldeinformationen auf die richtigen Einstellungen zu setzen. Welche Attribute gesetzt werden, und was genau bei diesen Eingetragen werden muss, ist in Tabelle 3.1 ersichtlich. Eine mögliche Konfiguration ist in Tabelle 3.2 ersichtlich.

Attribut	Was soll eingetragen werden
DB_CONNECTION	Hier muss die Zugriffsvariante auf die Datenbank eingetragen werden
DB_HOST	Hier muss die Domain des Datenbankservers, beispielsweise 127.0.0.1, wenn der Datenbankserver über den Localhost bereitgestellt wird, eingeben werden
DB_PORT	Hier der Port über welchen zugegriffen werden soll
DB_DATABASE	Hier der Name der Datenbank , auf welche zugegriffen werden soll
DB_USERNAME	Der Benutzer, mit welchem zugegriffen werden soll
DB_PASSWORD	Das Passwort zum Benutzer

Tabelle 3.1: Datenbank Anmeldeinformationen

Attribut	Was bei diesem Produkt eingetragen wurde
DB_CONNECTION	mysql
DB_HOST	127.0.0.1
DB_PORT	3306
DB_DATABASE	laravel
DB_USERNAME	root
DB_PASSWORD	Hier wurde nicht eingetragen

Tabelle 3.2: Datenbank Anmeldeinformationen Beispiel

### 3.2.2 Mail Server Konfigurationen

Um auf der Weboberfläche das Passwortzurücksetzen mittels E-Mail zu ermöglichen, muss Laravel eine E-Mail versenden. Dazu braucht es die nötigen Konfigurationen in der .env Datei. In der Tabelle 3.3 ist ersichtlich, welche Attribute gesetzt werden müssen, und welche Bedeutung diese haben.

Attribut	Was soll eingetragen werden
MAIL_MAILER	Hier muss das Protokoll eingetragen
Mail_HOST	Hier wird die Host Domain eingegeben, also welcher SMTP Server verwendet werden soll
MAIL_PORT	Der Port über welchen zugegriffen werden soll
MAIL_USERNAME	Hier muss die E-Mail Adresse, welche zum Absenden der Mail verwendet wird, eingetragen werden
MAIL_PASSWORD	Das dazugehörige Passwort zur Email
MAIL_ENCRYPTION	Hier gibt man an, mit welchem Verfahren verschlüsselt werden soll
MAIL_FROM_ADDRESS	Die Adresse, welche beim Empfänger als Sender E-Mail angezeigt wird eintragen
MAIL_FROM_NAME	Der Name, der dem Empfänger angezeigt wird

Tabelle 3.3: Mail Server Konfigurationen

Bei diesem Produkt hat die Konfiguration folgendermaßen ausgesehen:

Attribut	Was bei diesem Produkt eingetragen wurde
MAIL_MAILER	smtp
Mail_HOST	smtp.gmail.com
MAIL_PORT	465
MAIL_USERNAME	marcelentner2@gmail.com
MAIL_PASSWORD	Das dazugehörige Passwort zur Email
MAIL_ENCRYPTION	ssl
MAIL_FROM_ADDRESS	marcelentner2@gmail.com
MAIL_FROM_NAME	Best GmbH

Tabelle 3.4: Mail Server Konfigurationen Beispiel

### 3.2.3 Migrations

Die Migrations wurden dazu verwendet, um Tabellen zu gestalten. In ihnen wurden alle nötigen Attributnamen und deren Datentypen konfiguriert. Das Produkt verwendet folgende von Laravel automatisch erzeugten Migrations für die Benutzerauthentifizierung:

- 2014\_10\_12\_000000\_create\_users\_table.php
- 2014\_10\_12\_100000\_create\_password\_resets\_table.php
- 2019\_08\_19\_000000\_create\_failed\_jobs\_table.php
- 2019\_12\_14\_000001\_create\_personal\_access\_tokens\_table.php

Zusätzlich zu diesen bereits erstellten Migrations wurde für jede Energietechnologieart eine eigene Migration angelegt. Darüber hinaus wurden Templates<sup>1</sup> für ein Energiesystem und eine Energietechnologie erstellt. Im Energietechnologien Template sind alle Attribute, die bei jeder Energietechnologie gleich sind. Alle individuellen Attribute werden dann mithilfe der eigenen Migrations abgebildet.

## 3.3 Datenbankdesign

In diesem Kapitel der Diplomarbeit wird das Redesign der Datenbank und der Relationen zwischen den einzelnen Tabellen behandelt.

### 3.3.1 Erstellen eines neuen Schemas

Für das Produkt wurde ein neues vom alten unabhängiges Datenbankschema entwickelt. Hauptgründe hierfür waren die im Kapitel 2.1 festgestellten Mängel und Designfehler. Im neuen Design wurde darauf geachtet alle Design Richtlinien welche für Relationale Datenbanken gelten einzuhalten. Das neue Datenbankschema besteht aus 21 Tabellen für Individuelle Energietechnologien sowie aus einer Tabelle für Benutzer, einer Tabelle für erstellte Energiesysteme und einer Tabelle für die Generalisierung einiger Attribute von Energietechnologien.

#### 3.3.1.1 ER-Modell

Um das Datenbankdesign zu veranschaulichen und Unklarheiten mit dem Auftraggeber vorzubeugen, wurden mithilfe des Programms “MySQL Workbench” ein EER (Extended Entity Relationship) Diagramm erstellt. In diesem Diagramm ist jede Tabelle mit ihren Attributen sowie Primär und Fremdschlüsseln abgebildet. Weiters sind auch die einzelnen Relationen zwischen einer oder Mehreren Tabellen veranschaulicht. Das vollständige EER Modell der Datenbank ist in der Abbildung 3.2 zu sehen.

---

<sup>1</sup>ein Template ist ein Grundbauplan

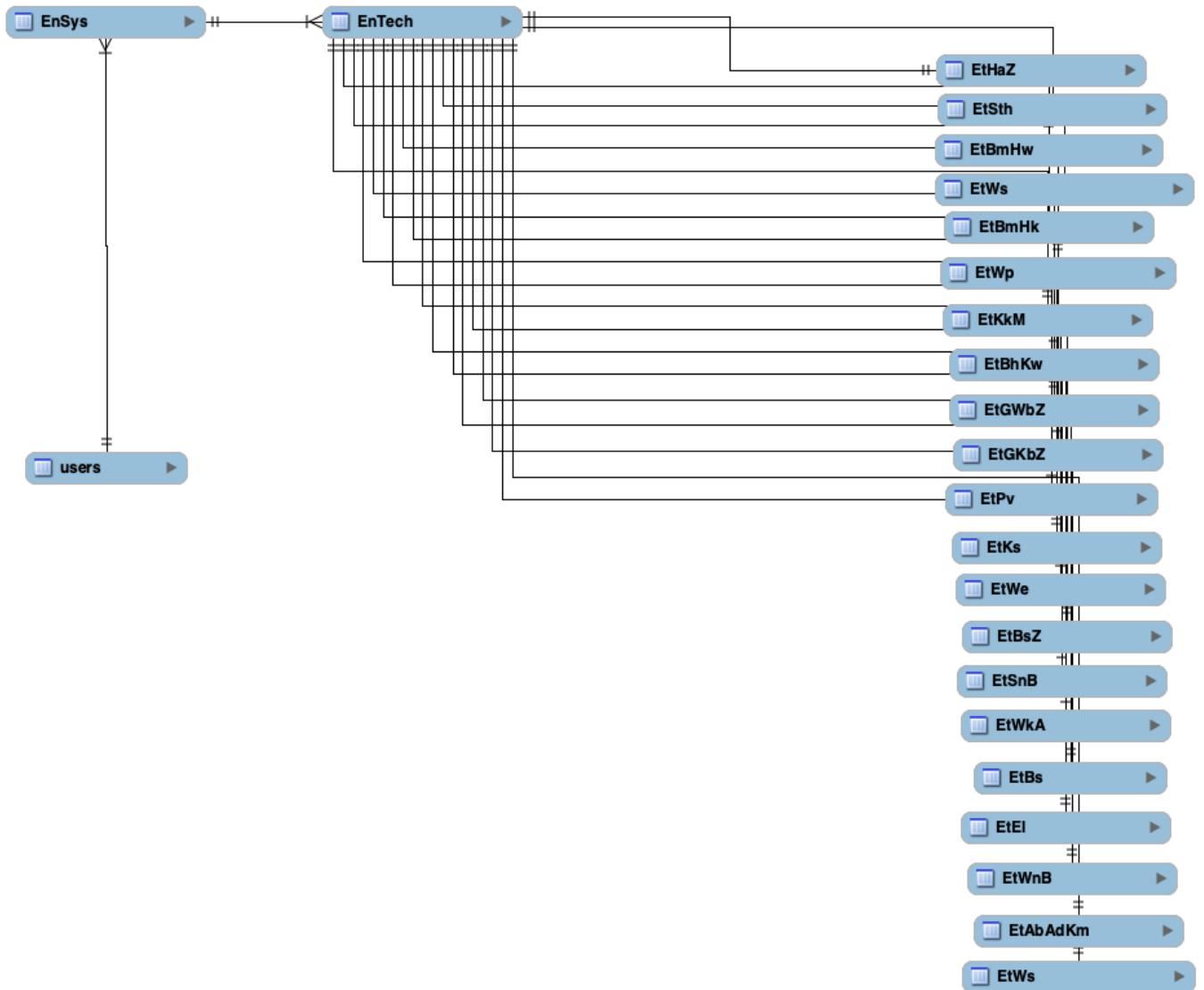


Abbildung 3.2: ER Model

### 3.3.1.2 Fremdschlüssel

Wie im Kapitel 3.4.1.1 wurden zwischen den Tabellen verschiedene Relationen verwendet. Zwischen den Tabellen "users" und "EnSys" besteht eine 1:N Relation. Dies bedeutet, dass in jedem Eintrag in der "EnSys" Tabelle der Primärschlüssel des Benutzers welcher, den Eintrag durchführt, als Fremdschlüssel gespeichert wird. Zwischen den Tabellen "EnSys" und "EnTech" besteht ebenfalls eine 1:N Relation. In diesem Fall werden die Primärschlüssel der "users" sowie der "EnSys" Tabelle als Fremdschlüssel in der "EnTech" Tabelle gespeichert. Zwischen der "EnTech" sowie jeder der 21 Tabellen für individuelle Energietechnologien besteht eine 1:1 Beziehung. Es gibt also zu jedem Eintrag in der "EnTech" Tabelle genau einen Eintrag in einer der 21 Tabellen für Energietechnologien. Hierbei wird der Primärschlüssel des Eintrages der "EnTech" Tabelle als Fremdschlüssel im korrespondierenden Eintrag in einer der 21 Tabellen für Energietechnologien gespeichert.

## 3.4 Corporate Design

Vom Projektteam wurde ein Corporate Design Manual erstellt. Dieses befindet sich auf der DVD der Diplomarbeit. In diesem Corporate Design Manual wurden folgende Themen bearbeitet:

- Oberfläche
- Definierte Farben
- Verwendung der definierten Farben
- Überschriften
- Buttons
- Interaktionsfarben
- Schriftarten
- Schriftgrade
- Logo
- Verwendete Icons
- Bedeutung der Icons
- Icons mit Funktionalitäten
- Website Design

Auf die wichtigsten Elemente wird in den nachfolgenden Kapiteln eingegangen.

### 3.4.1 Vorschläge

Nach Absprache der Grundanforderung des Produkts wurden, mithilfe der Adobe xd Software, drei Designvorschläge verfasst. Diese sind ebenso auf der DVD der Diplomarbeit einsehbar. Um diese aussagekräftig zu vergleichen, wird von jedem Vorschlag, die Energiesystemseite visuell dargestellt. Dies ist in den drei nachfolgenden Abbildungen 3.3, 3.4, 3.5 ersichtlich.

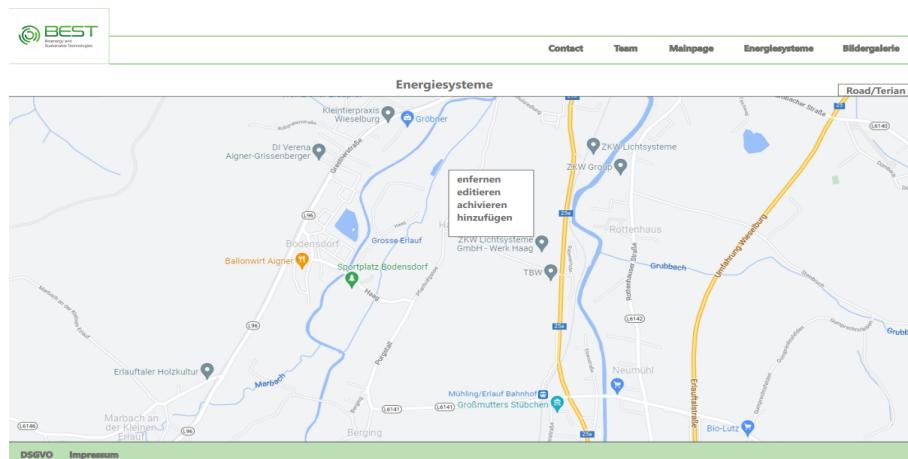


Abbildung 3.3: Design Vorschlag 1



Energiesysteme				
Name	Art	Breitengrad	Längengrad	Leistung
Pöchacker	PV-Anlage	27.97	31.93	81 kW
Pöchacker	PV-Anlage	27.97	31.93	81 kW
Pöchacker	PV-Anlage	27.97	31.93	81 kW

DSGVO Impressum

Abbildung 3.4: Design Vorschlag 2

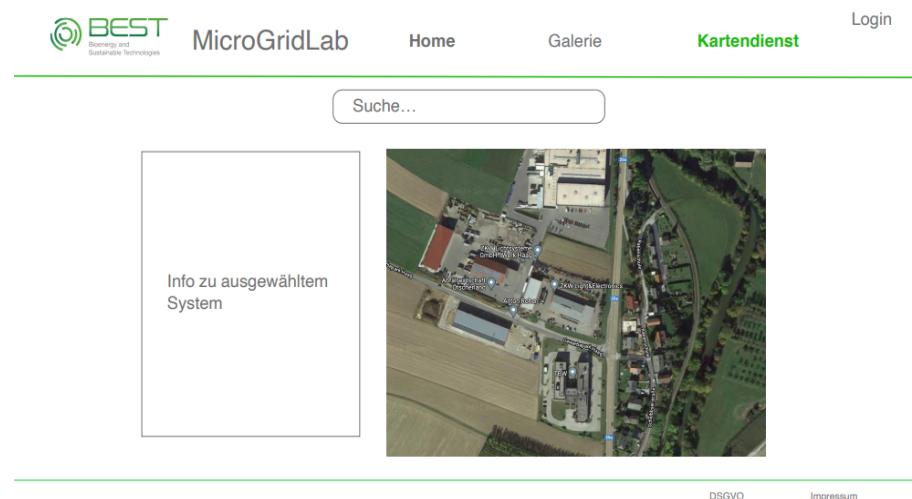


Abbildung 3.5: Design Vorschlag 3

### 3.4.2 Änderungsvorschläge

Die oben ersichtlichen Design Vorschläge wurden dem Auftraggeber über Skype präsentiert. Dieser favorisierte die Vorschläge 3.3 und 3.4. Von Seiten des Auftraggebers kamen viele Verbesserungsvorschläge und Ideen. Einige davon sind:

- Die Liste soll neben der Map platziert werden.
- Es soll folgende Unterseiten geben:

Home

Energiesysteme

Galerie

- Es soll Icons geben, welche es ermöglichen, Energiesysteme und Energietechnologien in der Liste zu löschen, bearbeiten oder einzusehen.
- Es sollen die Farben aus der CSS Datei der bereits vorhanden Website extrahiert werden.

### 3.4.3 Finales Design

Nach genauer Definition der Änderungsvorschläge im Kapitel 3.4.2, wurde erneut in Adobe XD ein Design Vorschlag erstellt. Dieser Vorschlag war eine Mischung aus dem Vorschlag 1 und dem Vorschlag 2 mit den gewünschten Änderungen des Auftraggebers. Auch dieser Vorschlag ist auf der DVD der Diplomarbeit einsehbar. Um einen Vergleich zwischen Vorschlag 1, Vorschlag 2 und dem finalen Design zu ermöglichen, wird nachfolgend auch die Energiesystem Seite des finalen Vorschlags visuelle dargestellt.

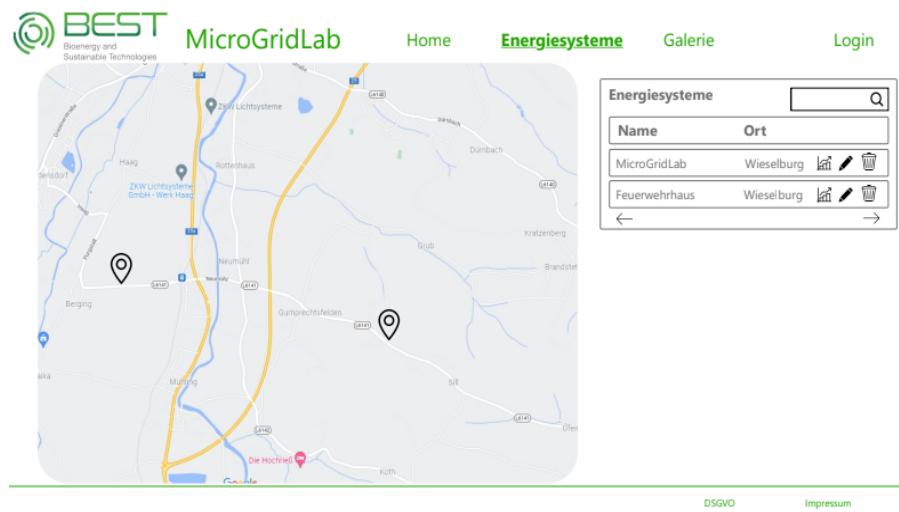


Abbildung 3.6: Finaler Design Vorschlag

Am Vorschlag in 3.6, wurde sich während der Entwicklung gerichtet. Aufgrund laufender Änderungswünsche des Auftraggebers variiert das finale Produktdesign jedoch stark vom finalen Vorschlag. Das finale Design ist im Kapitel 3.5.2 einsehbar.

### 3.4.4 Definierte Farben

Das Produktdesign basiert auf dem Front-End Framework Bootstrap. In Bootstrap werden alle Farben in der Datei "app.css" definiert. Hier ein textueller Ausschnitt, der vom Projektteam verwendeten Farben:

- –bs-primary: #1b8836;
- –bs-blue: #0d6efd;
- –bs-white: #fff;
- –bs-danger: #dc3545;

Auf Wunsch des Auftraggebers wurden jedoch einige Farben aus den CSS Dateien der bereits vorhanden Website extrahiert und bei diesem Produkt verwendet. Folgende Farben wurden extrahiert:

Hex	CMYK	RGB	HSV	HSL
#1b8836	80%, 0%, 60%, 47%	27, 136, 54	135, 80%, 53%	135, 67%, 32%
#f8f9fa	1%, 0%, 0%, 2%	247, 250, 250	80, 1%, 98%	80, 23%, 97%
#e84d3d	0%, 67%, 74%, 9%	232, 77, 61	6, 74%, 91%	6, 79%, 57%
#212529	20%, 10%, 0%, 84%	33, 37, 41	210, 20%, 16%	210, 11%, 15%
#f1f1f1	19%, 10%, 0%, 84%	241, 241, 241	210, 20%, 16%	210, 11%, 15%
#0d6efd	95%, 57%, 0%, 1%	13, 109, 252	216, 95%, 99%	216, 98%, 52%
#21a500	80%, 0%, 100%, 35%	33, 166, 0	108, 100%, 65%	108, 100%, 33%

Alle verwendeten Farben sind im Corporate Design Manuel, auf der DVD der Diplomarbeit unter dem Kapitel definierte Farben ersichtlich.

### 3.4.5 Überschriften

Überschriften werden bei diesem Produkt vorrangig verwendet, um anzuzeigen, über welches Thema der Fließtext, der sich meist direkt unter einer Überschrift befindet, handelt. Bei diesem Produkt wurden Überschriften der Gattung `<h1>`, `<h2>` und `<h3>` verwendet. Aus designtechnischen Gründen, befindet sich jede Überschrift immer direkt in der Mitte der Website. Jede Überschrift, welche außerhalb eines Fließtextes steht, ist in der Farbe #1b8836 eingefärbt und in der Schriftart Smooch Snans formatiert. Überschriften, welche sich in einem Fließtext befinden, werden in der Farbe #212529 eingefärbt. Folgende Überschriften sind beim Produkt vorhanden:

- Intelligente Strom- und Mikronetze
- Entwicklungsfelder und Anwendungsbereiche
- Forschungsprojekt Microgrid Lab
- Datenschutzerklärung
- DSGVO
- Impressum

## Intelligente Strom- und Mikronetze

Die 2017 neu gegründete Area „Smart- und Microgrids“ beschäftigt sich mit der konzeptionellen Planung und Steuerung von dezentralen Energieversorgungsprojekten und Microgrids. Dazu werden theoretische, physikalische und wissenschaftliche Zusammenhänge im Bereich der Planung und Steuerung von Smart- und Microgrids erforscht und experimentell entwickelt. Die angewandten Methoden inkludieren Mixed Integer Linear Programming (MILP) oder linearisierte MILP sowie Model-Predictive-Control(MPC)-Methoden. Eine ganzheitliche Betrachtung (Konzeptionierung und Betrieb) von Multienergiesystemen (Strom, Wärme, Kälte für Gebäude sowie für kommunale und industrielle Anwendungen) stellt eine komplexe Herausforderung dar, die nur durch den Einsatz der genannten Methoden nachhaltig lösbar ist. Das Hauptziel besteht darin, den Grad der Autonomie der Systeme auf allen hierarchischen Ebenen (einzelne Gebäude, Siedlungen, Energiegemeinschaften, Subnetze, Regionen) und in allen Sektoren des Energiesystems zu erhöhen. Dadurch steigt sich die Gesamteffizienz nachhaltig und die überregionale Infrastruktur wird entlastet. Die Arbeiten dienen als Grundlage für die Entwicklung von Werkzeugen, von welchen das gesamte Energiesystem langfristig profitieren soll.

## Entwicklungsfelder und Anwendungsbereiche

» Entwicklung von Methoden und Tools für die Energiesystemplanung und die sektorenübergreifende Energieinfrastruktur sowie die Identifizierung von Schwachstellen » Schaffung einer prädiktiven Steuerung via MPC, die Wechselwirkungen zwischen den Sektoren berücksichtigt und durch den Einsatz von Speichern und Lastverschiebungen mehr Flexibilität und Stabilität im Energiesystem bietet

## Forschungsprojekt Microgrid Lab

In dem Forschungsprojekt Microgrid Lab wird ein Microgrid für kommunale Energiekonzepte in einem realen Umfeld geplant, errichtet, evaluiert und auf wissenschaftlicher Ebene weiterentwickelt. Ziel ist die Etablierung des Forschungslabors für verschiedene Wirtschaftszweige, um Planungs-, Steuerungs-, Integrations- und Kommunikationskonzepte zu entwickeln und für den Markt zu testen. Das betrifft auch sektorenübergreifende Energienetze (Wärme, Strom, Gas, Wasserstoff). Projektinhalte sind die wissenschaftliche Planung, die Inbetriebnahme, ein standardisiertes Monitoring der Verbraucher/Erzeuger (u.a. Biomasse, PV, Batterie, E-Ladestationen, Absorptionskälte), die Entwicklung von Testzyklen, die Weiterentwicklung der Optimierungsalgorithmen und ein Wissenstransfer zu verschiedenen Stakeholdergruppen. Mithilfe entwickelter mathematischer Methoden wurde bereits ein optimales Energieplanungskonzept entwickelt, welches auf Gemeinden übertragen werden kann. Die innovativen Planungs- und Steuerungskonzepte ermöglichen CO<sub>2</sub>- und Kosteneinsparungen von bis zu 90% bzw. 40%.

Abbildung 3.7: Überschriften im Fließtext

## Impressum



Abbildung 3.8: Überschriften außerhalb des Fließtextes

### 3.4.6 Interaktionsfarben

Interaktionsfarben werden verwendet, um eine Aktion zu kennzeichnen. Bei diesem Produkt gibt es zwei verschiedene Interaktionsfarben. Die erste Farbe ist „#21a500“ und die zweite „#f1f1f1“. Die erste Farbe ist ersichtlich, wenn mit dem Mauszeiger über den DataTable<sup>2</sup> gefahren wird. Und die zweite Farbe wird ersichtlich, wenn mit dem Mauszeiger auf das Drop-Down in der Galerie gefahren wird, um ein Energiesystem auszuwählen. Wenn dies geschieht, verfärben sich die einzelnen Auswahlelemente in die Farbe „#f1f1f1“.

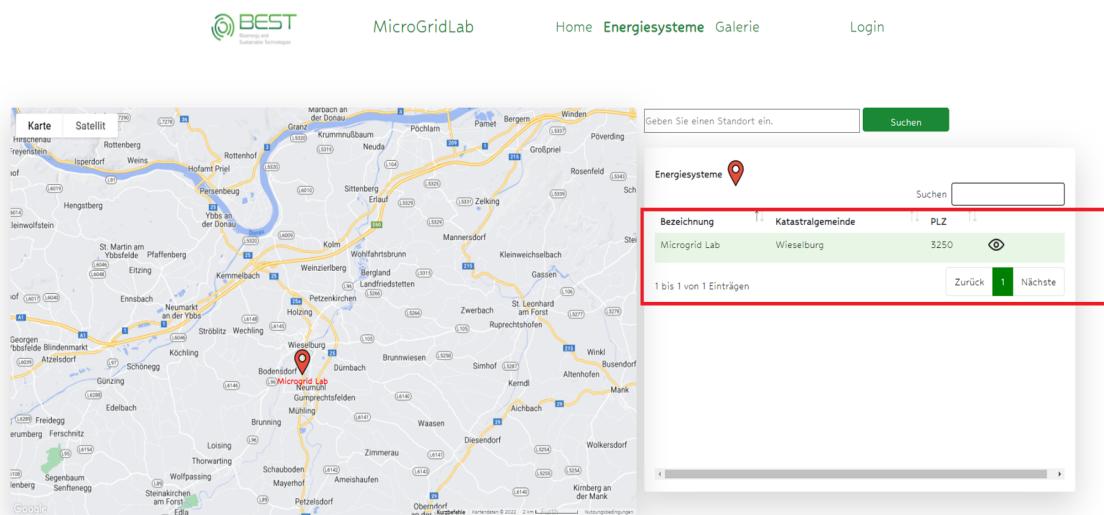


Abbildung 3.9: Interaktion mit dem DataTable

<sup>2</sup>genauer in Abschnitt 3.6 erklärt

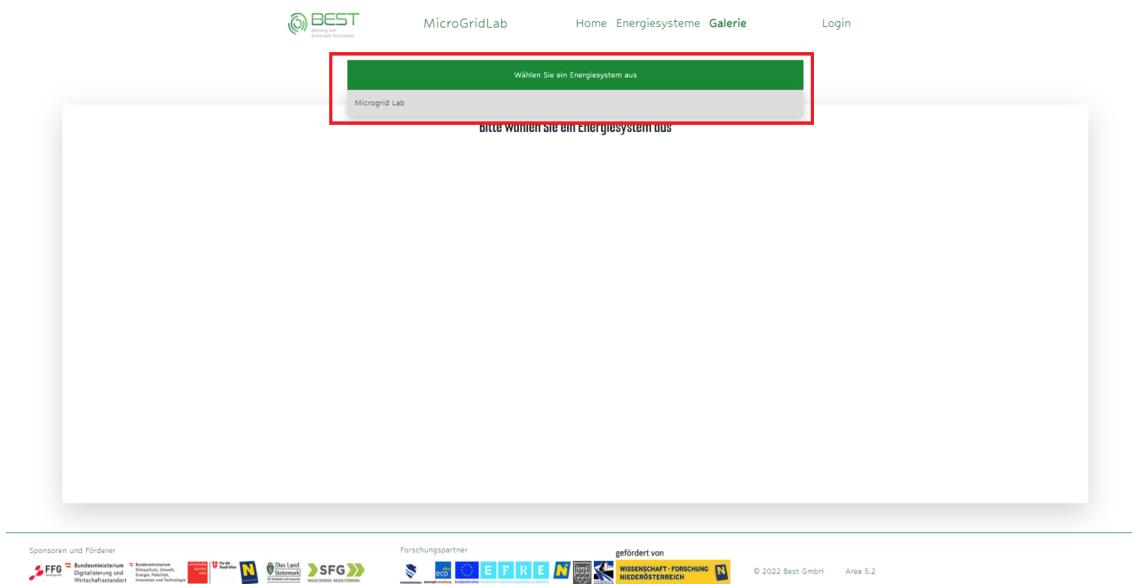


Abbildung 3.10: Interaktion mit dem Drop-Down in der Galerie

### 3.4.7 Schriftarten

Unser Produkt verwendet die Schriftarten Smooch Sans und Hubballi. Diese Schriftarten sind von Google Fonts frei zur Verfügung und zur Verwendung gestellt. Hubballi wurde für jede Art von Fließtext verwendet. Smooch Snans wurde verwendet, um die Überschriften zu formatieren. In der Abbildung 3.11 sind beide Schriften veranschaulicht.

## Intelligente Strom- und Mikronetze

Die 2017 neu gegründete Area „Smart- und Microgrids“ beschäftigt sich mit der konzeptionellen Planung und Steuerung von dezentralen Energieversorgungsprojekten und Microgrids. Dazu werden theoretische, physikalische und wissenschaftliche Zusammenhänge im Bereich der Planung und Steuerung von Smart- und Microgrids erforscht und experimentell entwickelt. Die angewandten Methoden inkludieren Mixed Integer Linear Programming (MILP) oder linearisierte MILP sowie Model-Predictive-Control(MPC)-Methoden. Eine ganzheitliche Betrachtung (Konzeptionierung und Betrieb) von Multi-Energiesystemen (Strom, Wärme, Kälte für Gebäude sowie für kommunale und industrielle Anwendungen) stellt eine komplexe Herausforderung dar, die nur durch den Einsatz der genannten Methoden nachhaltig lösbar ist. Das Hauptziel besteht darin, den Grad der Autonomie der Systeme auf allen hierarchischen Ebenen (einzelne Gebäude, Siedlungen, Energiegemeinschaften, Subnetze, Regionen) und in allen Sektoren des Energiesystems zu erhöhen. Dadurch steigt sich die Gesamteffizienz nachhaltig und die überregionale Infrastruktur wird entlastet. Die Arbeiten dienen als Grundlage für die Entwicklung von Werkzeugen, von welchen das gesamte Energiesystem langfristig profitieren soll.

Abbildung 3.11: Beispiel auf der Website

### 3.4.8 Schriftgrade

Die Schriftgrade variieren bei diesem Produkt nur bei der Anzeige der aktuellen Seite. Die Überschriften Home, Energiesysteme und Galerie werden mithilfe des **<b>** Tags beim Besuchen der gleichnamigen Seite fett gemacht. Dadurch, dass die aktuelle Seite immer fett geschrieben wird, wird die Navigation durch das Produkt erleichtert.



Abbildung 3.12: Header auf der Website

### 3.4.9 Logo

Beim Logo hat sich das Projektteam an den Wunsch des Auftraggebers gehalten und ein Logo, welches von Seitens des Auftraggebers zur Verfügung gestellt wurde, verwendet. Zwei Logos wurden zur Verfügung gestellt und gemeinsam wurde sich auf das Logo in Abbildung 3.13 geeinigt. Das Projektteam rät jedoch dazu, bei Logos SVG Grafiken zu verwenden und nicht wie in diesem Fall eine JPG Grafik. Da dies jedoch für den Auftraggeber keine Relevanz hatte, wurde wie besprochen das Logo aus Abbildung 3.13 verwendet.



Abbildung 3.13: Logo 1



Abbildung 3.14: Logo2

### 3.4.10 Verwendete Icons und deren Bedeutungen

Bei diesem Produkt werden Icons verwendet, um Energiesysteme oder Energietechnologien auf der Map visuell darzustellen. Ein weiteres Einsatzgebiet der Icons sind Formulare oder Übersichten. Auf diese Einsatzgebiete wird in den Kapiteln 3.4.11 und 3.4.12 genauer eingegangen. Wichtig zu unterscheiden sind Icons in schwarz-weiß und Icons in Farbe. Icons in Farbe werden verwendet um Energietechnologien oder Energiesysteme auf der Map dazustellen. Icons in schwarz-weiß werden verwendet um Daten anzuzeigen oder Funktionalitäten bereit zustellen. Eine Liste mit allen Icons und deren Bedeutungen ist im Coperate Design Manuel auf der DVD der Diplomarbeit ersichtlich.

### 3.4.11 Map Icons

Das Produkt verwendet die vom Auftraggeber bereitgestellte Icons, um Energietechnologien oder Energiesystem auf der Karte visuelle darzustellen. Die Icons auf der Map sehen folgendermaßen aus:

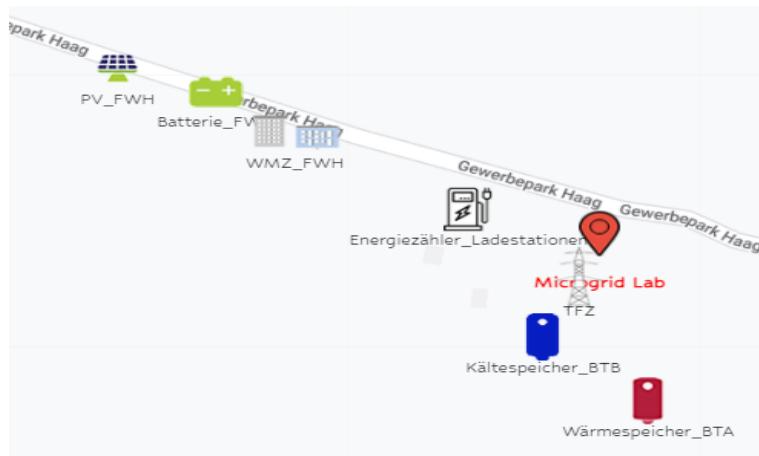


Abbildung 3.15: Icons auf der Map

Um Energiesysteme oder Energietechnologien zu verzeichnen, werden folgende Icons verwendet:



Abbildung 3.16: Icon markiert ein Energiesystem



Abbildung 3.17: Icon markiert eine Energietechnologie

### 3.4.12 Icons in Formularen

Alle weiteren verwendeten Icons wurden von flaticons.com implementiert. Diese Icons werden dazu genutzt, um in Formularen anzuzeigen, welche Art von Wert verlangt wird. Des Weiteren werden diese Icons in Übersichten verwendet, um dem Benutzer anzuzeigen, um welchen Wert es sich handelt. Um diese Icons jedoch verwenden zu dürfen, mussten alle Autoren in der Datenschutzseite verlinkt werden. Hier ein Beispiel, wie diese Anwendungsfälle auf der Website aussehen:



Abbildung 3.18: Icon in Übersichten



Abbildung 3.19: Icon in Formularen

Aus lizenztechnischen Gründen wurde jeder Autor eines Icons auf unserer Seite im Datenschutz verlinkt. Eine Verlinkung sieht folgendermaßen aus:

[Stift Icons erstellt von Icongeek26 - Flaticon](#)

Abbildung 3.20: Autoren Verlinkungen

### 3.4.13 Icons im DataTable

Folgende Icons werden verwendet, um Energiesystem oder Energietechnologien zu bearbeiten, zu löschen oder auf deren Grafana Statistiken zuzugreifen.

Icon	Funktion
	Mit einem Klick auf dieses Icon öffnet sich ein Pop-up, auf welchem die dazugehörigen Grafana Statistiken präsentiert werden
	Mit einem Klick auf dieses Icon wird das ausgewählte Energiesystem gelöscht
	Mit einem Klick auf dieses Icon werden erweiterte Infos zu dem ausgewählten System angezeigt
	Mit einem Klick auf dieses Icon öffnet sich eine Bearbeitungsmöglichkeit, auf welcher man die Eigenschaften des Systems ändern kann

Tabelle 3.5: Icons im DataTabale

Auf der Website werden diese Icons im DataTable hinter einer Energietechnologie oder einem Energiesystem eingeblendet. Dies sieht folgendermaßen aus:

Energietechnologien			Suchen
Bezeichnung	Typ	Ort	
Batterie_FWH	Batteriespeicher	Feuerwehrhaus	
Energiezähler_Ladestationen	E-Ladestation	TFZ	
Kältespeicher_BTB	Kältespeicher	TFZ Bauteil B	
PV_FWH	PV-Anlage	Dach	
TFZ	Stromnetzbezug	TFZ	

bis 5 von 7 Einträgen

Zurück **1** 2 Nächste

Abbildung 3.21: Icons im DataTable

### 3.4.14 Buttons

Buttons werden bei diesem Produkt ausschließlich genutzt, um Formulare abzusenden. Buttons befinden sich immer mittig unter einem Formular und haben die Hintergrundfarbe #1b8836. In Abbildung 3.22 ist ein Beispielbutton erschlich.

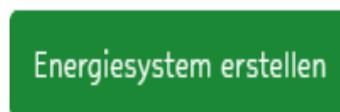


Abbildung 3.22: Beispiel eines Buttons

### 3.4.15 Tabelle mit generellen Informationen über einzelne HTML Elemente

Hier ist wichtig zu definieren, für was welches HTML Element verwendet wurde. <h1>, <h2> und <h3> wurden verwendet, um Überschriften zu definieren. In einem <p> Element befindet sich immer ein Fließtext. Mit dem <a> Element werden Links definiert. Bei diesem Produkt wurden im Header die Navigationsmöglichkeiten mit diesem Element versehen.

HTML Element	Schriftart	Größe	Farbe	Style class oder id
<h1>	Smooch Sans	2em	#1b8836 #212529	DsgvoUberschrift ImpressumUberschrift
<h2>	Smooch Sans	1.5em	#1b8836 #212529	NavUberschrift
<h3>	Smooch Sans	1.17em	#1b8836 #212529	FliessstextUberschrift
<p>	Hubballi	-	#000000	text-primary
<a>	Hubballi	-	#1b8836	dropdown-item

Tabelle 3.6: HTML Elemente

### 3.4.16 Datenformate

Auf dem Produkt befindet sich eine Diashow, die Bilder des Projekts Micro Grid Lab der Best GmbH präsentiert. Alle diese Bilder haben das Dateiformat JPG. Eines der präsentierten Bilder ist im Dateiformat PNG. Alle verwendeten Icons sind als PNG auf dem Produkt eingebunden. Die Fotos der Sponsoren aus dem Footer sind gemischt im PNG und im JPG Format. Das Logo, wie bereits im Kapitel 3.4.9 erwähnt, ist im Format JPG. Hier rät das Projektteam jedoch dazu, eine Vektorgrafik mit dem Dateiformat SVG zu verwenden.

## 3.5 Webanwendung

Im Abschnitt Weboberfläche wird speziell auf das Front-End sowie das Back-End des Produkts eingegangen. Bei dem Punkt Back-End vor allem auf die Funktionen, die für die Benutzer-Interaktionen zuständig sind, und bei dem Punkt Front-End auf das Layout sowie das Design der Weboberfläche.

### 3.5.1 Back-End

In diesem Abschnitt wird genauer auf den Ablauf des Codes im Back-End, der für die Funktionen auf der Weboberfläche zuständig ist, eingegangen. Für jedes Energiesystem oder jede Energietechnologie stehen die Funktionen Erstellen, Bearbeiten, Löschen, Lesen und Suchen zur Verfügung. In den folgenden Abschnitten wird genauer auf diese Funktionen eingegangen.

#### 3.5.1.1 Energiesystem erstellen

Für das Erstellen eines Energiesystems ist ein Mausklick auf der Karte notwendig. Daraufhin öffnet sich ein Pop-up Fenster, um die Kerndaten des Energiesystems einzugeben. Das Pop-up zum Erstellen eines Energiesystems ist in der Abbildung 3.23 ersichtlich. Beim Erstellen eines Energiesystems werden folgende Attribute in der Datenbank erfasst:

Eingabe des Benutzers:

- Bezeichnung
- Katastralgemeinde
- Postleitzahl

Automatisch ausgefüllte Attribute, welche nicht im Pop-up dargestellt werden:

- Längengrad
- Breitengrad
- User\_ID

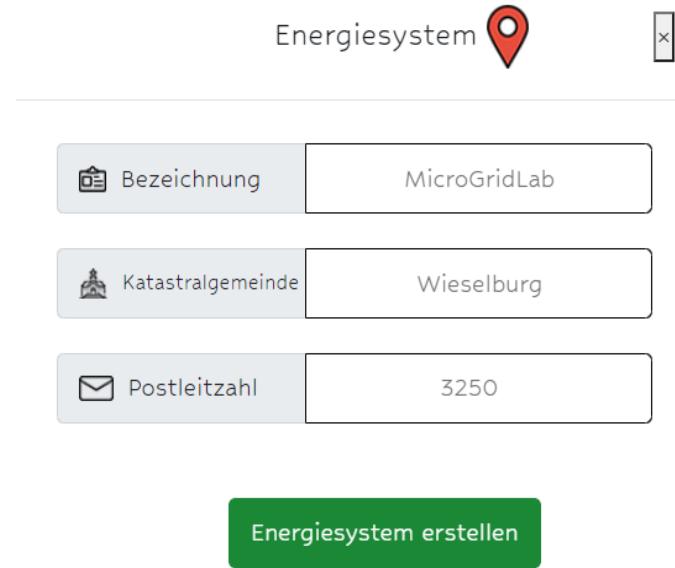


Abbildung 3.23: Energiesystem erstellen Pop-up

Die Attribute Längengrad und Breitengrad werden automatisch mithilfe des Mausklicks auf der Karte mit den entsprechenden Koordinaten beim Erstellen des Energiesystems befüllt. Das Attribut User\_ID wird automatisch mit der ID des gerade angemeldeten Benutzers ausgefüllt, um das Energiesystem einem Benutzer zuteilen zu können.

Nachdem der Benutzer alle Daten eingegeben hat, werden diese Daten an den Controller übermittelt, wo anschließend das Energiesystem erstellt und in der Datenbank erfasst wird. Das Ergebnis ist eine Datenbankabfrage aller vorhandenen Energiesysteme, welche zurück an die Weboberfläche übergeben werden, um die Energiesysteme-Marker auf der Karte zu platzieren sowie den Inhalt der Liste aller vorhandenen Energiesysteme zu aktualisieren. Die folgende Abbildung 3.24 zeigt den Ablauf, um ein Energiesystem zu erstellen.

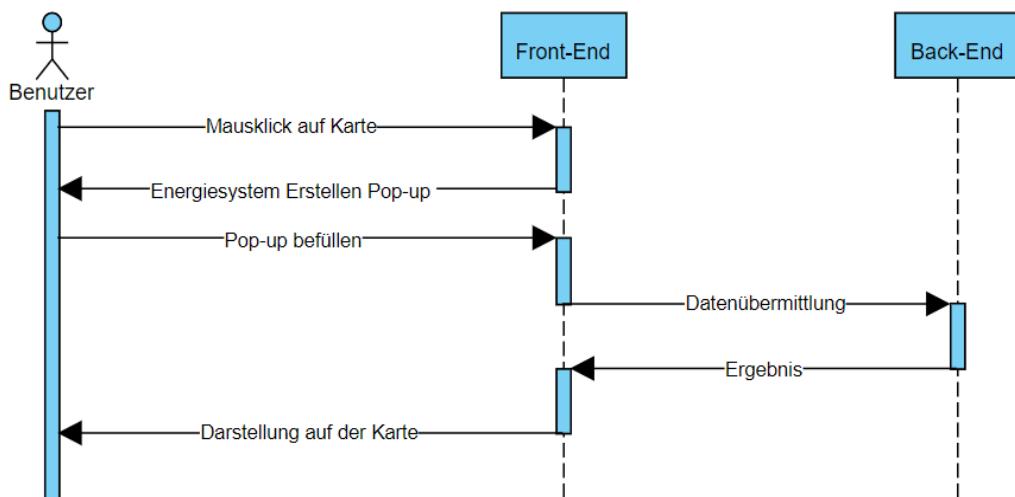


Abbildung 3.24: Energiesystem erstellen

### 3.5.1.2 Energiesystem bearbeiten

Mit dem Stift-Icon beim Energiesystem in der Liste ist es möglich, die Kerndaten eines Energiesystems zu bearbeiten. Die zu bearbeitenden Daten, welche einen weißen Hintergrund aufweisen, lauten auf „Bezeichnung“, „Katastralgemeinde“ und „Postleitzahl“. Die Koordinaten sowie die User\_ID des Energiesystems bleiben konstant und sind nicht änderbar und nicht im Pop-up ersichtlich. Folgende weitere Attribute werden bei jedem Energiesystem automatisch berechnet, angezeigt und sind nicht bearbeitbar, sondern dienen nur als weitere Information über das ausgewählte Energiesystem.

- Anzahl-Erzeugungstechnologien
- Anzahl-Verbraucher
- Anzahl-Speicher
- Ges-Nennleistung [kW]
- Ges-Energie [kW/h]
- Ges-Verbraucher-Leistung [kW]
- Ges-Verbraucher-Energie [kW/h]
- Ges-Erzeuger-Leistung [kW]
- Ges-Erzeuger-Energie [kW/h]
- Ges-Speicher-Kapazität [kW/h]
- Aktueller Netzbezug [kW]

Für genauere Informationen zu den einzelnen Attributen siehe Kapitel 3.4.10. Mit dem Button „Mehr Details zu diesem Energiesystem“ ist es möglich, die weiteren Daten des Energiesystems ein- und auszuklappen. Das Pop-up zum Bearbeiten eines Energiesystems ist in der Abbildung 3.25 ersichtlich.

Nachdem der Benutzer die Attribute des Energiesystems bearbeitet hat, werden diese mithilfe des Controllers in der Datenbank aktualisiert. Anschließend werden die neuen Daten der Weboberfläche übergeben, um auf der Karte sowie in der Liste den aktuellen Stand der Energiesysteme darstellen zu können. Der Ablauf der dieser Funktion ist gleichzusetzen mit dem Ablauf der Energiesystem Erstellen Funktion, mit dem Unterschied, dass sich das Bearbeiten Pop-up öffnet um die Daten des Energiesystems zu bearbeiten. Dieser Ablauf ist in der Abbildung 3.24 ersichtlich.



Abbildung 3.25: Energiesystem bearbeiten Pop-up

### 3.5.1.3 Energiesystem löschen

Um ein Energiesystem zu löschen, benötigt man das Müllmeimer-Icon, welches sich neben jedem Energiesystem in der Liste befindet, sofern der angemeldete Benutzer die erforderliche Berechtigung dazu hat. Nachdem dieses Icon betätigt wurde, wird die Information, dass ein Energiesystem gelöscht wurde, an den Controller übermittelt. Dieser löscht anschließend mithilfe der ID des ausgewählten Energiesystems dieses aus der Datenbank und übergibt alle anderen Energiesysteme zurück auf die Weboberfläche, um die vorhandenen Energiesysteme auf der Karte zu platzieren sowie in der Liste anzuzeigen.

### 3.5.1.4 Energietechnologie erstellen

Für das Erstellen einer Energietechnologie ist das Auswählen eines Energiesystems mit einem Doppelklick notwendig. Anschließend verwandelt sich der Cursor in das Energietechnologie-Icon, welches darauf hindeutet, dass jetzt das Hinzufügen einer Energietechnologie mittels eines Klicks auf der gewünschten Position auf der Karte möglich ist. Nach dem Klick auf der Karte öffnet sich ein Pop-up-Fenster, um die Kerndaten der Energietechnologie einzugeben. Dieses Pop-up ist in der Abbildung 3.26 ersichtlich.

Eingabe des Benutzers:

- Bezeichnung
- Typ
- Ort
- Bild einfügen

- Beschreibung

Automatisch ausgefüllt und nicht im Pop-up dargestellt:

- Längengrad
- Breitengrad
- Ensys\_ID
- User\_ID

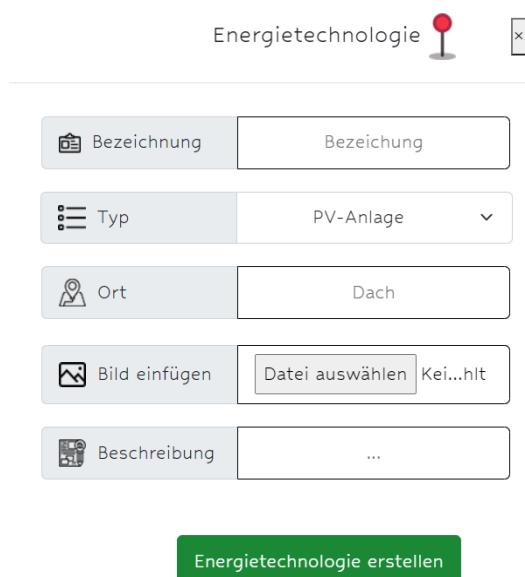


Abbildung 3.26: Energietechnologie erstellen Pop-up

Nachdem der Benutzer die Daten im Pop-up ausgefüllt hat, werden diese Informationen über den Controller in der Datenbank erfasst. Anschließend werden die Energietechnologie-Daten an die Weboberfläche übermittelt, um die Energietechnologien auf der Karte darzustellen sowie in der Liste anzuzeigen. Der Ablauf, um eine Energietechnologie zu erstellen, wird in der Abbildung 3.27 dargestellt:

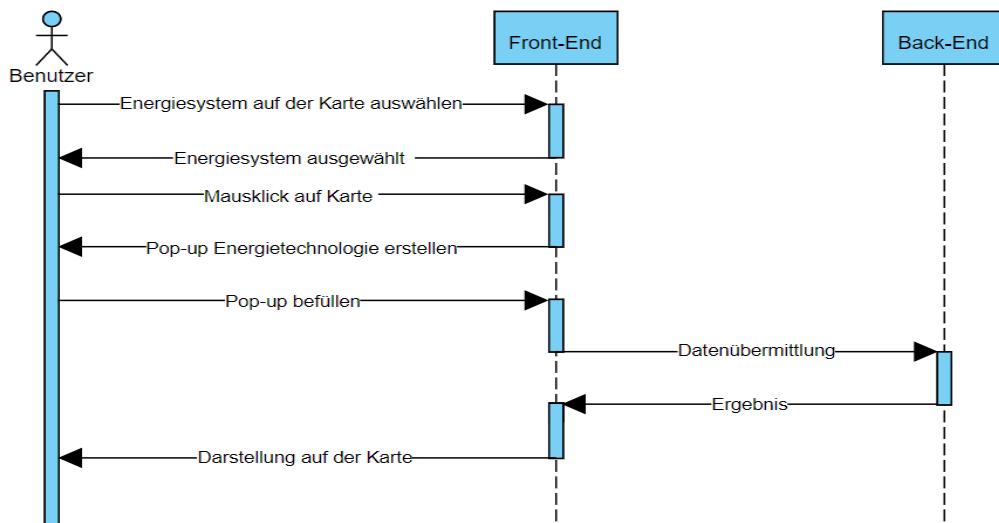


Abbildung 3.27: Energietechnologie erstellen

### 3.5.1.5 Energietechnologie bearbeiten

Um eine Energietechnologie zu bearbeiten, muss zuerst ein Energiesystem mit einem Doppelklick ausgewählt werden. Anschließend befinden sich rechts in der Liste alle dazugehörigen Energietechnologien, welche man mit dem Stift-Icon bearbeiten kann. Nach dem Betätigen des Stift-Icons öffnet sich ein Pop-up-Fenster, um die Attribute zu bearbeiten. Dieses Pop-up-Fenster ist in der Abbildung 3.28 ersichtlich.

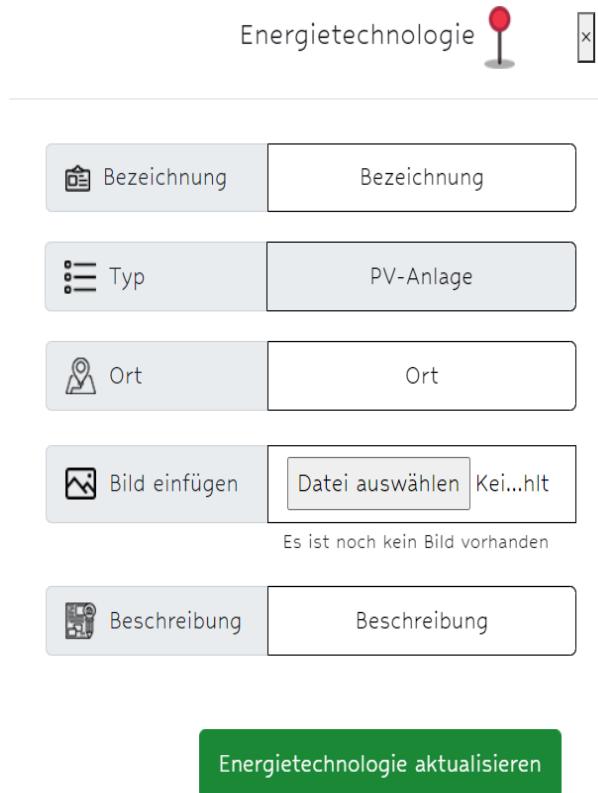


Abbildung 3.28: Energietechnologie bearbeiten

Dabei sind die Attribute „Bezeichnung“, „Ort“, „Bild“ und „Beschreibung“ bearbeitbar. Um dem Benutzer dies deutlich zu machen, weisen diese Attribute einen weißen Hintergrund auf. Die Attribute „Typ“, „Längengrad“, „Breitengrad“, „Ensys\_ID“ sowie „User\_ID“ sind nicht bearbeitbar. Nachdem der Benutzer die Daten bearbeitet hat, werden diese über den Controller in die Datenbank geschrieben. Anschließend werden die Energietechnologie-Daten an die Weboberfläche übergeben, um die Marker auf der Karte zu platzieren sowie den Inhalt der Liste zu aktualisieren. Der Ablauf der dieser Funktion ist gleichzusetzen mit dem Ablauf der Energietechnologien Erstellen Funktion, mit dem Unterschied, dass sich das Bearbeiten Pop-up öffnet um die Daten der Energietechnologie zu bearbeiten. Dieser Ablauf ist in der Abbildung 3.27 ersichtlich.

### 3.5.1.6 Energietechnologie löschen

Um eine Energietechnologie zu löschen, benötigt man das Mülleimer-Icon, welches sich in der Liste neben den Energietechnologien befindet, sofern ein Energiesystem von dem Benutzer ausgewählt wurde. Sobald der Benutzer dieses Icon betätigt, wird mithilfe der ID der ausgewählten Energietechnologie dem Controller mitgeteilt, dass diese Energietechnologie gelöscht werden soll. Daraufhin löscht der Controller diese Energietechnologie aus der Datenbank und übergibt anschließend alle anderen vorhandenen Energietechnologie-Daten an die Weboberfläche, um dort die Marker zu platzieren sowie den Inhalt der Liste zu aktualisieren.

### 3.5.1.7 Benutzerverwaltung

Je nach Benutzerrolle des aktuell angemeldeten Benutzers stehen dem Benutzer unterschiedliche Funktionen auf der Weboberfläche zur Verfügung. Folgende drei Rollen stehen zur Verfügung:

- Administrator
- Mitarbeiter
- Öffentlicher Benutzer

Wie eine generelle Überprüfung des Benutzers in Laravel umgesetzt werden kann, ist unter der Quelle x.y ersichtlich.

Folgende Überprüfung wurde selbst vom Projektteam entwickelt und ist somit nicht unter der genannten Quelle auffindbar.

Mit dieser Überprüfung wird kontrolliert, ob der gerade angemeldete Benutzer das Energiesystem oder die Energietechnologie selbst erstellt hat oder ob er die Rolle des Administrators aufweisen kann. Falls der Benutzer das Energiesystem oder die Energietechnologie selbst erstellt hat, ist der erste Teil der Überprüfung erfüllt und der Benutzer hat die Verwaltungsfunktionen zur Verfügung. Falls der Benutzer das Energiesystem oder die Energietechnologie nicht selbst erstellt hat, wird die zweite Überprüfung durchgeführt, welche überprüft, ob der Benutzer die Rolle des Administrators aufweisen kann. Falls er diese Rolle besitzt, hat er die Verwaltungsfunktionen zur Verfügung, andernfalls nicht.

Quellcode 3.1: Überprüfung der Benutzerrolle

```
1 @if ($userID->id == $d->users_idusers || $userID->role == 'Admin')
```

### 3.5.1.8 Adresssuche

Mithilfe des Adresssuchfeldes ist es möglich, einen Standort einzugeben, um anschließend auf der Karte zu diesem zu gelangen. Dadurch ist das Auffinden von bestimmten Orten auf der Karte problemlos möglich. Dabei wird die eingegebene Adresse in geografische Koordinaten umgewandelt, zu welchen man anschließend navigiert wird. In der folgenden Abbildung 3.29 ist das Adresssuchfeld ersichtlich.



Abbildung 3.29: Adresssuchfeld

Folgende Schreibweisen sind in der Suche möglich:

- Stadt
- Land
- Straße
- Postleitzahl

#### Automatische Vervollständigung

Um die Suche nach der richtigen Adresse zu vereinfachen, wird die Eingabe des Benutzers mit einer Autocomplete-Funktion unterstützt. Diese Funktion bietet mögliche Ziel-Adressen anhand der bisher eingegebenen Daten an, welche vom Benutzer ausgewählt werden können. Dafür wurde eine von Google bereitgestellte API verwendet. Quelle Xy Ein Beispiel dieser Funktion ist in der Abbildung 3.30 dargestellt.

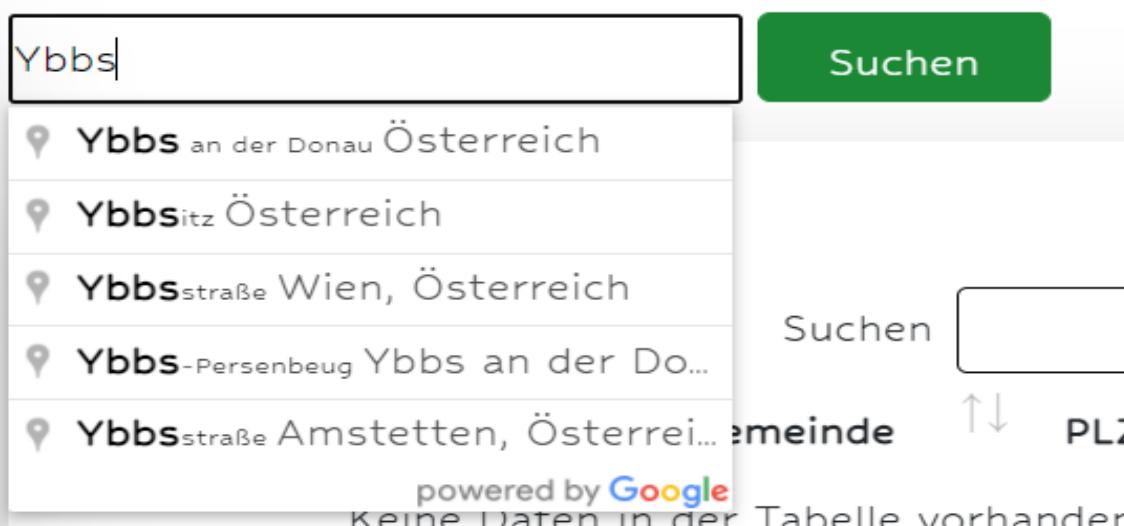


Abbildung 3.30: Automatische Vervollständigung

### Adresssuche durchführen

Die Adresssuche kann auf zwei verschiedene Arten ausgelöst werden. Die erste Variante ist das Benutzen des dazugehörigen Buttons mit der Beschriftung „Suchen“. Die zweite Variante ist das Betätigen der Enter-Taste auf der Tastatur. Beide Varianten führen zu dem gleichen Ergebnis, und zwar, dass die Adresssuche durchgeführt wird. Sobald die Adresssuche durchgeführt wird, werden mit Hilfe von Google-Cloud API-Aufrufe für die eingegebene Adresse die dazugehörigen Koordinaten berechnet. Sobald diese berechnet wurden, werden diese an die Karte übergeben, damit der Mittelpunkt der Karte auf diese Koordinaten gesetzt wird. Anschließend gelangt der Benutzer zu seiner eingegebenen Adresse auf der Karte und kann seine Interaktionen fortsetzen. Die Abbildung 3.31 beschreibt die Durchführung der Adresssuche.

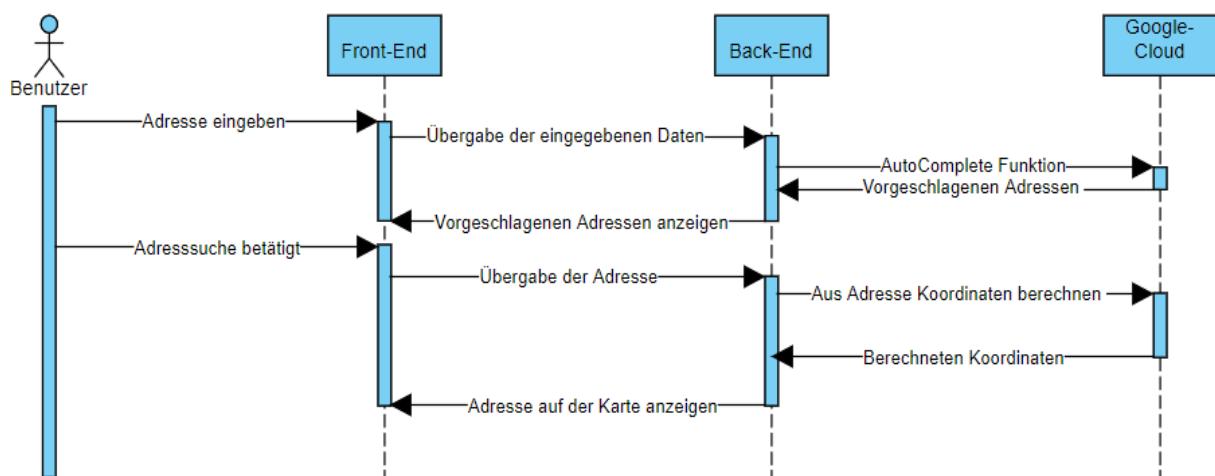


Abbildung 3.31: Adresssuche

### 3.5.2 Front-End

Das Projektteam hat sich bei der Front-End Gestaltung an die Vorgabe des Auftraggebers gehalten und sich an der bereits vorhanden Website der Best GmbH, welche in 3.32 ersichtlich ist, orientiert. Das Produkt wurde dann mithilfe der Laravel Layout<sup>3</sup>Funktion in Header Footer und Content unterteilt. Generell ist das Produkt gegliedert in folgende Unterseiten:

- Home
- Energiesysteme
- Galerie
- Impressum
- Datenschutz
- Registrierungsseite

Auf die oben aufgezählten Unterseiten wird in den nachfolgenden Kapiteln genauer eingegangen.



Abbildung 3.32: Website Best GmbH

<sup>3</sup>genauer in Abschnitt 2.7 erklärt

### 3.5.2.1 Home

Die Homeseite ist die erste Seite, die ein Benutzer zu Gesicht bekommt, wenn er das Produkt aufruft. Die Seite besteht aus einer Diashow, die Bilder des Micro Grid Lab Projekts der Best GmbH zeigt. Unter dieser befindet sich ein Textbereich, auf welchem das Projekt "Micro Grid Lab" kurz vorgestellt wird. Hier ein Bild der besagten Seite:

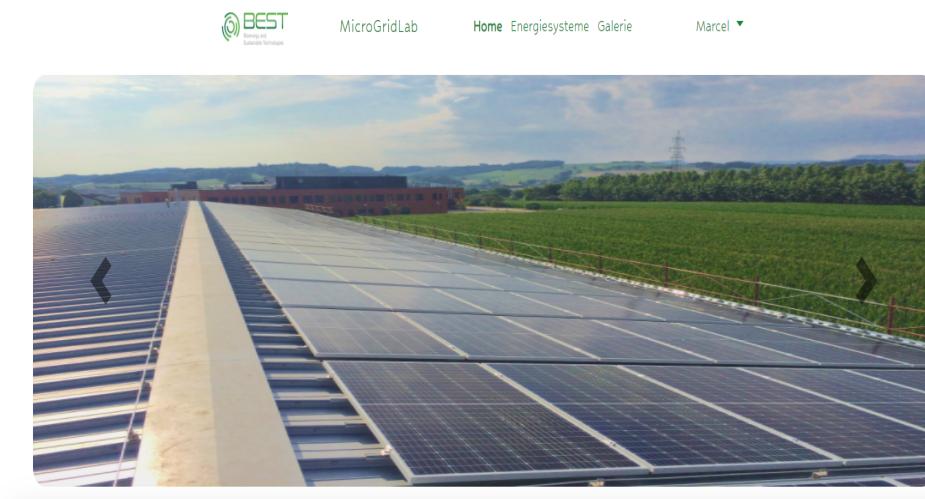


Abbildung 3.33: Oberer Teil der Home Seite

**Intelligente Strom- und Mikronetze**

Die 2017 neu gegründete Area „Smart- und Microgrids“ beschäftigt sich mit der konzeptionellen Planung und Steuerung von dezentralen Energieversorgungsprojekten und Microgrids. Dazu werden theoretische, physikalische und wissenschaftliche Zusammenhänge im Bereich der Planung und Steuerung von Smart- und Microgrids erforscht und experimentell entwickelt. Die anwendenden Methoden inkludieren Mixed Integer Linear Programming (MILP) oder linearisierte MILP sowie Model-Predictive-Control(MPC)-Methoden. Eine ganzheitliche Betrachtung (Konzeptplanung und Betrieb) von Multi-Energiesystemen (Strom, Wärme, Kläre für Gebäude sowie für kommunale und industrielle Anwendungen) stellt eine komplexe Herausforderung dar, die nur durch den Einsatz der genannten Methoden nachhaltig lösbar ist. Das Hauptziel besteht darin, den Grad der Autonomie auf allen hierarchischen Ebenen (einzelne Gebäude, Siedlungen, Energiegemeinschaften, Subnetze, Regionen) und in allen Sektoren des Energiesystems zu erhöhen. Dadurch steigert sich die Gesamteffizienz nachhaltig und die überregionale Infrastruktur wird entlastet. Die Arbeiten dienen als Grundlage für die Entwicklung von Werkzeugen, von welchen das gesamte Energiesystem langfristig profitieren soll.

**Entwicklungsfelder und Anwendungsbiete**

» Entwicklung von Methoden und Tools für die Energiesystemplanung und die sektorenübergreifende Energieinfrastruktur sowie die Identifizierung von Schwachstellen » Schaffung einer prädiktiven Steuerung via MPC, die die Wechselwirkungen zwischen den Sektoren berücksichtigt und durch den Einsatz von Speichern und Lastverschiebungen mehr Flexibilität und Stabilität im Energiesystem bietet

**Forschungsprojekt Microgrid Lab**

In dem Forschungsprojekt Microgrid Lab wird ein Microgrid für kommunale Energiekonzepte in einem realen Umfeld geplant, errichtet, evaluiert und auf wissenschaftlicher Ebene weiterentwickelt. Ziel ist die Etablierung des Forschungslabors für verschiedene Wirtschaftszweige, um Planungs-, Steuerungs-, Integrations- und Kommunikationskonzepte zu entwickeln und für den Markt zu testen. Das betrifft auch sektorenübergreifende Energienetze (Wärme, Strom, Gas, Wasserstoff). Projektinhalt sind die wissenschaftliche Planung, die Inbetriebnahme, ein standardisiertes Monitoring der Verbraucher/Erzeuger (u.a. Biomasse, PV, Batterie, E-Ladestationen, Absorptionskälte), die Entwicklung von Testzyklen, die Weiterentwicklung der Optimierungsalgorithmen und ein Wissenstransfer zu verschiedenen Stakeholdergruppen. Mithilfe entwickelter mathematischer Methoden wurde bereits ein optimales Energieplanungskonzept entwickelt, welches auf Gemeinden übertragen werden kann. Die innovativen Planungs- und Steuerungskonzepte ermöglichen CO<sub>2</sub>- und Kostenersparnisse von bis zu 90% bzw. 40%.

Sponsoren und Förderer

FFG – Bundesministerium für Digitalisierung und Wirtschaftstechnik | BMK – Bundesministerium für Arbeit, Familie, Energie, Mobilität, Klimaschutz und Sport | Das Land Niederösterreich | SFG – Österreichische Netzwerke

Forschungspartner

WU – Wirtschaftsuniversität Wien | ecu – energiecluster unternehmen | EFRE – Europäische Fonds für Regionale Entwicklung | NAWI – NAWI Austria | WISSENSCHAFT – FORSCHUNG – INDUSTRIE NIEDERÖSTERREICH

gefördert von

WISSENSCHAFT – FORSCHUNG – INDUSTRIE NIEDERÖSTERREICH

© 2022 Best GmbH Area 5.2

Impressum Datenschutz

Abbildung 3.34: Unterer Teil der Homeseite

### 3.5.2.2 Energiesysteme

Die Energiesystemeseite ist die Seite, wo die Hauptfunktion des Produkts verwendet werden kann. Auf dieser wird die Verwaltung der Energiesysteme und Energietechnologien mithilfe einer Map visuell dargestellt. Diese Seite besteht aus einer Google Maps Karte, welche sich auf der linken Seite der Website befindet. Ebenso einem DataTable auf der rechten Hälfte der Website. Hier ein Bild der eben beschriebenen Seite:

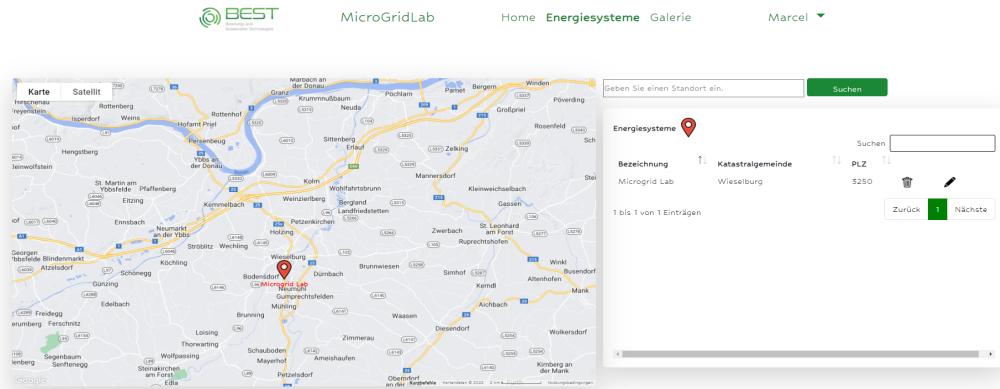


Abbildung 3.35: Die Energiesystem Seite

### 3.5.2.3 Galerie

Das Produkt bietet auch die Möglichkeit, zu jeder Energietechnologie ein dementsprechendes Bild zu speichern. Dieses Bild wird dann in der Bildergalerie angezeigt. Durch einen Dropdown, welches sich oben in der Mitte der Seite befindet, kann ein Energiesystem ausgewählt werden. Von diesem ausgewählten Energiesystem werden dann alle dazugehörigen Energietechnologien mit dem entsprechendem Bild angezeigt. Die Galerie sieht folgendermaßen aus:



Abbildung 3.36: Die Galerie

### 3.5.2.4 Impressum

Auf der Impressumsseite, welche durch den Footer erreichbar ist, ist das Impressum der Website einsehbar. Die Seite hat eine Überschrift “Impressum” und darunter befindet sich ein Textbereich mit dem dazugehörigen Impressum. Am Ende des Impressums befindet sich noch ein Link, welcher auf die Homepage der Best GmbH weiterleitet. Hier ist ein Foto der Impressumsseite:



Abbildung 3.37: Das Impressum

### 3.5.2.5 Datenschutz

Wie auch die Impressumsseite ist die Datenschutzseite durch den Footer erreichbar. Diese ist gleich aufgebaut wie die Impressumsseite. Auch hier gibt es wieder mittig eine Überschrift “Datenschutz”, unter welcher wieder ein Textbereich vorhanden ist. In diesem Textbereich ist die Datenschutzgrundverordnung des Produkts einsehbar. Hier ein Foto der Datenschutzseite:



Abbildung 3.38: Die Datenschutzseite

### 3.5.2.6 Registrierungsseite

Die Registrierungsseite kann nur von einem Administrator erreicht werden. Auf dieser Seite ist es möglich, neue Benutzer zu registrieren oder bereits vorhandene Benutzer zu löschen. Genauere Informationen zu dieser Seite sind im Kapitel 3.5.3 nachlesbar. Hier ein Foto der Registrierungsseite:

The screenshot shows a web interface for managing users. At the top, there is a navigation bar with the BEST logo, the text "MicroGridLab", and links for "Home", "Energiesysteme", "Galerie", and a dropdown menu for "Marcel". Below the navigation is a modal window titled "Neuen Benutzer anlegen" (Create new user). It contains fields for "Name", "E-Mail Adresse", "Passwort", and "Rolle". The "Rolle" field has two options: "Mitarbeiter" (selected) and "Admin". A button at the bottom of the modal says "Neuen Benutzer erstellen" (Create new user). Below the modal is a table listing existing users. The columns are "Name", "Email", "Role", and a delete icon. The users listed are: David (david.poehacker@sz-ybbs.ac.at, Admin), Tobias (tobias.kronsteiner@sz-ybbs.ac.at, Admin), Jürgen Mitterlehner (juergen.mitterlehner@best-research.eu, Admin), Michael Zeillinger (michael.zeillinger@best-research.eu, Admin), and Armin Cosic (armin.cosic@best-research.eu, Admin). At the bottom of the table, it says "1 bis 5 von 15 Einträgen" and has navigation buttons for "Zurück", "1", "2", "3", and "Nächste".

Abbildung 3.39: Die Registrierungsseite

### 3.5.3 Login

Möchte sich ein Benutzer beim Produkt anmelden, kann er dies beim Loginfeld, welches in Abbildung 3.40 ersichtlich ist. Dort meldet er sich mit einer E-Mail-Adresse und einem Passwort an, welche vorher auf der Registrierungsseite angelegt wurde. Nach Eingabe der Daten, wird überprüft, ob die Daten in der Datenbank vorhanden sind und das Passwort richtig eingegeben wurde. Das Loginfeld sieht folgendermaßen aus:

The screenshot shows a login form titled "Login". It has fields for "E-Mail Address" and "Password", both represented by large empty input boxes. Below the password field is a "Remember Me" checkbox. At the bottom of the form is a blue "Login" button. To the right of the "Login" button is a link "Forgot Your Password?".

Abbildung 3.40: Das Login Formular

### 3.5.4 Registrierung

Um neue Benutzer registrieren zu können, gibt es eine eigene Registrierungsseite. Diese Seite, ersichtlich in Abbildung xy Registrierungsseite, besteht aus einem Formular und einem DataTable und kann nur von Benutzern mit der Rolle Admin aufgerufen werden. Auf dem DataTable werden alle vorhandenen Benutzer angezeigt. Des Weiteren bietet er die Möglichkeit, vorhandene Benutzer zu löschen. Das Formular ermöglicht es, Benutzer mit folgenden Werten anzulegen:

- Name
- E-Mail-Adresse
- Passwort
- Rolle

The screenshot shows a user interface for creating a new user. At the top, it says "Neuen Benutzer anlegen". Below that are four input fields: "Name", "E-Mail Adresse", "Passwort", and "Rolle". Under "Rolle", there are two radio buttons: "Mitarbeiter" (selected) and "Admin". At the bottom is a green button labeled "Neuen Benutzer erstellen".

Abbildung 3.41: Das Registrierungsformular

### 3.5.5 Kartendienst Funktionalitäten

Neben den bekannten Kartendienst-Funktionen wie das Erstellen eines Energiesystems oder einer Energietechnologie bietet die Karte weitere Funktionen. Diese weiteren Funktionen sind das Auswählen und Abwählen eines Energiesystems.

#### 3.5.5.1 Auswählen eines Energiesystems

Das Auswählen eines Energiesystems ist notwendig, um die dazugehörigen Energietechnologien auf der Karte sowie in der Liste zu sehen. Ebenso ist es notwendig, um neue Energietechnologien erstellen zu können. Mit einem einfachen Klick auf das Energiesystem-Icon wird an dieses Energiesystem lediglich herangezoomt, jedoch noch nicht ausgewählt. Um ein Energiesystem auszuwählen, ist ein Doppelklick darauf notwendig. Dieser Doppelklick bewirkt, dass in der Liste sowie auf der Karte die dazugehörigen Energietechnologien angezeigt werden. Anschließend ist es für den Benutzer möglich, die Energietechnologien zu verwalten. Der Ablauf für das Auswählen eines Energiesystems wird in der Abbildung 3.42 dargestellt:

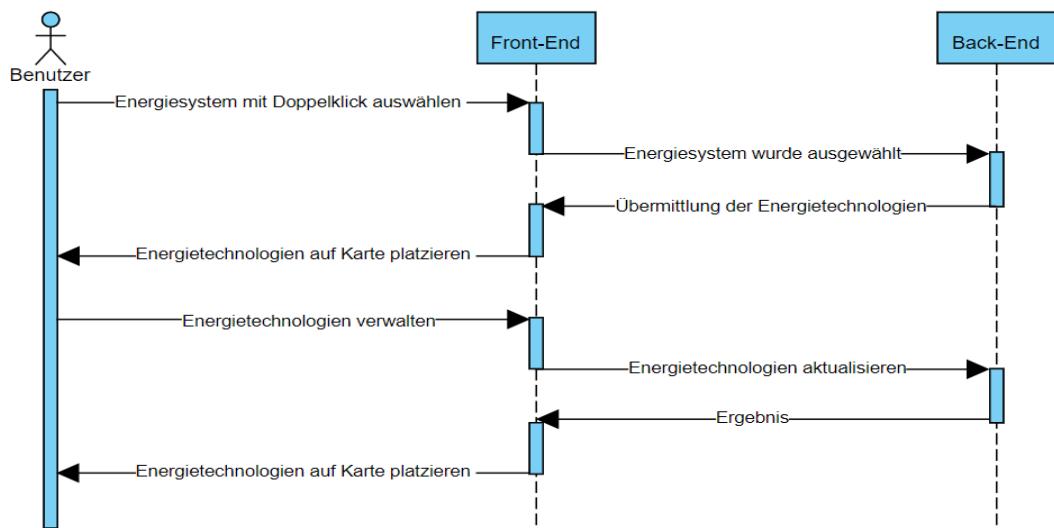


Abbildung 3.42: Energiesystem auswählen

### 3.5.5.2 Abwählen eines Energiesystems

Nachdem ein Energiesystem ausgewählt wurde, ist mit einem einfachen Linksklick auf das Energiesystem-Icon möglich, das ausgewählte Energiesystem wieder abzuwählen. Nach Abwählen des Energiesystems werden die dazugehörigen Technologien wieder von der Karte sowie aus der Tabelle entfernt. Stattdessen werden in der Tabelle wieder alle vorhandenen Energiesysteme angezeigt.

### 3.5.6 Anzeige von Energiesystemen und Energietechnologien auf der Karte

In diesem Abschnitt werden die Funktionen zum Darstellen und Entfernen der Marker auf der Karte erläutert.

#### 3.5.6.1 Energiesysteme Marker auf der Karte platzieren

Immer wenn die Map neu geladen wird, werden alle vorhandenen Energiesysteme in Form von Icons auf der Karte platziert. Dafür werden die Daten aller Energiesysteme aus der Datenbank gelesen, um anschließend die Marker auf der Karte platzieren zu können. Die dafür notwendigen Datenbank-Attribute, die ausgelesen werden müssen, sind „Bezeichnung“, „Breitengrad“ sowie „Längengrad“. Anschließend werden mit diesen Informationen die Marker erstellt und auf der Karte dargestellt. Die Abbildung 3.43 beinhaltet den Ablauf für das Platzieren der Energiesysteme-Marker auf der Karte.

#### 3.5.6.2 Energietechnologien Marker auf der Karte platzieren

Sobald ein Energiesystem ausgewählt wurde, werden die dazugehörigen Energietechnologien angezeigt. Für das Anzeigen eines Energietechnologie-Markers werden die Attribute „Bezeichnung“, „Typ“, „Breitengrad“ sowie „Längengrad“ aus der Datenbank ausgelesen. Anschließend werden die Energietechnologie-Marker erstellt und auf der Karte präsentiert. Die Abbildung 3.43 stellt den Ablauf für das Platzieren der Energietechnologien-Marker auf der Karte dar.

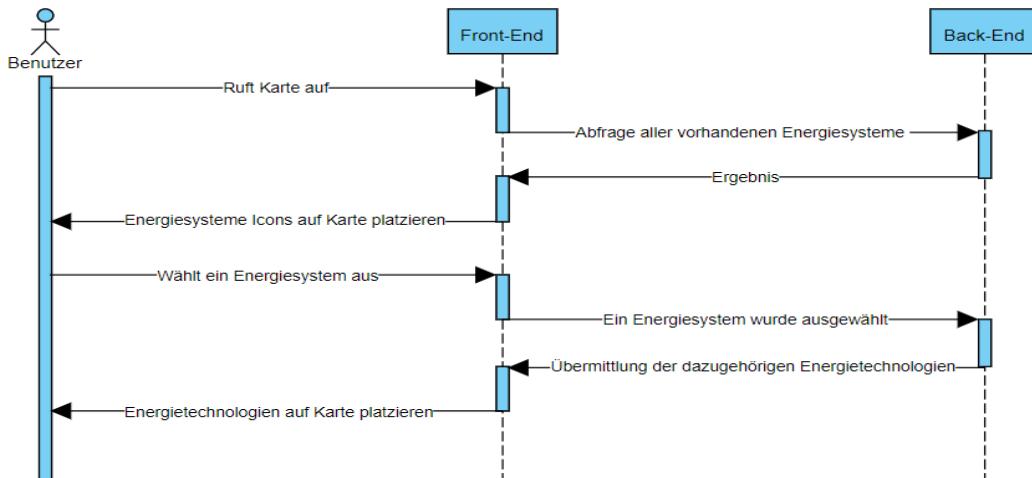


Abbildung 3.43: Energietechnologie Marker platzieren

### 3.5.7 Layoutvorlage der Website

Das Produkt verwendet die Layout Funktion vom Framework Laravel. Genauer Informationen wie dieses Layout funktioniert können im Abschnitt 2.7 nachgelesen werden. Jede Unterseite des Produkts ist in folgende Sections gegliedert:

- Header
- Content
- Footer

In der Section Header ist die Navigationsleiste vordefiniert. Die Section Content ist leer, da hier der individuelle Inhalt der einzelnen Unterseiten hineinkommt. In der Section Footer wurde ein selbst erstellter Footer eingefügt.

## 3.6 DataTable

Um die angezeigten Daten in der Tabelle zu sortieren, zu filtern oder deren Anzahl pro Seite zu begrenzen, wurde das Plug-in DataTable verwendet. DataTable ist ein Plug-In der JavaScript-Bibliothek jQuery. Dieses Tool bietet viele Funktionen wie die Suchfunktion, die Sortierfunktion und die Seitennummerierung.

Das Projektteam hat sich für eine Einbindung mittels CDN<sup>4</sup> entschieden. Für einen Standard-DataTable mittels CDN sind folgende Schritte notwendig:

JavaScript und CSS-Files einbinden

Je nach Verwendungsart des DataTables müssen unterschiedliche JavaScripts und CSS-Files eingebunden werden. Unter der Quelle x.y kann herausgefunden werden, welche Files eingebunden werden müssen.

Nachdem die Files eingebunden sind, muss folgende Funktion hinzugefügt werden.

Quellcode 3.2: DataTable Funktion

```
100 $(document).ready( function () {
101   $('#TableID').DataTable();
102 } );
```

Anschließend sind alle DataTable-Funktionen an diesen Table gegeben. Links oben befindet sich ein Input-Feld, um die Anzahl der Datensätze pro Seite festzulegen. Die Suchfunktion, um nach einem bestimmtem Datensatz zu suchen, befindet sich rechts oben. Für jedes Attribut ist eine Sortierfunktion gegeben, welche an den kleinen Pfeilen neben dem Attributnamen erkennbar ist. Links unten sieht man, wieviele Datensätze gerade auf dieser Seite angezeigt werden und wieviele es insgesamt gibt. Rechts unten ist es möglich, mit den Buttons zwischen den Seiten hin und her zu wechseln, falls mehrere Seiten vorhanden sind. Die Sprache, die für den Standard-DataTable verwendet wird, ist Englisch. In der Abbildung 3.44 ist ein DataTable mit den Standardeinstellungen ersichtlich.

---

<sup>4</sup>Content Delivery Network Quelle x

Show <input type="text"/> entries		Search: <input type="text"/>	
Bezeichnung	Katastralgemeinde	PLZ	
HTL Ybbs	Ybbs	3370	
MicroGridLab	Wieselburg	3250	
Showing 1 to 2 of 2 entries			
<a href="#">Previous</a>		<a href="#">1</a>	<a href="#">Next</a>

Abbildung 3.44: Standard DataTable

### 3.6.1 Individueller DataTable

Aufbauend auf den Standard-DataTable wurde der individuelle DataTable basierend auf den Anforderungen des Auftraggebers erstellt. Dabei ist die wichtigste Anforderung, dass pro Seite maximal 5 Datensätze dargestellt werden. Zuerst wurde die Sortierfunktion bei den Spalten drei, vier und fünf deaktiviert, da sich an diesen Stellen die Icons befinden und somit eine Sortierfunktion keinen Sinn macht. Die Auswahl für die Anzahl der Datensätze pro Seite wurde deaktiviert, da diese konstant auf den Wert fünf festgelegt wurde. Zum Schluss wurde die Sprache des Tables auf Deutsch geändert. In der Abbildung 3.45 ist der individuelle DataTable abgebildet.

Suchen <input type="text"/>			
Bezeichnung	Katastralgemeinde	PLZ	
HTL Ybbs	Ybbs	3370	
MicroGridLab	Wieselburg	3250	
1 bis 2 von 2 Einträgen			
<a href="#">Zurück</a>		<a href="#">1</a>	<a href="#">Nächste</a>

Abbildung 3.45: Individueller DataTable

### 3.6.2 Sortierfunktion

Mit der Sortierfunktion ist es möglich, nach jedem einzelnen Attribut in der Liste zu sortieren. Die Sortierfunktion ist nur bei jenen Attributen aktiviert, wo es auch Sinn macht. Somit ist die Sortierfunktion bei den Icons nicht gegeben. Beim Laden des Tables ist der Inhalt automatisch alphabetisch nach dem ersten Attribut sortiert. Nach dem Betätigen der Sortierfunktion wird anschließend alphabetisch rückwärts sortiert.

### 3.6.3 Suchfunktion

Die Suchfunktion ermöglicht eine sofortige Textsuche des Inhaltes in der Liste. Dabei ist es möglich, mit Buchstaben oder mit Zahlen zu suchen. Jeder Datensatz, der den eingegebenen Buchstaben oder die eingegebene Zahl beinhaltet, wird angezeigt. Der Rest wird ausgeblendet und ist erst nach dem Beenden der Suche wieder sichtbar.

### 3.6.4 Seitenanzahl

Wenn auf einer Seite maximal fünf Datensätze angezeigt werden, entstehen bei einer großen Anzahl von Energiesystemen sowie Energietechnologien entsprechend viele Seiten. Diese Seiten sind mithilfe der Buttons rechts unten navigierbar. Links unten steht die Information darüber, wieviele Seiten es insgesamt gibt, und wieviele Einträge von allen vorhandenen gerade auf dieser Seite angezeigt werden.

### 3.6.5 Icons

Die Verwaltung der einzelnen Energiesysteme und der Energietechnologien ist mit den dazugehörigen Icons möglich. Welche Icons zu jedem Energiesystem oder jeder Energietechnologie zur Verfügung stehen, hängt von den Rechten des angemeldeten Benutzers ab. Es gibt zwei verschiedene Kombinationen von verfügbaren Icons. Entweder man hat nur das Icon „Auge“ zur Verfügung, welches bedeutet, dass man dieses Energiesystem oder diese Energietechnologie entweder nicht erstellt hat, nicht mit einem Administrator-Benutzer angemeldet ist oder gerade nicht auf der Weboberfläche angemeldet ist. Mit diesem Icon hat man die Möglichkeit, die Kerndaten eines Systems zu sehen, welche aber nicht bearbeitet werden können. Die zweite Variante ist, dass man die Icons „Mülleimer“, „Statistik“ und „Stift“ zur Verfügung hat, was bedeutet, dass der angemeldete Benutzer dieses System erstellt hat oder die Administrator-Berechtigungen besitzt. Auf Funktionen der einzelnen Icons wird im Kapitel 2.6.6.9 genauer eingegangen.

### 3.6.6 MoveToMarker

Um das Finden eines Energiesystems trotz des bereits vorhandenen Adress-Suchfeldes noch leichter zu ermöglichen, gibt es die Funktion MoveToMarker. Diese Funktion wird dann ausgeführt, wenn in der Liste auf die Bezeichnung, die Katastralgemeinde oder die Postleitzahl eines Energiesystems gedrückt wird. Das Ergebnis dieser Funktion ist, dass der Benutzer nach dem Klick auf ein Energiesystem in der Liste gleich zu dessen Position auf der Karte gelangt, um eine längere Suche danach zu ersparen. Der Ablauf der MoveToMarker-Funktion wird in der Abbildung 3.46 nähergebracht.

Nachdem der Benutzer zum Standort des ausgewählten Energiesystems navigiert wurde, hat dieser dort wieder alle Funktionalitäten der Karte, wie das Hinzufügen eines neuen Energiesystems sowie einer Energietechnologie, gegeben.

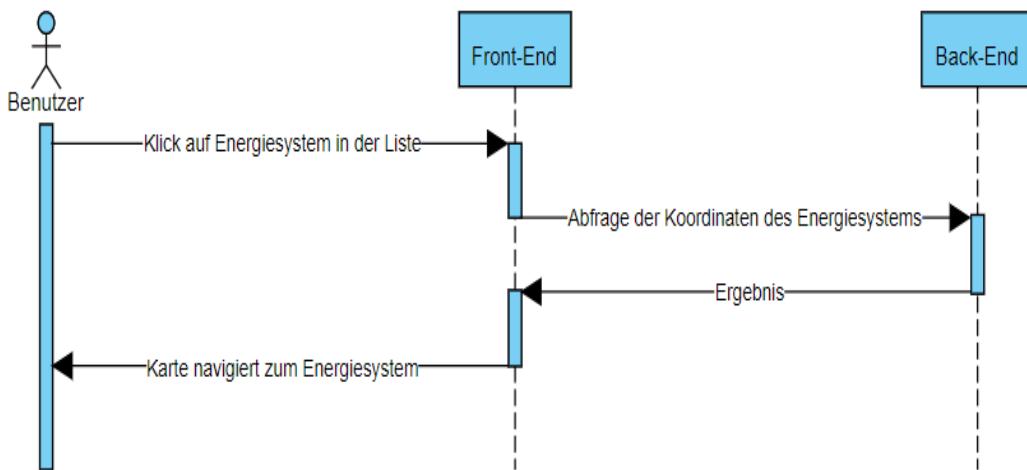


Abbildung 3.46: MoveToMarker

## 3.7 Galerie Funktionen

Auf der Seite „Galerie“ ist es möglich, die Energietechnologien von einem ausgewählten Energiesystem anzeigen zu lassen. Dabei steht das ausgewählte Bild beim Erstellen einer Energietechnologie im Vordergrund. Dieses wird in Form einer Card mit deren Bezeichnung und Beschreibung präsentiert.

### 3.7.1 Auswahl eines Energiesystems

Die Auswahl eines Energiesystems ist mithilfe eines Drop-Down-Menüs möglich. Dieses Drop-Down-Menü beinhaltet alle vorhanden Energiesysteme. In Abbildung 3.47 ist das Drop-Down-Menü mit allen vorhandenen Energiesystemen ersichtlich.

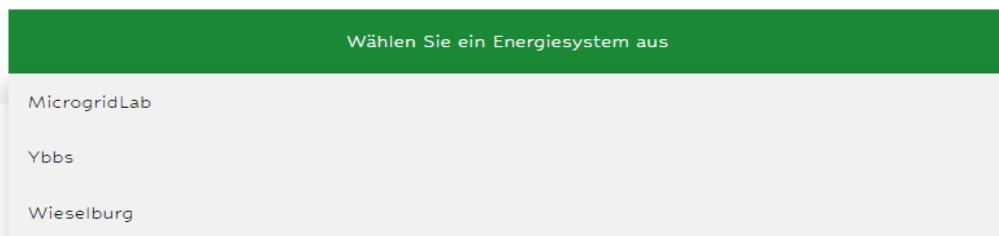


Abbildung 3.47: Auswahl eines Energiesystems

### 3.7.2 Energietechnologien des Energiesystems anzeigen

Wenn der Benutzer ein Energiesystem ausgewählt hat, werden die dazugehörigen Energietechnologien in Form von Cards dargestellt. Als Haupt-Überschrift über alle Energietechnologien dient die Bezeichnung des ausgewählten Energiesystems. Bei den einzelnen Energietechnologien, die dargestellt werden, wird zuerst das eingefügte Bild angezeigt, welches beim Erstellen einer Energietechnologie ausgewählt werden kann. Falls der Benutzer bei einer Energietechnologie kein Bild hinzugefügt hat, wird automatisch ein Standardbild eingefügt. Unter dem Bild dient die Bezeichnung sowie die Beschreibung der Energietechnologie als Bildunterschrift. In der Abbildung 3.48 sind Energietechnologien des Energiesystems MicroGridLab zu sehen.

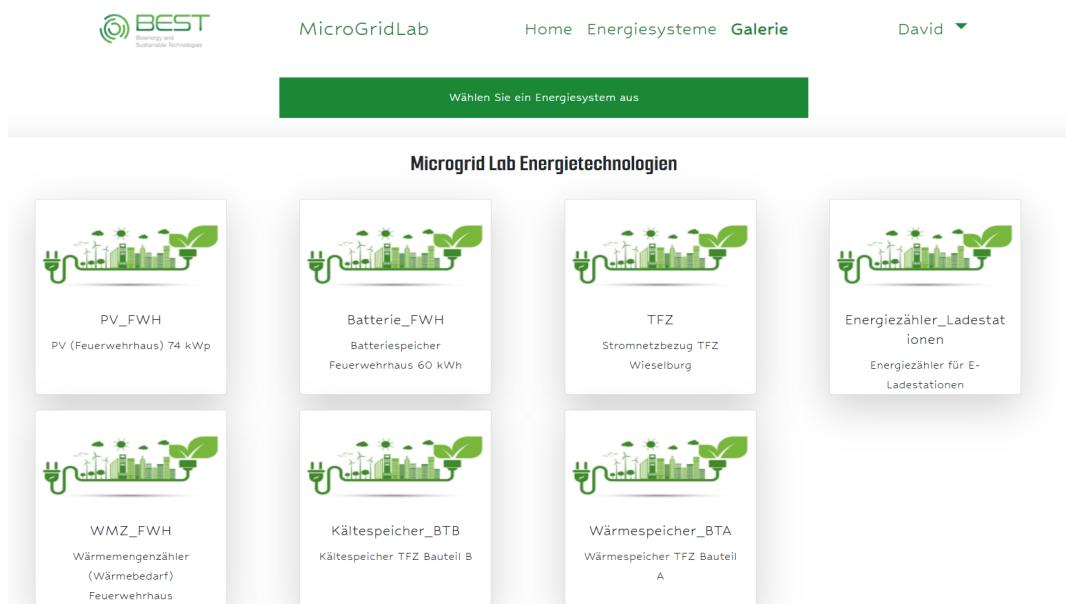


Abbildung 3.48: Galerie

## 3.8 Grafana

In diesem Kapitel wird das automatische Erstellen von Dashboards via der Grafana HTTP API beschrieben.

### 3.8.1 Automatisches Erstellen der Dashboards

Sobald in der Benutzeroberfläche ein Energiesystem erstellt wird, werden die Daten an die “store” Methode im “EnSysController” weitergereicht in welcher der API Aufruf zum Erstellen eines Dashboards getätigert wird. Hierzu werden die Daten welche vom Benutzer eingegeben wurden in das “json” Grundgerüst eingesetzt, welches dann an die API gesendet wird. Grafana setzt auf eine Bereichsverwaltung mit sogenannten “Organisations”, das Produkt erstellt die Dashboards und in diesen enthaltenen Panels in der sogenannten “annonymus organisation”. Diese Organisation ist von jedem einsehbar und erfordert keine Benutzerauthentifizierung.

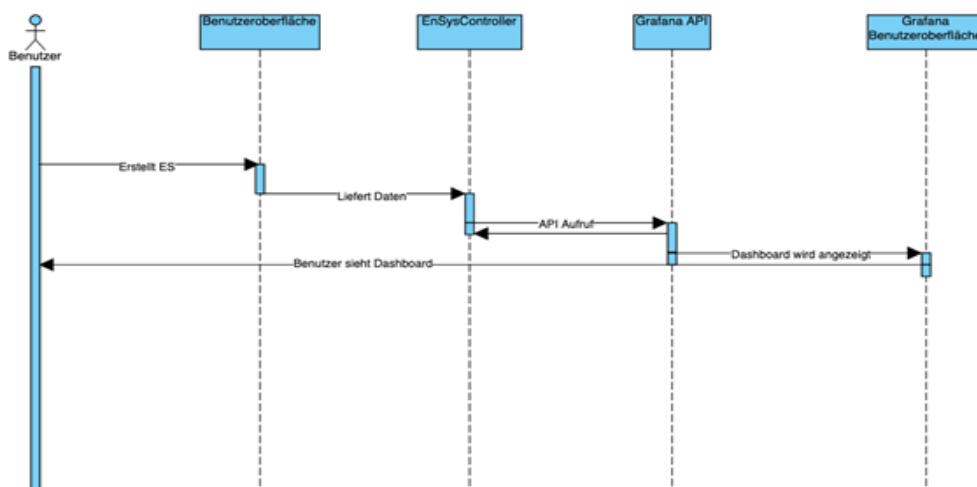


Abbildung 3.49: Dashboard erstellen

### 3.8.2 Automatisches Erstellen der Panels

Wird in einem Energiesystem eine Energietechnologie erstellt, so werden die Daten wiederum an die Methode “store” im “EnTechController” übergeben. In dieser Methode werden alle in einem Dashboard bereits vorhandenen Panels per API Aufruf ausgelesen und in einem Array gespeichert, wird nun ein neues Panel erstellt, so wird dieses dem Array hinzugefügt und mit einem API Aufruf an Grafana gesendet. Weiters wird beim Erstellen des Panels mit hilfe einer “switch” Anweisung unterschieden um welchen Typ von Energietechnologie es sich handelt, um dann die richtige SQL Abfrage zu formulieren welche die Daten aus der Datenbank in das Panel liefert.

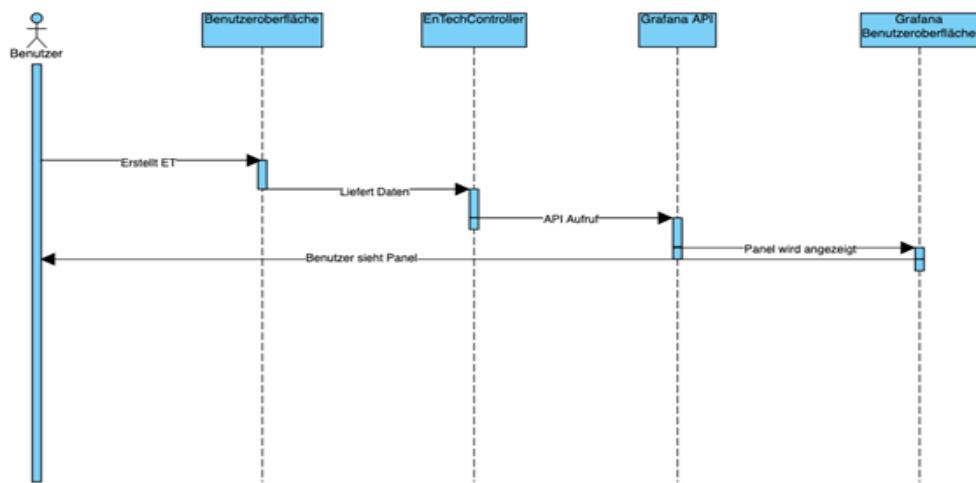


Abbildung 3.50: Panel erstellen

### 3.8.3 Energietechnologien Statistiken anzeigen

Um nun in der Benutzeroberfläche des Produkts, die mit den API Aufrufen erstellten Grafana Statistiken anzeigen zu können ist, auf der Benutzeroberfläche ein Statistik Icon vorhanden. Wird dieses vom Benutzer ausgewählt, so wird eine JavaScript Funktion welche die ID der Energietechnologie als Parameter hat ausgeführt. In dieser JavaScript Funktion wird ein Pop-Up Fenster aufgerufen sowie der "src" Parameter eines HTML "iframe" Elementes gesetzt. Dieses "iframe" Element wird nun in dem Pop-Up Fenster angezeigt.

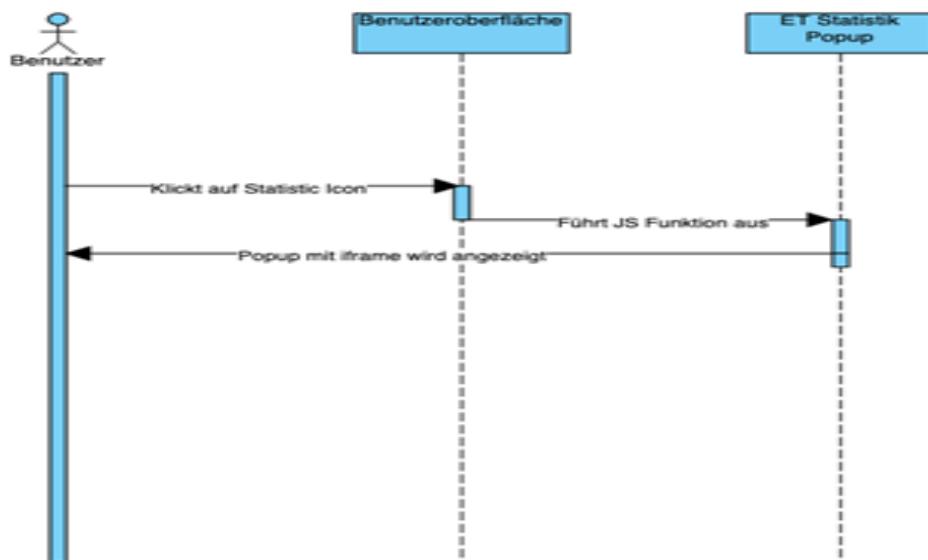


Abbildung 3.51: Statistik anzeigen

## 3.9 Einbindung von Google Maps

Um eine Karte zur Verwaltung und Visualisierung auf dem Produkt zu Verfügung zu stellen, hat sich das Projektteam für den Kartendienst Anbieter Google Maps entschieden. Um die Funktionen von Google Maps zu nutzen, muss ein Google Cloud Account erstellt und eingebunden werden. Was Google Cloud ist und wie es verwendet wird, ist in den nachfolgenden Kapiteln genauer erklärt.

### 3.9.1 Google Cloud

Google Cloud ist die zentrale Verwaltung aller von Google bereitgestellten Cloud Dienste. Das Projektteam benötigt diese Plattform, um die Google Maps Karte in das Produkt einzubinden. In den nachfolgenden Kapiteln werden alle Schritte erklärt, die es benötigt, um das Projekt zusammen mit Google Cloud verwenden zu können.

### 3.9.2 Google Cloud Plattform Account erstellen

Um Zugriff auf die Google Cloud Dienste zu erlangen ist es nötig, sich ein Konto dort zu erstellen. Aufrufbar ist die Seite über den Link : <https://cloud.google.com>. Um eine genaue Anleitung zu bekommen, wie man einen Account auf dieser Seite erstellt, ist im Dokument Google Maps API zu finden.

### 3.9.3 API's aktivieren und einbinden

Um diverse Funktionen auf der Map zu ermöglichen, werden die von Google bereitgestellten API Dienste benötigt. Diese API Dienste müssen auf der Google Cloud Website aktiviert werden. Folgende API's werden benötigt:

- Geocoding API
- Javascript API
- Places API

Nach Aktivierung der oben aufgezählten API's, muss ein dazugehöriger API Key erstellt werden. Dieser API Key ermöglicht es, mit dem Produkt auf die Google Cloud Dienste zuzugreifen. Auf Google Cloud sind die Keys unter der Navigier Möglichkeit "Anmeldedaten" ersichtlich. Dort können diese auch erstellt oder gegebenenfalls gelöscht werden. Wenn ein API Key erstellt wurde, muss dieser im Programm eingebunden werden. Er wird überall dort eingebunden, wo diese Dienste verwendet werden oder eine Anfrage an die API gesendet wird. Um genauere Informationen und eine Step by Step Anleitung zu bekommen, lesen Sie in der Datei Google Maps API nach.

### **3.9.4 Individuelle Map erstellen und einbinden**

Eine weitere Funktion, die von Google Cloud angeboten wird, ist das Erstellen einer eigenen Map. Um eine eigene Map einzurichten, muss zuerst ein Map Design erstellt werden, welches später mit der Map verknüpft wird. Das Map Design bietet die Möglichkeit, diverse Orte, Gebäude und Lokale ein oder auszublenden. Straßen oder Straßenbezeichnungen können angezeigt oder versteckt werden. Wenn die erstellte Map mit dem richtigen Map Design verbunden ist, muss ein dazugehöriger Map Key erstellt werden. Dieser Map Key muss dann in jeder Stelle im Code eingebunden werden, wo eine Google Maps Karte initialisiert wird. Um eine Step by Step Anleitung zu bekommen, lesen Sie in der Datei Google Maps API nach.

## **Kapitel 4**

# **Resümee und Ausblick**

Die Diplomarbeit ist reibungslos verlaufen, jedoch würde das Projektteam andere Ansätze wählen bei einem Projekt gleicher Art. Die Kommunikation mit dem Auftraggeber hat sehr gut funktioniert, jedoch sind bei den laufenden Meetings immer wieder neue Änderungsvorschläge vonseiten des Auftraggebers gekommen. Dadurch entstanden für das Projektteam immer wieder neue Herausforderungen, welche zusätzlich zum definierten Projektziel umgesetzt werden sollten. Am Start der Entwicklung hatte das Projektteam mehrere Komplikationen bei der Verwendung von Git, da dabei immer wieder unerklärliche Fehler aufgetreten sind. Diese Fehler konnten jedoch mit der Zeit vom Projektteam gelöst werden, womit eine reibungslose Zusammenarbeit möglich wurde. Das Projektteam wird sich nach Abschluss der Diplomarbeit darum kümmern, die letzten Änderungsvorschläge des Auftraggebers umzusetzen. Anschließend ist das Projektteam nicht länger für das Produkt und dessen Verwaltung zuständig.

## **Kapitel 5**

### **Quellen und Literatur**

## **Kapitel 6**

### **Verzeichnisse**

# Abbildungsverzeichnis

1.1	Team . . . . .	3
2.1	Vorhandene System des Auftraggebers . . . . .	5
2.2	Architektur des Zielsystems . . . . .	9
2.3	Clientseitiger Aufbau der einzelnen Elemente . . . . .	10
2.4	Laravel MVC . . . . .	12
2.5	Angular Logo . . . . .	12
2.6	ASP.NET Logo . . . . .	13
2.7	React Logo . . . . .	13
2.8	Bootstrap Logo . . . . .	15
2.9	Tailwind Logo . . . . .	16
2.10	Vue.js Logo . . . . .	16
2.11	Google Maps Logo . . . . .	21
2.12	Open Street Map Logo . . . . .	21
2.13	Beispiel eines CSS-Systems . . . . .	22
2.14	Aufbau einer Ressource Route . . . . .	31
2.15	Aufbau einer GET Route . . . . .	32
2.16	MVC Design Pattern . . . . .	32
3.1	Grafana Einbindung . . . . .	35
3.2	ER Model . . . . .	39
3.3	Design Vorschlag 1 . . . . .	41
3.4	Design Vorschlag 2 . . . . .	41
3.5	Design Vorschlag 3 . . . . .	42
3.6	Finaler Design Vorschlag . . . . .	43

3.7 Überschriften im Fließtext . . . . .	45
3.8 Überschriften außerhalb des Fließtextes . . . . .	45
3.9 Interaktion mit dem DataTable . . . . .	46
3.10 Interaktion mit dem Drop-Down in der Galerie . . . . .	47
3.11 Beispiel auf der Website . . . . .	47
3.12 Header auf der Website . . . . .	48
3.13 Logo 1 . . . . .	48
3.14 Logo2 . . . . .	48
3.15 Icons auf der Map . . . . .	49
3.16 Icon markiert ein Energiesystem . . . . .	49
3.17 Icon markiert eine Energietechnologie . . . . .	49
3.18 Icon in Übersichten . . . . .	50
3.19 Icon in Formularen . . . . .	51
3.20 Autoren Verlinkungen . . . . .	51
3.21 Icons im DataTable . . . . .	52
3.22 Beispiel eines Buttons . . . . .	52
3.23 Energiesystem erstellen Pop-up . . . . .	55
3.24 Energiesystem erstellen . . . . .	55
3.25 Energiesystem bearbeiten Pop-up . . . . .	57
3.26 Energietechnologie erstellen Pop-up . . . . .	58
3.27 Energietechnologie erstellen . . . . .	59
3.28 Energietechnologie bearbeiten . . . . .	60
3.29 Adresssuchfeld . . . . .	62
3.30 Automatische Vervollständigung . . . . .	62
3.31 Adresssuche . . . . .	63
3.32 Website Best GmbH . . . . .	64
3.33 Oberer Teil der Home Seite . . . . .	65
3.34 Unterer Teil der Homeseite . . . . .	65
3.35 Die Energiesystem Seite . . . . .	66
3.36 Die Galerie . . . . .	66
3.37 Das Impressum . . . . .	67
3.38 Die Datenschutzseite . . . . .	67
3.39 Die Registrierungsseite . . . . .	68

---

3.40 Das Login Formular . . . . .	68
3.41 Das Registrierungsformular . . . . .	69
3.42 Energiesystem auswählen . . . . .	70
3.43 Energietechnologie Marker platzieren . . . . .	71
3.44 Standard DataTable . . . . .	73
3.45 Individueller DataTable . . . . .	73
3.46 MoveToMarker . . . . .	75
3.47 Auswahl eines Energiesystems . . . . .	75
3.48 Galerie . . . . .	76
3.49 Dashboard erstellen . . . . .	77
3.50 Panel erstellen . . . . .	78
3.51 Statistik anzeigen . . . . .	78
8.1 Projektstrukturplan . . . . .	102
8.2 Verantwortungsmatrix . . . . .	103

# Tabellenverzeichnis

2.1	Energietechnologien . . . . .	7
2.2	Browser und Versionen in denen das Produkt getestet wurde . . . . .	10
2.3	Entscheidung des Frameworks . . . . .	14
2.4	Vor- und Nachteile der Front-End Templates . . . . .	17
2.5	Vor- und Nachteile der Anbieter . . . . .	23
3.1	Datenbank Anmeldeinformationen . . . . .	36
3.2	Datenbank Anmeldeinformationen Beispiel . . . . .	36
3.3	Mail Server Konfigurationen . . . . .	37
3.4	Mail Server Konfigurationen Beispiel . . . . .	37
3.5	Icons im DataTabale . . . . .	51
3.6	HTML Elemente . . . . .	53
7.1	Begleitprotokoll David Pöchacker . . . . .	92
7.2	Begleitprotokoll Tobias Kronsteiner . . . . .	94

# Codeverzeichnis

2.1	Return Statement mit Compact . . . . .	19
2.2	Zugriff mittels Foreach Schleife . . . . .	19
2.3	Attributübermittlung in einem return View Statement . . . . .	19
2.4	Zugriff im Programm . . . . .	20
2.5	Platzhalter in der Layout Datei Konfigurieren . . . . .	26
2.6	Platzhalter auf einen Wert setzen . . . . .	26
2.7	Beginn einer Section kennzeichnen . . . . .	27
2.8	Ende einer Section kennzeichnen . . . . .	27
2.9	Einbindung der Layout Datei . . . . .	27
2.10	Erstellen einer neuen Migration . . . . .	28
2.11	Aufruf der up() Methode in der Migration . . . . .	28
2.12	Befehl um den Status der Migrations zu bekommen . . . . .	28
2.13	Migrations rückgängig machen . . . . .	28
2.14	Datenbank Seeder erstellen . . . . .	28
2.15	Datenbank Factory erstellen . . . . .	28
2.16	Datenbank Seeder ausführen . . . . .	29
2.17	Erstellen eines neuen Models . . . . .	29
2.18	Erstellen eines neuen Controllers . . . . .	29
2.19	Erstellen eines neuen Resource Controllers . . . . .	29
2.20	Start des Laravel Entwicklungs Servers . . . . .	30
2.21	APP_Key generieren . . . . .	30
2.22	Alle nötigen Pakete installieren . . . . .	30
3.1	Überprüfung der Benutzerrolle . . . . .	61
3.2	DataTable Funktion . . . . .	72

## Kapitel 7

# Begleitprotokoll gem. § 9 Abs. 2 PrO-BHS

Das Begleitprotokoll repräsentiert, welches Teammitglied woran und wie lange gearbeitet hat. Für das Verfassen der Begleitprotokolle wurden folgende Abkürzungen verwendet:

- Besprechung(D, A, Bu)
  - D = Diplomant
  - A = Auftraggeber
  - Bu = Dipl.-Ing. Johann Burgstaller
- SP = Sitzungsprotokoll

## 7.1 Begleitprotokoll David Pöchacker

KW/JAHR	TÄTIGKEIT	AUFWAND (h)
13/21	Erste Besprechung (D, A, Bu)	3
17/21	Grafische Skizze vorhandenes System	1
18/21	Besprechung (D, A)	1
20/21	Besprechung (D, A, Bu), Pflichtenheft erstellt	5,5
22/21	Besprechung (D, A, Bu), SP verfasst	2,5
25/21	Begleitprotokoll, Begriffserklärung Katalog, Lastenheft erstellt Besprechung (D, A, Bu), SP erstellt	7,5
27/21	Meilensteinplan erstellt, Datenbank-Schema diskutiert Besprechung (D, A, Bu), SP verfasst	3
28/21	Besprechung (D, Bu) über den Meilensteinplan, Diplomarbeitsantrag erstellt	1,5
29/21	Projekthandbuch, MS Project Datei, Controlling Sheet erstellt	7,25
30/21	Arbeitsstunden in MS Project Datei eingetragen, Besprechung (D, Bu)	7
31/21	Besprechung (D, A), SP verfasst	1
32/21	Laravel Installation und Recherche, Back-End Entwicklung (Laravel)	4,75
33/21	Laravel Recherche, Laravel Dokumentation verfasst	5,5
34/21	Back-End Entwicklung (Views erstellt, Datenbankverbindung, Routen)	7,75
35/21	Back-End Entwicklung (Datensätze Erstellen, Löschen), GitHub eingerichtet Besprechung (D, Bu), SP verfasst, Kartendienst Implementierung	12

KW/JAHR	TÄTIGKEIT	AUFWAND (h)
36/21	Diplomarbeitsantrag in der Datenbank befüllt, Start-Präsentation durchgeführt Erste Design-Vorschläge verfasst	6
37/21	Besprechung (D, Bu) Diplomarbeitsantrag eingereicht Design-Vorschläge verbessert	5
38/21	Besprechung (D, A, Bu) bezüglich Design-Vorschläge Finalen Design-Vorschlag und Benutzerhandbuch erstellt	10
39/21	Besprechung (D, A, Bu) bezüglich finalen Design-Vorschlag, SP verfasst genaue Anforderung der Datenbankattribute und Grafana Statistiken	2,5
40/21	Besprechung (D, A) Anforderung der Datenbankattribute und Grafana Statistiken überarbeitet	1
41/21	Entwicklungsumgebung mit GitHub eingerichtet	1,5
43/21	Front-End Entwicklung (Header, Footer, Liste, Drop-Down bei Galerie Seite) Bootstrap Recherche, CSS-Anpassungen, Icons in Photoshop angepasst Finale Design in Adobe XD erstellt	10
44/21	Back-End Entwicklung (Energiesystem hinzufügen mit Pop-up) Besprechung (D, A, Bu) über aktuellen Stand der Entwicklung	6
45/21	Google Maps Einbindung Recherche Google Maps API-Dokumentation verfasst Google Cloud Account erstellt + Google Maps eingebunden	9,5
46/21	Back-End Entwicklung (Map-Funktionen, Funktionalitäten der Liste Icons, Geo-Koordinaten werden beim Erstellen automatisch eingefügt) Marker Recherche, Eigene Google Map erstellt Energiesysteme-Marker aus Datenbank auslesen und auf der Karte platzieren	10,5
47/21	Back-End Entwicklung (Authentifizierung, Energietechnologien eines Energiesystems anzeigen) Besprechung (D,A, Bu) aktuellen Stand der Entwicklung + SP verfasst Back-End Entwicklung (Energietechnologie Erstellen Funktion)	14
48/21	Attribute User-ID und ES-ID wurden zu der Energietechnologie Tabelle hinzugefügt, Funktionalitäten wieder gegeben	2
49/21	Front-End Entwicklung (Pop-ups verschönert mit CSS) Richtigen Icons der Energietechnologien auf Karte anzeigen Bei Energiesystem Auswahl werden die richtigen Energietechnologien angezeigt Adresssuche GeoCoding API für Autocomplete implementiert DataTable implementiert Karte bearbeitet (Zoom und Vollbild Button entfernt)	14,5

KW/JAHR	TÄTIGKEIT	AUFWAND (h)
50/21	Besprechung (D,A, Bu) aktuellen Stand der Entwicklung + SP verfasst Energiesystem Pop-Up um weitere Attribute wie Az-Energietechnologien erweitert Bildergalerie wird richtiges Bild einer Energietechnologie angezeigt	8,5
51/21	Besprechung (D,A, Bu) aktuellen Stand der Entwicklung + SP verfasst	0,5
52/21	Schriftliche Arbeit Inhaltsverzeichnis erstellt, Mostviertler-Bewerb Anmeldung	5
01/22	Besprechung (D, Bu) Mostviertler-Bewerb und Inhaltsverzeichnis	13,5
02/22	Besprechung (D,A, Bu) aktuellen Stand der Entwicklung + SP verfasst	0,5
03/22	Back-End Entwicklung (Bildergalerie Standardbild ,MoveToMarker Funktion implementiert, Clean Code, Register-Seite implementiert)	15
04/22	Schriftliche Arbeit	6,5
05/22	Besprechung (D,A, Bu) aktuellen Stand der Entwicklung + SP verfasst Schriftliche Arbeit	6
06/22	Schriftliche Arbeit Dokumente für die Übergabe des Projektes erstellt Datenbank alle Attribute auf Englisch geändert DSGVO und Impressum erstellt	18
07/22	LaTex Recherche und Inhaltsverzeichnis erstellt Schriftliche Arbeit	6
08/22	Schriftliche Arbeit Besprechung (D,A, Bu) aktuellen Stand der Entwicklung + SP verfasst	19
09/22	Schriftliche Arbeit, Besprechung (D,A, Bu) Go für Testphase Schriftart der Weboberfläche geändert	6
10/22	Schriftliche Arbeit	6
11/22	Schriftliche Arbeit	11

Tabelle 7.1: Begleitprotokoll David Pöchacker

## 7.2 Begleitprotokoll Tobias Kronsteiner

KW/JAHR	TÄTIGKEIT	AUFWAND (h)
13/21	Erste Besprechung (D, A, Bu)	3
17/21	Grafische Skizze vorhandenes System	1
18/21	Besprechung (D, A)	1
20/21	Besprechung (D, A, Bu), Pflichtenheft erstellt	5,5
22/21	Besprechung (D, A, Bu)	2
25/21	Begriffserklärungskatalog, Lastenheft, Vorbereitung auf Besprechung Besprechung (D, A, Bu)	6,5
27/21	Meilensteinplan, Datenbank-Schema, Recherche Skalierbarkeit Besprechung (D, A, Bu)	2,5
28/21	Besprechung (D, Bu) über den Meilensteinplan, Diplomarbeitsantrag erstellt	1
29/21	Projekthandbuch, MS Project Datei, Controlling Sheet erstellt	7,5
30/21	Arbeitsstunden in MS Project Datei eingetragen, Besprechung (D, Bu)	7
31/21	Besprechung (D, A)	1
32/21	Laravel Installation und Recherche, Back-End Entwicklung (Laravel)	5
33/21	Laravel Recherche, Laravel Dokumentation verfasst	6
34/21	Back-End Entwicklung (Views erstellt, Datenbankverbindung, Routen)	7,5
35/21	Back-End Entwicklung (Datensätze Erstellen, Löschen), GitHub eingerichtet Besprechung (D, Bu), Kartendienst Implementierung	12
36/21	Startpräsentation erstellt und präsentiert Mit Design Vorschlägen begonnen	5
37/21	Diplomarbeitsantrag überarbeitet, Design Vorschläge weiterentwickelt Besprechung (D, Bu)	6
38/21	Besprechung (D, A, Bu) Design Vorschläge fertiggestellt Benutzerhandbuch erstellt Datenbankattribute sowie Grafana Statistiken festgelegt	11
39/21	Besprechung (D, A, Bu)	2
40/21	Besprechung (D, A) Datenbank Attribute sowie Statistiken überarbeitet	1,5
41/21	Entwicklungsumgebung sowie Versionskontrolle 'Github' konfiguriert	1,5
42/21	Einbindung von Bootstrap Bootstrap Recherche Frontend Entwicklung Finaler Designvorschlag erstellt	10
44/21	Beginn der Backendentwicklung Besprechung (D, A, Bu)	6

KW/JAHR	TÄTIGKEIT	AUFWAND (h)
45/21	Besprechung (D, Bu), Finales Datenbankschema implementiert Google Cloud Platform Account erstellt Google Maps implementiert	9,5
46/21	Frontend Entwicklung weitergeführt Backend Entwicklung weitergeführt	10
47/21	Login System implementiert Benutzer Authentifizierung implementiert Datenmodell ET ES beim Löschen überarbeitet	12
48/21	Datenschema überarbeitet	1
49/21	Frontend Entwicklung, Backend Entwicklung Benutzer Authentifizierung	10
50/21	Besprechung (D, A, Bu) Grafana HTTP API Recherche	6
51/21	Besprechung(D, A, Bu) Grafana erstellen und hinzufügen von Dashboards via API implementiert	11
52/21	Schriftliche Arbeit Inhaltsverzeichnis erstellt MV - Bewerb Zusammenfassung für die Anmeldung erstellt	5
01/22	Datenbank überarbeitet Besprechung(D, Bu)	6
02/22	Besprechung (D, A, Bu)	0,5
03/22	Grafana Recherche, Grafana Problembehandlung Alternative zu Grafana gesucht (canvas.js)	6
04/22	Grafana ET als Panel abilden und dynamisch hinzufügen und löschen	6
05/22	Grafana Panel Teilen Algorithmus entwickelt Besprechung (D, A, Bu)	6
06/22	Grafana API Aufrufe auf Produktiv Server ändern und testen	6
07/22	Grafana Panel erstellen, Datasource und Statistiken erstellen	6
08/22	Deployment auf Webserver der Best GmbH und beheben aller Fehler Besprechung (D, A, Bu)	8
09/22	Verfassen der Diplomarbeit Besprechung (D, A, Bu)	6
10/22	Verfassen der Diplomarbeit Datenkatalog, Projektmanagement	7,5
11/22	Verfassen der Diplomarbeit	7,5

Tabelle 7.2: Begleitprotokoll Tobias Kronsteiner

### 7.3 Begleitprotokoll Marcel Entner

KW/JAHR	TÄTIGKEIT	AUFWAND (h)
13/21	Erste Besprechung (D, A, Bu)	3
17/21	Skizze und Übersicht der Aufgabenstellung verfasst	1
18/21	Besprechung (D, A)	1
20/21	Besprechung (D, A, Bu), Pflichtenheft erstellt	5,5
22/21	Besprechung (D, A, Bu), SP verfasst	2,5
25/21	Google Maps vs Open Street Map Dokument verfasst, Google Api getestet und Developer Account eingerichtet. Besprechung (D, A, Bu), SP erstellt	9,1
27/21	Datenbankstruktur anschauen und Meilensteinplan verfassen Besprechung (D, A, Bu), SP verfasst	3
28/21	Besprechung (D, Bu) über den Meilensteinplan, Diplomarbeitsantrag erstellt	3
29/21	Projekthandbuch, MS Project Datei, Controlling Sheet erstellt	7,25
30/21	Besprechung (D, Bu)	2
31/21	Besprechung (D, A)	1
32/21	Laravel Installation und Recherche, Video Get Started Laravel ansehen	5
33/21	Video Get Started Laravel ansehen	3
34/21	Erste Laravel Website erstellt routen getestet und Datenbank in Laravel angelegt	4
35/21	Edit und Delete Funktion hinzufügen Besprechung (D, Bu), Maps Einbindung und User Authentifikation testen	12
36/21	Diplomarbeitsdatenbanksantrag stellen Diplomarbeitskapitel Grundlagen und Methoden anfangen zu schreiben Startpräsentation erstellen und proben Linux VM aufsetzen, um zu testen, ob das Projekt mit Linux konvertibel ist.	4
37/21	Dokument Google Maps vs Open Street Map verfassen, Google Api Testung, Einrichtung eines Google Cloud Account	7
38/21	Besprechung (D, A, Bu) bezüglich Design-Vorschläge Finaler Vorschlag überarbeitet und User Manuell angefertigt	4,3
39/21	Besprechung (D, A, Bu)	2
40/21	Besprechung (D, A)	1,5

KW/JAHR	TÄTIGKEIT	AUFWAND (h)
41/21	Aufsetzen des Diplomarbeit Programms in Laravel Einrichtung von Git und Testung Git am Laptop aufsetzen Projekt pushen und Git ignore Datei anpassen Laravel Layout erstellt und Bootstrap einbinden	5
43/21	erstellen der ersten Seite(Homepage) mithilfe von Bootstrap Überarbeitung der Homepage und Einrichtung der Navigierung Seiten Layouts erstellen und auf allen Seiten anwenden Routen und Dateinamen ändern für das bessere Verständnis Diashow Icons anpassen Header und Footer einbinden und bearbeiten erstellen des Impressums	10
44/21	Video zu Google Maps Api + Laravel anschauen Besprechung (D, A, Bu)	1,5
45/21	Video zu Google Maps Api + Installation der dazugehörigen Pakete Dokument Google Maps API erstellt Open Street Map Api Nachforschung Google Cloud Registrierung und Einbindung der Karte	6
46/21	Liste aktualisieren und Icons auf die richtige Größe anpassen Pop-up neu gestaltet, switch eingebunden und formatiert Google Maps Geo Locations auslesen und in die Edit übergeben	6
47/21	Besprechung (D, A, Bu) JavaScript Funktionen der Google Map einbinden	4
48/21	Grafana lokal installiert und Einbindung auf der Website getestet	3
49/21	Map Funktionen erweitert wenn ein Energiesystem abgewählt wird, keine Energie Technologien mehr anzeigen Suchfeld einfügen	3,5
50/21	Besprechung (D, A, Bu) Galerie Funktionen schreiben, bzw. Et erstellen Modal anpassen	7
51/21	Besprechung (D, A, Bu) Grafana HTTP Api nachforschen Grafana HTML Api Testung	4
52/21	Inhaltsverzeichnis erstellen, Wettbewerbsanträge ausfüllen Inhaltsverzeichnis überarbeiten und Bewerbungstext für Wettbewerb verfassen	4
1/22	Finales Design in Adobe xd erstellen Verfassen des Dokuments Design Handbuch Inhaltsverzeichnis mit Johann Burgstaller überarbeitet Telefonat mit EAS, route /home nur für Admin freigeben Besprechung(D, A, Bu) Routen /register und /login nur für Admin erlauben	9

KW/JAHR	TÄTIGKEIT	AUFWAND (h)
2/22	Routen register wieder überarbeitet Besprechung (D, A, Bu)	2
3/22	Video Laravel Register System anschauen Registerseite testen (Fehler beim Erstellen des Benutzers) Komplette Register Seite mit allen Funktionen und anlegen des DataTables allen aktiven Benutzern + löschen Funktion zu jedem Benutzer	6
4/22	Registerseite überarbeitet Unterscheidung von Admin und Mitarbeiter beim Anlegen Erstellen des User Controllers	4
5/22	Besprechung (D, A, Bu) Layout des DataTables überarbeitet	3
6/22	Verfassen des Dokuments Google Maps Apis Passwort vergessen Funktion welche eine E-Mail verschickt konfiguriert und getestet	4
7/22	Alle css Farben anpassen auf die gleiche Farbe und alle Texte auf die gleiche Schriftart	9
8/22	Verfassen der Diplomarbeit Besprechung (D, A, Bu)	9 9
9/22	Besprechung (D, A, Bu) Verfassen der Diplomarbeit	9
10/22	Verfassen der Diplomarbeit	9
11/22	Verfassen der Diplomarbeit	9

# **Kapitel 8**

## **Anhang**

Im Anhang befindet sich eine Auflistung der Verfasser der einzelnen Kapiteln sowie die vom Projektteam verwendete Software. Ein weiteres Element des Anhangs ist die Projektplanung.

### **8.1 Verfasser der Kapitel**

In den nachfolgenden Kapiteln wird auf die Verfasser der einzelnen Kapitel eingegangen.

#### **8.1.1 David Pöchacker**

Kurzfassung der Diplomarbeit / Abstract

Danksagung

1.1 Zielsetzung und Aufgabenstellung

1.2.1 David Pöchacker

2.1 Analyse des vorhandenen Systems

2.1.1 Begriffe

2.2.1 Schutz von vertraulichen Informationen

2.2.2 Verteilung der Verwaltung von Energiesystemen und Energietechnologien

2.2.3 Visuelle Darstellung der Energiesysteme / Energietechnologien auf einer Karte

2.3 Architektur des Zielsystems

2.3.4 Clientseitig

2.3.6 Front-End Templates

2.4 Berechtigungssystem Benutzer

3.5.1 Back-End

3.5.5 Kartendienst Funktionalitäten

3.5.6 Anzeige von Energiesystemen/Energietechnologien auf der Karte

3.6 DataTable

3.7 Galerie Funktionen

7.1 Begleitprotokoll David Pöchacker

- 8.1.1 David Pöchacker
- 8.2.1 Visual Studio Code
- 8.2.7 Adobe XD
- 8.2.8 Adobe Photoshop

### **8.1.2 Marcel Entner**

- 1.2.2 Marcel Entner
- 2.3.1 Endgeräte
- 2.3.5 Framework
- 2.3.7 Verbindung der Datenbank mit Laravel
- 2.4 Visuelle Darstellung der Energiesysteme/ Energietechnologien
- 2.6 Ui/Ux Design
- 2.7 Template-Layout
- 2.8 Laravel Befehle
- 3.2 Datenbankanbindung in Laravel
- 3.5 Corporate Design
- 3.5.2 Front-End
- 3.5.3 Login
- 3.5.3 Reisterstierung
- 3.5.7 Layoutvorlage der Webseite
- 3.9 Einbindung von Google Maps
- 8.1.2 Marcel Entner
- 8.2.3 Composer
- 8.2.4 Windows Eingabeaufforderung (CMD)
- 8.2.9 LaTex
- 8.3.5 Terminplan
- 8.3.4 Meilensteinplan

### **8.1.3 Tobias Kronsteiner**

Text

## 8.2 Verwendete Software

Im folgenden Abschnitt wird die bei dieser Diplomarbeit verwendete Software präsentiert.

### 8.2.1 Visual Studio Code

Visual Studio Code ist ein von Microsoft entwickelter Quelltext Editor. Dieser Editor bietet verschiedene Programmierhilfen wie Einfärbungen oder Autovervollständigungen. Dieser Editor unterstützt standardmäßig sehr viele Programmiersprachen, jedoch können jederzeit weitere Sprachen mittels Add-ons dazu installiert werden, um das Programmieren für den Anwender zu erleichtern. Die von diesem Projektteam verwendeten Sprachen wie HTML, CSS, PHP und JavaScript werden alle von diesem Editor standardmäßig unterstützt.

### 8.2.2 Apache WebServer

Bei dem Apache Webserver handelt es sich um eine quelloffenes Projekt, welches im Jahr 1995 erschienen ist. Aktuell wird der Apache Webserver in der Version 2.4 angeboten.

### 8.2.3 Composer

Composer ist ein Paketmanager für PHP. Composer wird über die Kommandozeile ausgeführt und installiert zugehörige Abhängigkeiten eines PHP-Programmes. Informationen über Pakete, die mit Composer installiert werden können, sind auf der Website Packagist auffindbar. Das Projektteam hat dieses Programm dafür verwendet, um nach dem Git Pull, wie in 2.8.5 beschrieben, alle neuen Pakete zu installieren und um Laravel generell zu downloaden.

### 8.2.4 Windows Eingabeaufforderung (CMD)

Das CMD (“Craniomandibuläre Dysfunktion“) auch Windows-Eingabeaufforderung genannt, wurde vom Projektteam hauptsächlich verwendet, um die Befehle aus Kapitel 2.8 auszuführen. CMD ist der Kommandozeileninterpreter von Microsoft Windows und wird gebraucht, um Text Befehle in dem System auszuführen.

### 8.2.5 Github VCS und Github Desktop GUI

Bei Github handelt es sich um einen sogenannten “Version Control Service” welcher, seit 2008 auf dem Markt ist und seit 2016 zu Microsoft gehört. Github Desktop ist die Implementation einer GUI für github und ist für Windows Mac und Linux erhältlich.

### **8.2.6 phpMyAdmin**

Bei phpMyAdmin handelt es sich um eine quelloffene Implementierung eines Grafischen Editors für MySql und deren Fork MariaDB. Mit phpmyadmin ist es möglich, komplette Datenbank Systeme zu administrieren. Es ist dabei nicht erforderlich, manuel SQL Abfragen zu schreiben, dies kann alles über eine grafische Oberfläche erfolgen.

### **8.2.7 Adobe XD**

Adobe XD ist eine von Adobe Systems entwickelte Grafik-Software zum Erstellen von grafischen Benutzeroberflächen für Web-Anwendungen. Verwendbar ist dieses Tool auf mehreren Betriebssystemen wie Windows, MacOS oder Linux. Diese Software wurde dem Projektteam von der Schule bereitgestellt, da Adobe-Programme nicht kostenfrei sind. Die erstellten Design-Vorschläge für die Weboberfläche wurden mit dieser Software erstellt.

### **8.2.8 Adobe Photoshop**

Adobe Photoshop ist ein von Adobe Inc. entwickeltes Bildbearbeitungsprogramm, welches bereits weltweit sehr verbreitet und beliebt ist. Dieses Programm ist im Jahre 1990 erschienen und ist nicht lizenziert. Aufgrund dessen wurde dieses Programm dem Projektteam ebenso von der Schule zur Verfügung gestellt. Mit dieser Software wurden alle Bilder auf der Weboberfläche auf die passende Größe skaliert und bearbeitet.

### **8.2.9 LaTex**

LaTex wird vom Projektteam zum Verfassen der Diplomarbeit benutzt. Es vereinfacht mit Makros das Schreiben im Textsystem Tex. Es hat den Vorteil, dass viele Layout Strukturen bereits vorkonfiguriert sind und so jedes in LaTex verfasste Dokument ähnlich aussieht. Latex wird vorrangig auf Universitäten und Fachhochschulen verwendet.

## 8.3 Projektplanung

In diesem Kapitel wird näher auf das Projektmanagement eingegangen.

### 8.3.1 Projektkommunikation

Im Laufe der Diplomarbeit wurden zahlreiche Besprechungen mit dem Diplomarbeitsbetreuer Herrn Johann Burgstaller gehalten, um Maßnahmen sowie weitere Vorgehensweisen abzuklären. Ebenso waren die Kooperationspartner Stefan Aigenbauer, Jürgen Mitterlehner, Michael Zellinger und Armin Cosic bei diesen Besprechungen dabei, um deren Anforderungen sowie Wünsche besser umsetzen zu können. Diese Besprechungen fanden online über Skype statt, weitere Kommunikation wurde über E-Mail fortgeführt.

### 8.3.2 Projektstrukturplan

Für eine bessere Darstellung der einzelnen Projektphasen wurde ein Projektstrukturplan erstellt. In der Abbildung 8.1 ist der Projektstrukturplan ersichtlich.

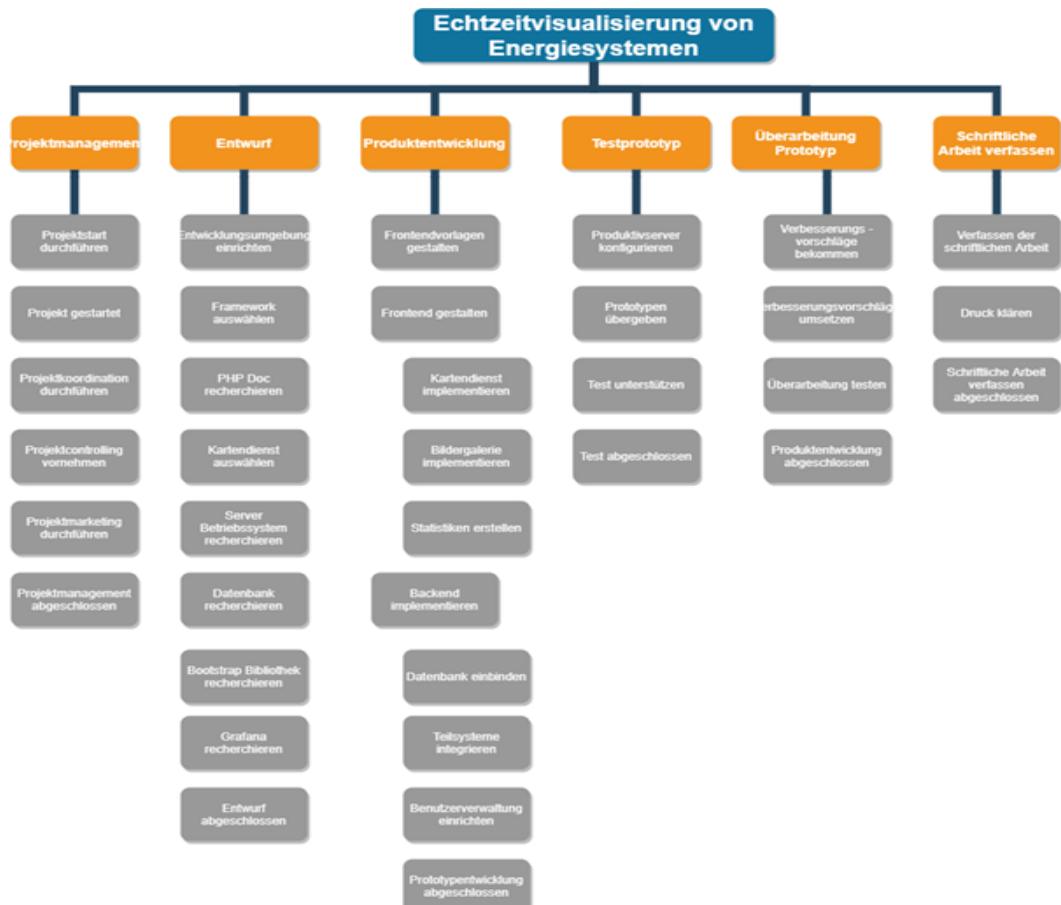


Abbildung 8.1: Projektstrukturplan

### 8.3.3 Verantwortungsmatrix

Folgende Abbildung repräsentiert die Zuständigkeiten der Mitarbeiter zu den Arbeitspaketen. Dabei wurden folgende Abkürzungen verwendet:

- Verantwortlicher (V)
- Mitarbeiter (M)
- wird informiert (I)

PSP	Aktivität	DPö	MEn	TKr	Jbu	SAi
<b>1</b>	<b>Echtzeit Visualisierung von Energiesystemen</b>					
<b>1.1</b>	<b>Projektmanagement</b>					
1.1.1	Projektstart durchführen	V	M	M	M	M
<b>1.1.2</b>	<b>Projekt gestartet</b>					
1.1.3	Projektkoordination durchführen	V	M	M	I	I
1.1.4	Projektcontrolling vornehmen	V	M	M	I	I
1.1.5	Projektmarketing durchführen	V	M	M	I	I
<b>1.1.6</b>	<b>Projektmanagement abgeschlossen</b>					
<b>1.2</b>	<b>Entwurf</b>					
1.2.1	Entwicklungsumgebung einrichten	V	M	M	I	I
1.2.2	Framework recherchieren	V	M	M	I	I
1.2.3	PHP Doc recherchieren	M	M	V	I	I
1.2.4	Kartendienst recherchieren	M	V	M	I	I
1.2.5	Server Betriebssystem recherchieren	M	M	V	I	I
1.2.6	LaTeX einrichten	V	M	M	I	I
1.2.7	Grundlagen und Methoden verfassen	V	M	M	I	I
1.2.8	Datenbank recherchieren	M	M	V	I	I
1.2.9	Bootstrap Bibliothek recherchieren	M	M	V	I	I
1.2.10	Designgrundlagen für UI/UX festlegen	M	V	M	I	I
1.2.11	Grafana recherchieren	M	M	V	I	I
<b>1.2.12</b>	<b>Entwurf abgeschlossen</b>					
<b>1.3</b>	<b>Produktentwicklung</b>					
1.3.1	Frontendvorlagen gestalten	M	M	V	I	I
<b>1.3.2</b>	<b>Frontend gestalten</b>					
1.3.2.1	Kartendienst implementieren	M	V	M	I	I
1.3.2.2	Bildergalerie implementieren	M	V	M	I	I
1.3.2.3	Statistiken erstellen	M	M	V	I	I
<b>1.3.3</b>	<b>Backend implementieren</b>					
1.3.3.1	Datenbank einbinden	M	M	V	I	I
1.3.3.2	Teilsysteme integrieren	V	M	M	I	I
1.3.3.3	Benutzerverwaltung einrichten	V	M	M	I	I
<b>1.3.4</b>	<b>Prototypentwicklung abgeschlossen</b>					
<b>1.4</b>	<b>Testprototyp</b>					
1.4.1	Produktivserver konfigurieren	M	M	V	I	I
1.4.2	Prototyp testen	V	M	M	I	I
1.4.3	Prototypen übergeben	V	M	M	I	I
1.4.4	Test unterstützen	V	M	M	I	I
<b>1.4.5</b>	<b>Test abgeschlossen</b>					
<b>1.5</b>	<b>Überarbeitung Prototyp</b>					
1.5.1	Verbesserungsvorschläge bekommen	V	M	M	I	I
1.5.2	Verbesserungsvorschläge übernehmen	V	M	M	I	I
1.5.3	Überarbeitung testen	V	M	M	I	I
<b>1.5.4</b>	<b>Produktentwicklung abgeschlossen</b>					
<b>1.6</b>	<b>Schriftliche Arbeit verfassen</b>					
1.6.1	Verfassen der schriftlichen Arbeit	V	M	M	I	I
1.6.2	Druck klären	V	M	M	I	I
<b>1.6.3</b>	<b>Schriftliche Arbeit verfassen abgeschlossen</b>					

Abbildung 8.2: Verantwortungsmatrix

DPö.... David Pöchacker

MEn... Marcel Entner

TKr.... Tobias Kronsteiner

JBu... Johann Burgstaller

SAi.... Stefan Aigenbauer (Kooperationspartner)

### 8.3.4 Meilensteinplan

Das Projektteam hat zum Plan der Diplomarbeit, einen Meilenstein Plan verwendet. Auf diesem Plan sind Meilensteine mit einem dazugehörigen Datum vermerkt. Das Datum gibt an, zu welchem Zeitpunkt dieser Meilenstein erfüllt werden soll. Folgender Meilensteinplan wurde vom Projektteam erstellt:

- 31.07.2021 Projektmanagement fertig
  - Projectcontrolling, Meilensteinplan, Projektantrag
  - Fortlaufendes Projektmanagement, Projekthandbuch,
  - Grundlagen und Methoden
- 31.08.2021 Recherche abgeschlossen
  - Vertraut machen mit dem Framework Laravel
  - Installieren der nötigen Software lokal bzw. am Server
- 29.11.2021 Produktentwicklung beendet
  - Produkt entspricht den Voraussetzungen
- 30.11.2021 Übergabe des Alpha Prototypes
  - Alpha Prototypen bereitstellen
- 10.12.2021 Feedback des Auftraggebers erhalten
  - Verbesserungsvorschläge bezüglich des Alpha Prototypen
- 31.12.2021 Alpha Prototyp überarbeitet
  - Verbesserungsvorschläge umsetzen
- 01.01.2022 Produkt fertig
  - Fertiges Produkt
- 31.03.2022 Schriftliche Arbeit verfasst
  - Verfassen der schriftlichen Arbeit

### 8.3.5 Terminplan

Während der Diplomarbeit gab es viele Termine, die vom Projektteam eingehalten werden mussten. Diese Termine wurden meistens mit unserem Ansprechpartner der Best GmbH Stefan Aigenbauer, oder mit dem Diplomarbeitsbetreuer Johann Burgstaller beschlossen. Unten eine Liste mit allen Terminen, die für diese Diplomarbeit relevant waren.

- 05.07.2021 Start des Projekts
- 31.07.2021 Projektmanagement abschließen ( Diplomarbeit Antrag fertigstellen, Arbeitspakete definieren, Projekt Controlling durchführen)
- 31.08.2021 Recherche Abgeschlossen und alle nötigen Tools/Programme/Software installiert.
- 01.09.2021 Start der Produktentwicklung
- 09.09.2021 Start Präsentation halten
- 28.11.2021 Front End Vorlagen mit dem Auftraggeber abgestimmt
- 03.02.2022 Fertigstellung einer Alpha Prototypen
- 03.03.2022 Übergabe an den Auftraggeber und beginn der Testphase
- 31.03.2022 Fertigstellung der Schriftlichen Arbeit
- 04.04.2022 Abgabe der Schriftlichen Arbeit
- 08.04.2022 Ende des Projekts

### 8.4 Inhalt von GitHub

Das Produkt ist unter dem Link [https://github.com/MarcelEntner/Echtzeit\\_Visualisierung\\_Energiesysteme.git](https://github.com/MarcelEntner/Echtzeit_Visualisierung_Energiesysteme.git) aufrufbar.