

PROJEKTARBEIT

Entwicklung einer Sprechtags Terminvereinbarung für IServ



Verfasser:	Johannes Theiner
Matrikelnummer:	7010923
Prüfer:	Prof. Dr. Niels Streekmann
Semester:	Sommersemester 2021
letzte Änderung:	14. Januar 2022

Inhaltsverzeichnis

1. Einleitung	3
2. Grundlagen	3
2.1. IServ	3
2.2. IServ Module	4
2.3. eingesetzte Frameworks & Bibliotheken	4
2.3.1. Symfony Framework	4
2.3.2. Doctrine	5
2.3.3. Twig	6
2.3.4. VueJS	6
3. Analyse	7
3.1. Anforderungsanalyse	7
3.1.1. Zielbestimmung	8
3.1.2. Produktfunktionen	8
3.1.3. Produktdaten	8
3.1.4. Qualitätsanforderungen	9
3.2. Analyse von Alternativen	10
3.2.1. Nutzung des IServ Moduls Kurswahlen	10
3.2.2. Bisherige Lösung am Max Windmüller Gymnasium	10
3.2.3. Externe Tools	10
4. Realisierung	11
4.1. Mockups	11
4.2. Vorbereitung	13
4.2.1. Installation eines Testservers	13
4.2.2. Auslieferung von Assets	14
4.2.3. Erstellung eines Debian Paketes	14
4.2.4. Continuous Integration and Delivery	15
4.3. Programmierung	15
4.3.1. Einstellungsoberfläche	15
4.3.2. Buchungsoberfläche	17
4.3.3. REST-Schnittstellen	20
4.3.4. Speichern der Einträge	22
4.3.5. Generierung von PDF Dateien	23
5. Evaluation	24
6. Zusammenfassung und Ausblick	24
6.1. Mögliche Erweiterungen	24
6.2. Probleme während der Entwicklung	25
6.3. Ausblick	25
A. Literatur	26
B. Glossar	27
C. Anhang	29
C.1. Dateistruktur	29

C.2. Ausschnitte aus dem Quellcode 31

1. Einleitung

Während der Arbeit als DigiHelfer am Max-Windmüller Gymnasium Emden sollte neben der Wartung der bereits vorhandenen EDV Anlagen der Schule auch zwei alte Webtools ersetzt werden. Das erste, ein Tool zur Erfassung von Kontaktdaten Ehemaliger Schüler wurde vor Beginn dieser Projektarbeit implementiert.

Nachfolgend sollte ein Tool entwickelt werden, welches das bereits vorhandene EspT Modul ersetzt. Durch die inzwischen hinzugekommene Tätigkeit in der Unterstützung der an der Integrierten Gesamtschule Emden eingesetzten DigiHelferin wurde festgestellt das auch diese Schule Interesse an einem IServ Modul, speziell für die Terminbuchung der Elternsprechtage hatte. Die Entwicklung dieses Moduls wird in dieser Projektarbeit betrachtet.

2. Grundlagen

Für die Entwicklung dieses Moduls sind Grundlagen in der Entwicklung von IServ Modulen nötig, welche in diesem Abschnitt beleuchtet werden.

2.1. IServ

Die IServ Schulplattform besteht aus einem Server, der in der Schule oder in einem IServ Rechenzentrum installiert ist. Optional installierbare Module, entwickelt von der IServ GmbH und Dritten(externe Entwickler & Schulen/Landkreise) fügen Funktionen hinzu.

Einige der bereitgestellten Module sind:

- Messenger & E-Mail
- Foren & News
- eigene und Gruppendateien
- Klausur- und Studienpläne
- Umfragen
- Einbindung von Infobildschirmen
- Videokonferenzen (über BigBlueButton)
- Aufgaben
- Drucken
- Softwareverteilung & Geräteverwaltung
- Fernwartung & Backup

[1]

Technisch gesehen ist ein IServ Server ein einfacher Debian Server mit bereits vorinstallierter Open Source und proprietärer Software. Einzelne Komponenten (z.B. E-Mail und Videokonferenzen) kommunizieren mit Servern der IServ GmbH um ihre Funktionalität bereitzustellen.[2]

Die Weboberfläche ist mit der Programmiersprachen PHP und JavaScript umgesetzt. Als Frameworks kommen Symfony für das Backend und VueJS für das Frontend zum Einsatz. Die Daten werden in einer Postgres Datenbank gespeichert.[3]

2.2. IServ Module

Es sind mehrere Möglichkeiten Erweiterungen für IServ zu erstellen, zum einen kann eine separate Anwendung auf Basis der Daten aus der REST-API¹ oder auf auf Basis der IServ eigenen *Portal-Web*-Schnittstelle entwickelt werden.

Ein über die Paketverwaltung in IServ installierbares Modul ist ein einfaches Debian Packet, welches über spezielle Tags identifiziert wurde und enthält den Quellcode des Moduls in Klartext, sowie bereits kompilierte Assets.

Ein solches Modul kann Funktionen aus dem Symfony Framework verwenden, sowie einige zusätzlich von IServ bereitgestellte Erweiterungen. Als Referenz während der Entwicklung dient die Dokumentation innerhalb des Demo Moduls, sowie der Quellcode der bereits vorhandenen Module, welche im Ordner `/usr/share/iserv/web/modules` liegen. Auf die Verschleierung des Quellcodes wurde bewusst verzichtet um die Kontrolle auf Sicherheitslücken und das entwickeln schuleigener Module zu vereinfachen.

Für viele in IServ verwendeten Daten und Funktionen(Abfrage von Nutzerdaten, Erstellung von Menüs, etc.) existieren in der API definierte Methoden.² Der Rest kann mit den Möglichkeiten des Symfony Frameworks und anderer bereits mit IServ mitgelieferten Bibliotheken realisiert werden.³

2.3. eingesetzte Frameworks & Bibliotheken

2.3.1. Symfony Framework

Das Symfony Projekt⁴ stellt seit 2005 eine große Liste an Komponenten⁵ zur Verfügung. Jede dieser Komponenten, oder auch Bundle genannt, hat nur eine Aufgabe(etwa Authorisierung, E-Mail oder Internationalisierung) und kann separat zu einem Projekt hinzugefügt werden(siehe Abbildung 1). Als Framework nach dem MVC Prinzip erlaubt es Symfony Aufgaben in verschiedene Dateien zu verteilen.[4]

In IServ ist jedes Modul als Bundle erstellt, so kann jederzeit auf Funktionen anderer Module zugegriffen werden. Dies ist besonders wichtig für das *CoreBundle* und das *CrudBundle* welche wichtige und häufig verwendete Funktionen enthalten.

¹<https://iserv.de/doc/development/api/>

²Häufig als Wrapper um Fremdbibliotheken

³Zusätzliche Abhängigkeiten auf andere Bibliotheken können nicht definiert werden

⁴<https://symfony.com/>

⁵<https://symfony.com/components>

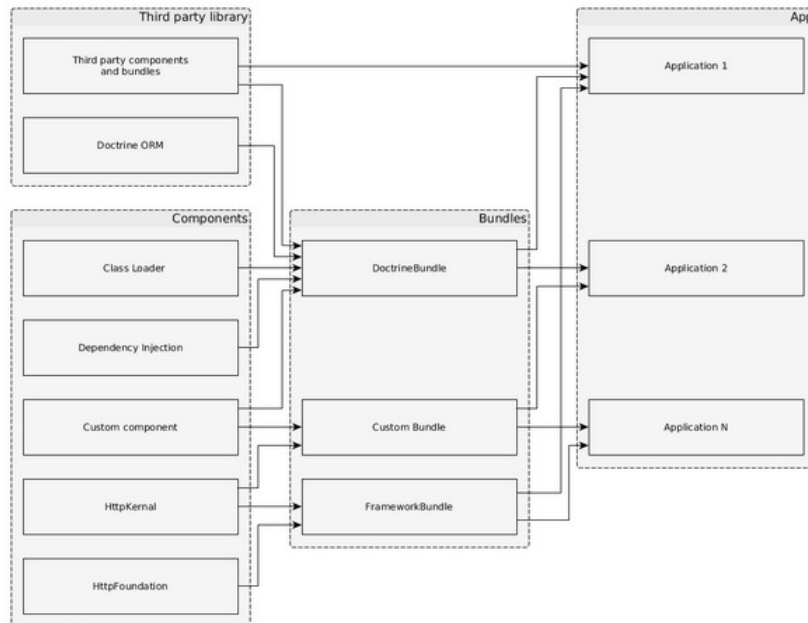


Abbildung 1: Architektur des Symfony Frameworks [5]

2.3.2. Doctrine

Das Doctrine Project⁶ stellt ebenfalls wie das Symfony Project eine Sammlung an Bibliotheken bereit⁷ die die Arbeit mit relationellen Datenbanken vereinfachen.

Für das Modul wurde die *Object Relational Mapper* Bibliothek verwendet.

Als Data Mapper stellt die Doctrine ORM Bibliothek Mappings für Eigenschaften und Datentypen zur Verfügung. Ein Beispiel kann in Listing 1 gesehen werden[6].

```
1 <?php
2 /** @Entity */
3 class Message
4 {
5     /** @Column(type="integer") */
6     private $id;
7     /** @Column(length=140) */
8     private $text;
9     /** @Column(type="datetime", name="posted_at") */
10    private $postedAt;
11 }
```

⁶<https://www.doctrine-project.org/>

⁷<https://www.doctrine-project.org/projects.html>

Auflistung 1: Beispielcode Doctrine ORM

Zur Abfrage der Daten kann die Doctrine Query Language (Beispiel in Listing 2) verwendet werden. Diese Sprache ermöglicht es direkt Abfragen auf Basis der Objekte zu stellen, ohne die darunterliegende Datenbank zu kennen[7].

```
1 <?php
2 $query = $em->createQuery('SELECT u.id FROM CmsUser u WHERE :groupId
   ↳ MEMBER OF u.groups');
3 $query->setParameter('groupId', $group);
4 $ids = $query->getResult();
```

Auflistung 2: Beispielcode DQL

2.3.3. Twig

Twig⁸ ist eine vom Symfony Projekt entwickelte Template Engine für PHP. Bei einer Template Engine werden Platzhalter innerhalb einer Vorlage ersetzt und verarbeitet.

So stellt der Beispielcode in Listing 3 eine Liste aller Nutzernamen innerhalb des in der *layout.html.twig* Datei festgelegten *content* Bereichs dar.

```
1 {% extends "layout.html.twig" %}
2 {% block content %}
3     {% for user in users %}
4         * {{ user.name }}
5     {% else %}
6         No users have been found.
7     {% endfor %}
8 {% endblock %}
```

Auflistung 3: Beispielcode Twig

2.3.4. VueJS

VueJS⁹ ist ein JavaScript Framework für die Entwicklung von graphischer Webanwendungen und existiert seit 2014. Als Templating Sprache baut Vue auf HTML auf und fügt eigene Attribute ein [8].

⁸<https://twig.symfony.com/>

⁹<https://vuejs.org/>

Der Beispielcode in Auflistung 4 iteriert so über ein Array an Nutzern und gibt deren Namen in einer Liste aus. Bei einem Klick wird die separat gepflegte `click()` Methode aufgerufen die hier den Nutzer in der Konsole ausgibt.

```
1 <template>
2 <ul class="event" v-for="user in users" :key="user.id"
  ↳ @click="click(user)">
3   <li>{{ user.name }}</li>
4 </ul>
5 <template>
6   <script>
7     export default {
8       name: 'UserList',
9       methods: {
10        click: function (user) {
11          console.log("clicked " + user)
12        },
13      },
14    }
15  </script>
```

Auflistung 4: Beispielcode VueJS

Durch die Nutzung des Javascript Ökosystems können fremde Bibliotheken und fertige Komponenten per npm¹⁰ dem eigenem Projekt hinzugefügt werden.

In IServ wird Vue für die Erstellung dynamischer Inhalte, etwa im Messenger Modul verwendet.

3. Analyse

Im folgenden werden nun die Anforderungen an das Modul erfasst und es wird überprüft ob diese nicht bereits von einer Alternative vollständig erfüllt werden.

3.1. Anforderungsanalyse

Bei der Anforderungsanalyse wird die gewünschte Funktionalität vom Kunden umschrieben und dokumentiert, auch werden nicht funktionale Anforderungen, etwa an die Performance erfasst.[9, S. 136]

¹⁰<https://www.npmjs.com/>

3.1.1. Zielbestimmung

Es wird beschrieben, welche Ziele durch den Einsatz des Produkts erreicht werden sollen.

Das Modul soll die Terminreservierung an Elternsprechtagen, sowie deren Durchführung erleichtern. Dazu bietet es Möglichkeiten zum Definieren von Zeiträumen (Dauer und Pausen) reservieren von Terminen, einladen von Schülern sowie das Anzeigen von fertigen Terminlisten.

Für eine einfache Verwaltung der Stammdaten soll das Modul in IServ eingebunden werden.

3.1.2. Produktfunktionen

Für welche Anforderungsbereiche und für welche Zielgruppen ist das Produkt vorgesehen?

Das Modul ist für den Einsatz an Schulen mit eigenem IServ Server vorgesehen.

1. Schüler/Eltern:

- a) Termine für den nächsten Sprechtag reservieren
- b) nur selbst gebuchte Termine stornieren
- c) Einsehen und Drucken der eigenen Terminliste

2. Lehrkräfte:

- a) direkte Einladung von Schülern
- b) Einsehen und Drucken der eigenen Terminliste

3. Verwaltung:

- a) Definierung der Zeiträume und Pausen
- b) Definierung der Dauer für die Terminarten (Buchung/Einladung)
- c) Definierung der Fristen für Anmeldung, Einladung und Ausdrucken der Terminlisten
- d) Definierung der Gruppenzugehörigkeit von Lehrkräften
- e) Einzelne Lehrkräfte von der Auswahlliste entfernen
- f) Einzelne Zeiträume für einzelne Lehrkräfte sperren
- g) Bedingungen für Einladungstermine stellen (z.B. maximal die ersten 6 Termine)
- h) einsehen und Ausdrucken aller Terminlisten
- i) einsehen und Ausdrucken einer Raumliste (welche Lehrkraft in welchem Raum zu finden ist)
- j) Zurücksetzen der Datenbank nach durchgeführtem Sprechtag

3.1.3. Produktdaten

Welche Daten müssen gespeichert werden ?

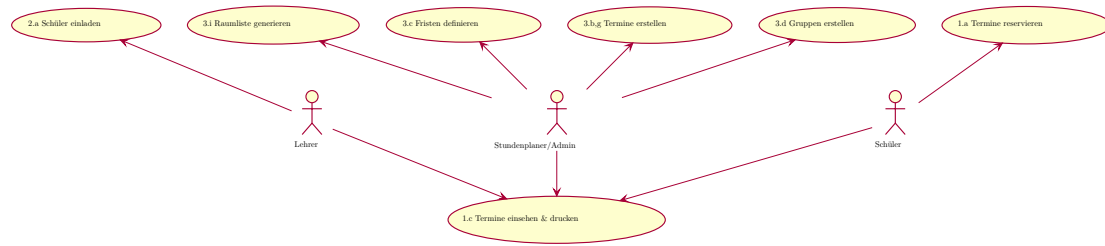


Abbildung 2: Produktfunktionen als Anwendungsfalldiagramm

1. Anfang & Ende aller Zeiträume
2. mögliche Termine & Art der Termine
3. wer hat welchen Termin gebucht
4. blockierte Termine & Lehrkräfte
5. Gruppenzugehörigkeit der Lehrkräfte
6. Raumnummern für Gruppen

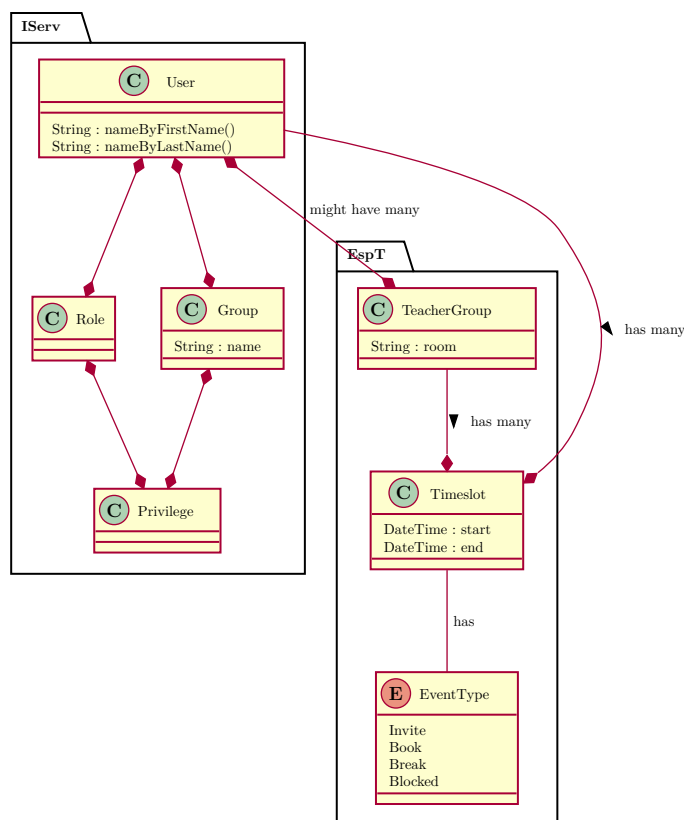


Abbildung 3: Produktdaten als Klassendiagramm

3.1.4. Qualitätsanforderungen

Die wichtigsten Qualitätsanforderungen sollten aufgeführt werden. Dazu gehören Zuverlässigkeit, Benutzbarkeit und Effizienz.

1. Buchung muss zu vorher angegebener Zeit möglich sein

2. aussagekräftige Fehlermeldung bei belegten Terminen
3. Einfache Nutzung für Schüler & Eltern auch ohne große Technik Kenntnisse
4. hohe Wartbarkeit und gute Dokumentation
5. performant, auch bei vielen Nutzern gleichzeitig

3.2. Analyse von Alternativen

3.2.1. Nutzung des IServ Moduls Kurswahlen

An der integrierten Gesamtschule Emden wird aktuell das bereits in IServ vorhandene Modul Kurswahlen entsprechend der Anleitung der GGS Sandesneben[10] eingesetzt.

Hier müssen die Wahlen von den durchführenden Lehrkräften einzeln eingestellt werden, oder es kann eine bereits erstellte Vorlage kopiert werden. Spätere Änderungen an Buchungen ist durch Administratoren und Lehrkräfte immer möglich. Bei Schülerinnen und Schülern nur, wenn die entsprechende Einstellung aktiviert ist. Zugleich können sich Schüler dann auch aus von Lehrern gestellten Einladungen austragen (Produktfunktion 1.b nicht erfüllt).

3.2.2. Bisherige Lösung am Max Windmüller Gymnasium

Am Max-Windmüller Gymnasium wurde 2010 von einem ehemaligen stellvertretenden Schulleiter mehrere kleine Webtools programmiert. Die Tools zur Dokumentation von Arbeits-, Sozialverhalten und individueller Lernentwicklung wurden seit dem durch andere Tools ersetzt. Das Elternsprechtags-Tool ist das letzte aktiv genutzte Modul und soll ersetzt werden, um den Verwaltungsaufwand zu reduzieren. So muss aktuell die Liste der Lehrkräfte und die Schüleraccounts getrennt verwaltet werden. Die entsprechenden zusätzlichen Passwörter werden zu Beginn der Schulkarriere erstellt und zusammen mit anderen Unterlagen von den Klassenlehrern verteilt.

Buchbare Termine können nur durch eine Änderung im Quellcode angepasst werden (Produktfunktion 3.a nicht erfüllt). Einladungstermine werden nicht unterstützt und müssen manuell von der Buchung ausgeschlossen werden (Produktfunktion 2.a nicht erfüllt). Zum Betrieb ist ein Webserver mit separater Datenbank notwendig.

3.2.3. Externe Tools

Externe Webdienste wie elternsprechtag-online.de, elternsprechtag-digital.de, otis-elternsprechtag.de und elterntermin.de erfordern zusätzlichen Verwaltungsaufwand, erzeugen extra Kosten und müssen zusätzlich auf Einhaltung des Datenschutzes geprüft werden. Diese Tools wurden daher nicht weiter auf die Anforderungen geprüft.

In Tabelle 1 werden die beiden Tools verglichen.

Anforderung / Tool	IServ Kurswahlen	altes EspT Modul vom MWG
1.a	ja	ja
1.b	nein	ja
1.c	nein(nur einsehen)	ja
2.a	nein	nein
3.a	ja(keine Pausen)	ja(keine Pausen)
3.b	nein(nur direkte Eingabe der Zeitfenster)	nein(siehe links)
3.c	ja	ja/
3.d	nein	nein
3.e	ja	jagit
3.f	nein	ja
3.g	nein	nein
3.h	ja	ja
3.i	nein	nein
3.j	ja	ja

Tabelle 1: Kompatibilität der getesteten vorhandenen Tools mit den Produktfunktionen aus den Anforderungen

4. Realisierung

Im folgenden Kapitel wird die Implementation der vorhandenen Anforderungen beschrieben. Die Einhaltung dieser wurde in mehreren Besprechungen gegen Ende der Implementierungszeit besprochen und das Modul von späteren Nutzern ausprobiert.

4.1. Mockups

Aus den Anforderungen wurden Mock-ups mit einem Webtool¹¹ zur besseren Visualisierung vor der Umsetzung erstellt.

Diese Mockups enthalten die wichtigsten Seiten des Moduls.

¹¹<https://wireframe.cc>

The screenshot shows a form titled 'Sprechtag am' with several input fields:

- Sprechtag am: [] von [] bis [] Uhr
- Anmeldezeitraum vom [] bis []
- Termindauer: []
- Einladungstermindauer: []
- maximale Anzahl an Einladungsterminen: []

 A section labeled 'Pausen:' contains a row of fields: 'von [] bis [] Uhr' followed by a trash icon and a plus icon. Below this is a 'Vorschau:' section with a horizontal bar divided into 12 segments, where the 6th segment is filled black. An 'Erstellen' button is located at the bottom right.

Abbildung 4: Mockup: Erstellung eines neuen Elternsprechtags

In Abbildung 4 ist ein Formular dargestellt welches für die Erstellung eines Elternsprechtags umsetzen soll. Als Daten sind neben Zeiträumen für die Anmeldung, den eigentlichen Sprechtag und die Dauer der einzelnen Zeitfenster auch Pausen vorhanden. Aus diesen Daten soll dann ein Ablaufplan erstellt werden, der in einer graphischen Vorschau gezeigt wird. (Produktfunktion 3.a,b,c & g)

The screenshot shows a booking view for parent consultation days. It features a header with the text: 'Der nächste Sprechtag ist am 20.06 von 14:00 - 18:00 Uhr. Hier können sie sich bis zum 15.06 um 20:00 Uhr ihren Termin reservieren'. Below this, there are five rows, each representing a staff member ('Lehrkraft 1' through 'Lehrkraft 5'). Each row contains a horizontal bar divided into 12 segments. The segments are color-coded: white (free), red (blocked), and black (self-booked). For example, Lehrkraft 3 has a black segment in the 6th position, while Lehrkraft 4 has a black segment in the 8th position.

Abbildung 5: Mockup: Auswahl der gewünschten Termine

Das Mockup der Buchungsansicht (Abbildung 5) besteht aus einer Tabellenartigen Ansicht bei der jede Zeile eine Lehrkraft oder eine Gruppe von Lehrkräften darstellt. Die Zeit wird durch die horizontale dargestellt. Jedes einzelne Zeitfenster ist entsprechend dem Buchungsstatus farblich gekennzeichnet (weiß = frei; rot = blockiert; schwarz = selbst gebucht). (Produktfunktion 1.a & Qualitätsanforderung 3)

			Lehrkraft		Raum
14:00	Schüler 1	9C	14:00	Lehrkraft 1	301
14:15					
14:15	Schüler 2	5A	14:15	Lehrkraft 2	102
14:30					
14:30	Schüler 3	9B	14:30	Lehrkraft 3 Lehrkraft 4	203
14:45					
14:45			14:45		
15:00			15:00		
15:00	PAUSE		15:00	PAUSE	
15:15					
15:15	PAUSE		15:15	PAUSE	
15:30					
15:30	Schüler 4	7A	15:30	Lehrkraft 5 Lehrkraft 6	Musik P1
15:45					
15:45	Schüler 5	7B	15:45	Lehrkraft 7 Lehrkraft 8	215
16:00					
16:00			16:00		
16:15			16:15		

(a) Lehrkraft

(b) Schüler

Abbildung 6: Mockup: Ausgedruckte Liste

In Abbildung 6 sind generierte Listen für Lehrkräfte und Schüler dargestellt, welche nach Ende der Buchungsperiode generiert werden können. Diese enthalten Informationen über bevorstehende Termine, wie Uhrzeit und beteiligte Personen. (Produktfunktion 1.c)

Das Feedback zu den Mockups war durchweg positiv, am Max Windmüller Gymnasium wurden die Vorstellungen genau getroffen, während an der IGS Emden ein Mockup zur Bearbeitung einzelner Termine durch Lehrkräfte gefehlt hat.

4.2. Vorbereitung

4.2.1. Installation eines Testservers

Für die Installation reicht ein einfacher Server, eine entsprechende Entwicklungslizenz kann per E-Mail bei der IServ GmbH kostenlos anfragt werden. Abweichend von der originalen Anleitung¹² muss in diesem Fall DHCP deaktiviert werden, weil im Schulinternen Netzwerk bereits ein produktiver IServ Server im Einsatz war.

Für einen externen Zugriff müssen im Router mehrere Portfreigaben¹³ erstellt werden. Da in diesem Fall bereits der produktive IServ diese verwendet, sollte vom Medienzentrum der Stadt ein weiterer Anschluss geschaffen werden. Weil dieser während der Laufzeit

¹²<https://iserv.de/doc/install/portal/>

¹³<https://iserv.de/doc/install/ports/>

des Projektes nicht freigeschaltet wurde, ist eine VPN Verbindung mit einem externen Server eingerichtet worden.

4.2.2. Auslieferung von Assets

Für das Verpacken und Konvertieren von Bildern, CSS und JavaScript Dateien wird von IServ der Bundler Webpack¹⁴ mit ausgeliefert. Dieses Tool packt verbundene Assets zusammen, versioniert und minimiert diese.

Dazu erstellt es automatisch einen Abhängigkeitsgraphen durch das rekursive Scannen der verwendeten Module von einem definierten Einsprungpunkt.[11] Weil Webpack PHP und Twig Dateien nicht automatisch scannen kann, müssen die verwendeten Assets manuell konfiguriert werden.

Um auch von IServ erkannt zu werden, müssen die hiermit erzeugten Ausgabedateien im Ordner `/usr/share/iserv/web/public/assets/$bundle` liegen. Im Twig Code können die Assets über den Befehl `{{ asset('$pfad', '$bundle') }}` eingebettet werden.

4.2.3. Erstellung eines Debian Paketes

Damit später das Paket, die zugehörigen Abhängigkeiten und Updates für dieses einfach eingespielt werden können wird das Modul in ein Debian Paket gebracht, damit die Paketverwaltung auf dem System sich um die automatische Installation von Updates kümmern kann.

Ein Debian Paket enthält unter anderem folgende Dateien:

- *control* enthält unter anderem eine Kurzbeschreibung, Name des Erstellers, Abhängigkeiten und Informationen über die unterstützten Prozessorarchitekturen
- Skripte, die bei der Installation/Deinstallation ausgeführt werden
- Prüfsummen
- und die Binärdateien.

[12, S. 189. Ff.]

Für die bessere Integration in IServ kann ein bereitgestellter Wrapper verwendet werden der unter anderem die Unterscheidung zwischen alten IServ 3 Modulen auf Basis von Symfony, neuen Modulen auf Basis der *Portal-Web* Schnittstelle¹⁵ sowie Softwarepaketen die auf die verwalteten Client-PCs installiert werden können, ermöglicht.[13]

Ist ein Paket mit den Tags `suite::iserv:portal, role::plugin` versehen, wird es als Modul in der IServ Paketverwaltung angezeigt.

¹⁴<https://webpack.js.org/>

¹⁵kann auch Symfony Schnittstellen verwenden, welchen Modus dieses Modul nutzt geht nicht aus der IServ Dokumentation hervor

Als Referenz für die den grundlegenden Aufbau und die Dateistruktur (siehe Abbildungen 16, 17 & 18; Abschnitt ??) wurde ein Modul der Stadtteilschule Blankenese¹⁶ verwendet.

Wegen eines Fehlers in der APT Komponente des verwendeten Repository Managers [14] wurde die Systemarchitektur vom Standardwert `all` auf `amd64` gesetzt ¹⁷.

4.2.4. Continuous Integration and Delivery

Unter Continuous Integration wird in der Software-Entwicklung ein Verfahren verstanden bei dem die verschiedenen Entwickler ihren Code regelmäßig mit den Änderungen der anderen Entwickler integrieren. Diese Integration, zusammen mit Tests wird üblicherweise durch ein automatisches System überprüft um Fehler zu finden.[15]

Bei der Continuous Delivery wird dafür gesorgt das jederzeit eine lauffähige Version, welche nah am aktuellen Entwicklungsstand ist, ausgeliefert werden kann. Auch dies wird üblicherweise durch ein automatisches System durchgeführt.[16]

Zur automatischen Ausführung aller Schritte zur Erstellung einer auslieferungsfähigen Version wurde auf dem Test Server ein Build Agent des CI/CD Systems TeamCity¹⁸ installiert. Dieser führt automatisiert alle vorhandenen Unit Tests aus, verarbeitet die Assets(Abschnitt 4.2.2) und erstellt das Debian Packet. Dieses wird dann in eine Instanz des Nexus Repository Managers¹⁹ hochgeladen. So wird aus jeder Änderung im Git Repositorys²⁰ automatisch ein Executable generiert.

4.3. Programmierung

4.3.1. Einstellungsoberfläche

Über mehrere Formulare im Verwaltungsbereich von IServ kann der Sprechtag verwaltet werden. Durch das Absenden des Formular für Registrierungs- und Sprechtagszeiträume wird die Datenbank zurückgesetzt und die Vorlagen für die Zeiträume übernommen. Die Bearbeitung von Zeitfenster Vorlagen, Gruppen von Lehrkräften und den vergebenen Zeitfenstern erfolgt per CRUD Formularen.

IServ stellt mit der Klasse `IServ\CrudBundle\Crud\ServiceBundle` und der Erweiterung `IServ\AdminBundle\Admin\AdminServiceBundle` eine einfache Möglichkeit bereit entsprechende CRUD Formulare zu erstellen. Dabei muss nur konfiguriert werden welche Daten, in welcher Darstellung gezeigt werden, sowie welche Aktionen mit diesen möglich sind. Das Beispiel in Listing 7(Abschnitt C.2) erzeugt so automatisch die Seiten in den Abbildungen 7, 8 & 9. Abbildung 10 zeigt die Bearbeitung einer Vorlage für

¹⁶<https://github.com/stsbl/iserv-billboard>

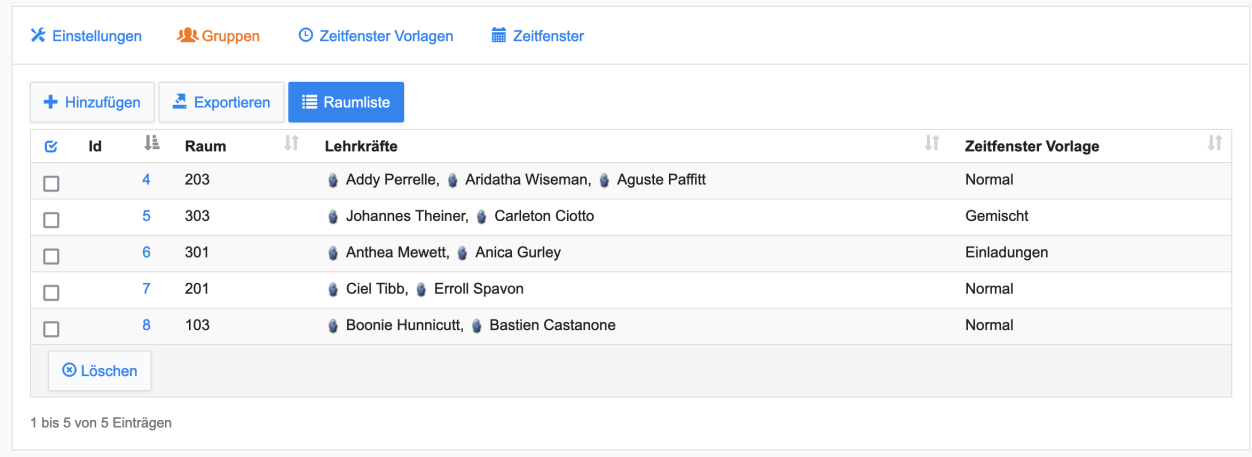
¹⁷i386 = 32 Bit, amd64 = 64 Bit

¹⁸<https://jetbrains.com/teamcity>

¹⁹<https://sonatype.com/products/repository-oss>

²⁰<https://github.com/joethei/iserv-EspT>

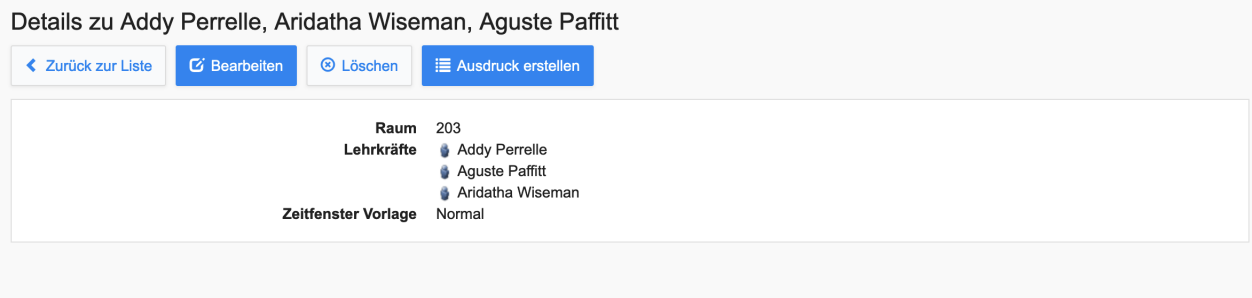
Zeitfenster mit einer dynamischen Anzahl an Datensätzen. (Implementation in Listing 8; Abschnitt C.2)



<input type="checkbox"/>	Id	Raum	Lehrkräfte	Zeitfenster Vorlage
<input type="checkbox"/>	4	203	Ady Perrelle, Aridatha Wiseman, Aguste Paffitt	Normal
<input type="checkbox"/>	5	303	Johannes Theiner, Carleton Ciotto	Gemischt
<input type="checkbox"/>	6	301	Anthea Mewett, Anica Gurley	Einladungen
<input type="checkbox"/>	7	201	Ciel Tibb, Erroll Spavon	Normal
<input type="checkbox"/>	8	103	Boonie Hunnicutt, Bastien Castanone	Normal

1 bis 5 von 5 Einträgen

Abbildung 7: Übersicht aller Gruppen

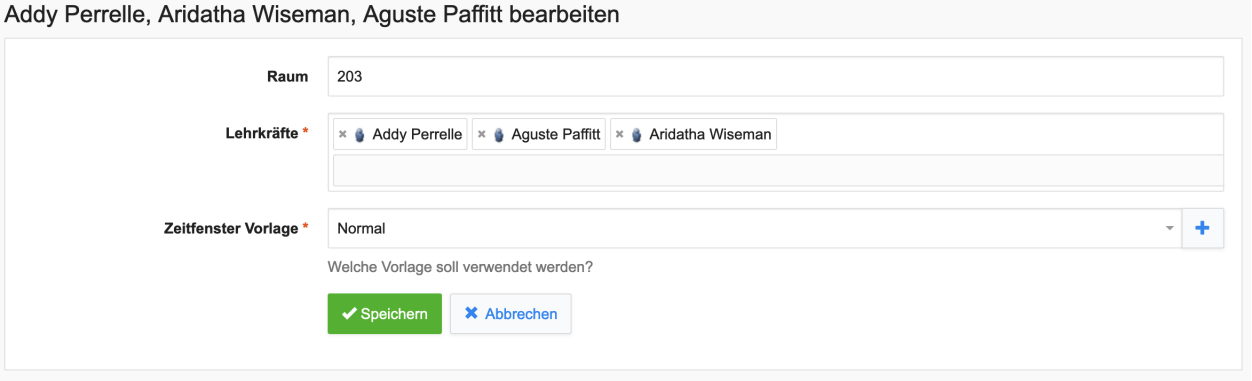


Details zu Addy Perrelle, Aridatha Wiseman, Aguste Paffitt

[← Zurück zur Liste](#)
[Bearbeiten](#)
[Löschen](#)
[Ausdruck erstellen](#)

Raum	203
Lehrkräfte	Addy Perrelle Aguste Paffitt Aridatha Wiseman
Zeitfenster Vorlage	Normal

Abbildung 8: Detailansicht einer Gruppe



Addy Perrelle, Aridatha Wiseman, Aguste Paffitt bearbeiten

Raum

Lehrkräfte *

Zeitfenster Vorlage *

Welche Vorlage soll verwendet werden?

Abbildung 9: Bearbeitung einer Gruppe

Testing bearbeiten

Name * Testing

Zeitfenster *

Beginn * 19:00

Wann soll dieser Termin starten?

Ende * 19:20

Wann soll dieser Termin enden?

Art * Einladung

Wie soll dieses Zeitfenster behandelt werden?

Beginn * -- : --

Wann soll dieser Termin starten?

Ende * -- : --

Wann soll dieser Termin enden?

Art * Blockiert

Wie soll dieses Zeitfenster behandelt werden?

Abbildung 10: Bearbeitung einer Vorlage für Zeitfenster

4.3.2. Buchungsoberfläche

Für die Darstellung der vergebbaren Termine für Lehrkräfte und Eltern wurden mehrere Komponenten für VueJS zur Darstellung eines Kalenders ausprobiert.

Folgende Anforderungen muss die später genutzte Komponente erfüllen:

1. kompatibel mit VueJS, jQuery oder normalem Javascript
2. Darstellung der Termine nebeneinander & Gruppen untereinander(ähnlich Abbildung 5)
3. Darstellung von unterschiedlich großen Zeitfenstern
4. kleinste Einheit eine Minute
5. deutsches Datumsformat(optional konfigurierbar)
6. Möglichkeit eigenen Code beim klicken auszuführen
7. Frei verfügbar und ohne Einschränkungen dauerhaft nutzbar

Im weiteren Verlauf wurden einige bereits verfügbare Komponenten auf ihre Kompatibilität mit diesen Anforderungen getestet.

Die bereits in IServ für das Kalendermodul eingebundene Komponentenbibliothek *Fullcalendar* enthält eine entsprechende Darstellung²¹, allerdings nur in der Premium Version, die von IServ nicht verwendet wird. Auch die optisch ansprechende und sonst gut geeignete *gantt-schedule-timeline-calendar*²² Komponentenbibliothek benötigt eine (kostenlose) Lizenz für jede Installation die jährlich aktualisiert werden muss(Anforderung 7).

²¹<https://fullcalendar.io/docs/timeline-view>

²²<https://www.npmjs.com/package/gantt-schedule-timeline-calendar>

Weitere Komponenten wie *jquery.schedule*²³ und *vue-scheduler-lite*²⁴ unterstützen nur volle Stunden in der Darstellung (Anforderung 4), zusätzlich ist das Datumsformat fest auf das amerikanische Format `Jahr/Tag/Monat` eingestellt (Anforderung 5).

Eine weitere Komponente, die *Siping Gantt Chart Pool* Komponente²⁵ unterstützt darüber hinaus keine Verarbeitung vom Mausclicks (Anforderung 6).

Komponente/Anforderung	1	2	3	4	5	6	7
<i>Fullcalendar</i>	ja	ja	ja	ja	ja	ja	nein
<i>Gantt Schedule Timeline Calendar</i>	ja	ja	ja	ja	ja	ja	nein
<i>jquery.schedule</i>	ja	ja	ja	nein	nein	ja	ja
<i>vue-scheduler-lite</i>	ja	ja	ja	nein	nein	ja	ja
<i>Siping Gantt Chart Pool</i>	ja	ja	ja	ja	nein	nein	ja

Tabelle 2: Kompatibilität der getesteten Komponenten mit den Anforderungen

Weil keine der getesteten Komponenten den Anforderungen entsprach (siehe Tabelle 2) wurde eine eigene Komponente implementiert, welche eine vom Backend generierte JSON Datei konsumiert die für jeden Tag Daten über die Zeitfenster (Abbildung 11) enthält.

Jeder Tag enthält Metadaten über Beginn & Ende für die Darstellung der Zeitleiste, sowie einen Faktor für die Skalierung der einzelnen Termine (Differenz zwischen Beginn & Ende, gerundet auf volle Stunden). Weiterhin enthalten sind alle Gruppen die Termine an diesem Tag anbieten mit Raumnummer, sowie alle an diesem Tag von der Gruppe angebotenen Termine.

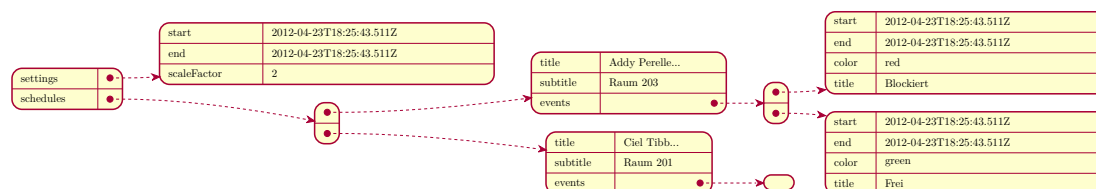


Abbildung 11: Struktur der Daten für die Darstellung im Kalender

Die Breite eines jeden Zeitfensters wird aus der Dauer in Minuten geteilt durch den Skalierungsfaktor berechnet und ist in Prozent angegeben. Die Breite einer Stunde in der oberen Zeitleiste ist dementsprechend $\frac{60}{\text{Gesamtdauer}}\%$.

Aufgrund der vorherigen Sortierung durch das Backend können die Daten durch eine einfache Schleife auf der Webseite ausgegeben werden.

```

1 //nur fortfahren wenn Buchung erlaubt.
2 if(event.color !== 'lightgreen' && event.color !== 'yellow') {
3     return;

```

²³<https://atelliee.github.io/jquery.schedule/demo/>

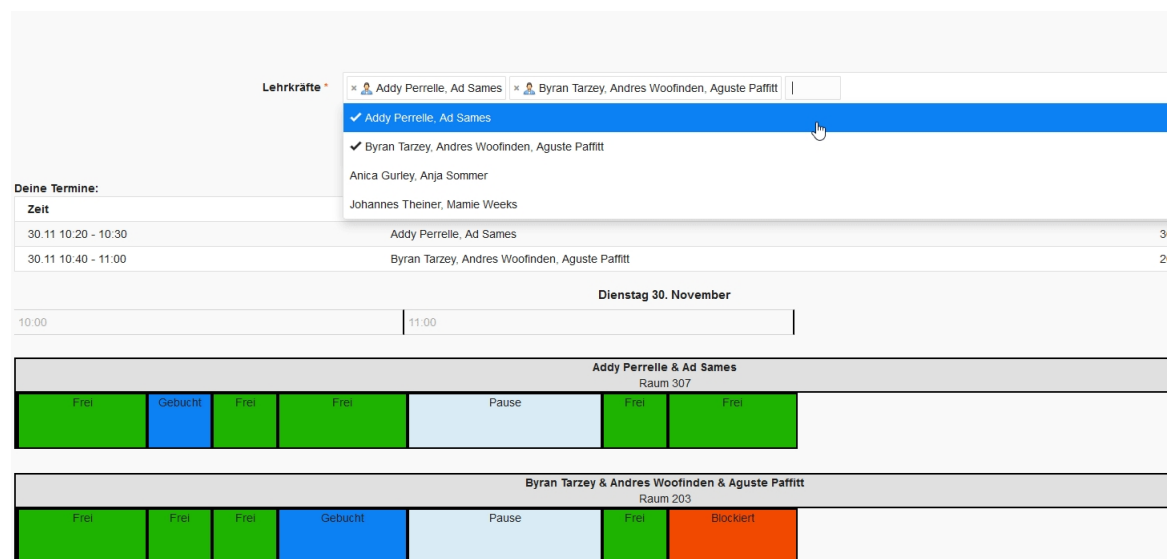
²⁴<https://linmasahiro.github.io/vue-scheduler-lite/dist/>

²⁵<https://www.npmjs.com/package/siping-gantt-chart-pool>

```
4 }
5
6 Confirm.confirm({
7   title: _('espt_confirm'),
8   //Inhalt basiert auf Farbe, da keine genaue Information über
9   //Buchungsstatus vorhanden
10  content: event.color === 'lightgreen' ? _('espt_confirm_text') :
11  //↳ _('espt_confirm_cancel_text'),
12  buttons: {
13    confirmButton: {
14      text: _('OK'),
15      btnClass: 'btn-primary',
16      action: () => {
17        //Generierung der Route zur Buchung & Aufruf
18        $.post(Routing.generate('espt_timeslots_reserve'), {id:
19        //↳ event.id}, (data) => {
20          if(data.success !== undefined && data.success ===
21          //↳ true) {
22            Message.success(_('espt_registered'), 5000,
23            //↳ false);
24
25            let event = new Event("updateData", {bubbles:
26            //↳ true});
27            document.dispatchEvent(event);
28          }else {
29            Message.error(_('Error') + ": " + data.error,
30            //↳ 50000, false);
31          }
32        });
33      }
34    },
35    cancelButton: {
36      text: _('Cancel'),
37      action: function () {
38      }
39    }
40  }
41 });
```

Auflistung 5: Implementation Click Handling Kalenderkomponente für Schüler/Eltern
Lehrkräfte werden bei einem Klick auf ein Zeitfenster auf ein Formular weitergeleitet wel-

ches die Bearbeitung des zugewiesenen ermöglicht. ²⁶ Schüler/Eltern wird der geklickte Termin sofort gebucht, sollte dies möglich und bestätigt worden sein(Listing 5).



The screenshot shows a web interface for booking appointments. At the top, there is a filter dropdown menu labeled 'Lehrkräfte' with two options: 'Addy Perrelle, Ad Sames' (selected) and 'Byran Tarzey, Andres Woofinden, Aguste Paffitt'. Below the filter is a table titled 'Deine Termine:' with columns for 'Zeit' and 'Lehrkräfte'. The table shows two appointments: one from 10:20 to 10:30 by Addy Perrelle, Ad Sames, and another from 10:40 to 11:00 by Byran Tarzey, Andres Woofinden, Aguste Paffitt. Below the table is a calendar grid for 'Dienstag 30. November'. The grid shows two rows of appointments. The first row is for 'Addy Perrelle & Ad Sames' in 'Raum 307' and shows a booking from 10:20 to 10:30. The second row is for 'Byran Tarzey & Andres Woofinden & Aguste Paffitt' in 'Raum 203' and shows a booking from 10:40 to 11:00. The grid also shows 'Frei' (free) slots and a 'Pause' (break) slot.

Abbildung 12: Buchungsansicht

Die Buchungsansicht(Abbildung 12) enthält neben der eben beschriebenen Kalenderansicht auch eine Tabelle mit allen Termine, sowie ein Formular mit dem die angezeigten Gruppen gefiltert werden können. Der hiermit erstellte Filter wird in die Datenbank geschrieben und stellt die Basis für die Generierung der JSON Datei die die Kalender Komponente speist. Anders als beim Kalender werden die Daten für die Anzeige in der Tabelle nicht gefiltert.

Der Kalender wird periodisch aktualisiert um Buchungen anderer Nutzer zu reflektieren, während die Tabelle nur bei eigenen Aktionen aktualisiert wird.

4.3.3. REST-Schnittstellen

Um auf die in der Datenbank gespeicherten Zeitfenster zuzugreifen wurden mehrere Schnittstellen nach RESTful HTTP Standard definiert. Dabei werden Ressourcen auf dem Server per HTTP Methoden gelesen und manipuliert[17, S. 149. Ff.].

GET /timeslots In diesem Endpunkt werden die Daten für die Darstellung der Kalender Ansicht im Frontend (Abschnitt 4.3.2) generiert. Dazu werden die Zeitfenster entsprechend der Privilegien des Nutzers aus der Datenbank gelesen. Lehrkräfte können so nur Einladungstermine sehen, Schüler/Eltern können alle Termine, außer Einladungstermine sehen. Zusätzlich werden für Schüler/Eltern die Einträge durch den aus der Datenbank gelesenen Filter für angezeigte Gruppen weiter eingeschränkt. Die Daten aus der Datenbank werden im Anschluss in das von der Kalender Komponente verwendete Format umgewandelt.

²⁶Die eingendlich geplante Nutzung eines modalen Dialogs konnte nicht umgesetzt werden weil die vorhandene API aufgrund (vmtl.) fehlender Konfiguration nicht funktioniert hat.

GET /timeslots/user Dieser Endpunkt stellt die Daten für die Darstellung der Tabelle im Frontend (Abschnitt 4.3.2) im JSON Format. Dazu werden die Daten entsprechend den Berechtigungen des Nutzers aus der Datenbank gelesen (Listing 6; Zeile 7-11) und für die Darstellung aufbereitet (Listing 6; Zeile 14-28).

```
1  /**
2  * @param TimeslotRepository $timeslotRepository
3  * @return Response
4  * @Route("/user", name="espt_timeslots_user", options={"expose": true})
5  */
6  public function timeslotsForUser(TimeslotRepository
7  ↪ $timeslotRepository) : Response {
8  if($this->isGranted(Privilege::TEACHER)) {
9  $timeslots =
10 ↪ $timeslotRepository->findForTeacher($this->authenticatedUser());
11 }else {
12 $timeslots =
13 ↪ $timeslotRepository->findForUser($this->authenticatedUser());
14 }
15 $result = array();
16
17 /** @var Timeslot $timeslot*/
18 foreach ($timeslots as $timeslot) {
19
20 //don't show timeslots without bookings
21 if($timeslot->getUser() != null) {
22 $result[] = array(
23 'start' => $timeslot->getStart()->format("d.m G:i"),
24 'end' => $timeslot->getEnd()->format("G:i"),
25 'type' => $timeslot->getType()->getName(),
26 'user' => $timeslot->getUser()->getNameByFirstname(),
27 'group' => implode(', ', $timeslot->getGroup()->getUsers()->toArray()),
28 'room' => $timeslot->getGroup()->getRoom()
29 );
30 }
31 }
32 }
33 return $this->json($result);
34 }
```

Auflistung 6: Implementation des Endpunkts für Termine eines Nutzers

POST /timeslots/reserve/{id} Dieser Endpunkt wird aufgerufen, wenn ein Schüler/Elternteil auf einen der noch freien oder einen vorher selbst gebuchten Termin klickt. Dabei wird der Eintrag für den zugewiesenen Nutzer des geklickten Termins in der Datenbank geändert, bei der Buchung wird zuvor noch validiert. Die Bedingungen dieser Validierung können in der Konfiguration des Moduls angepasst werden und lassen standardmäßig keine Überschneidungen mit bereits gebuchten Terminen, sowie keine Mehrfachbuchungen bei einer Gruppe zu. Zusätzlich wird auf keinen Fall die Buchung eines bereits von einem anderen Nutzer gebuchten Termins zugelassen.

4.3.4. Speichern der Einträge

Mit der Einstellungsoberfläche(Abschnitt 4.3.1) können die Daten in den Tabellen für Lehrergruppen, Zeitfenster Vorlagen und Einstellungen bearbeitet werden.

Bei der Erstellung eines Elternsprechtags werden die Uhrzeiten aus den Vorlagen um das Datum des jeweils zugeordneten Tages ergänzt und in eine neue Tabelle *espt_timeslot* eingetragen. Alle Nutzeroberflächen ziehen ihre Daten aus dieser Tabelle, kombiniert mit der Tabelle für Lehrergruppen *espt_teacher_group*.

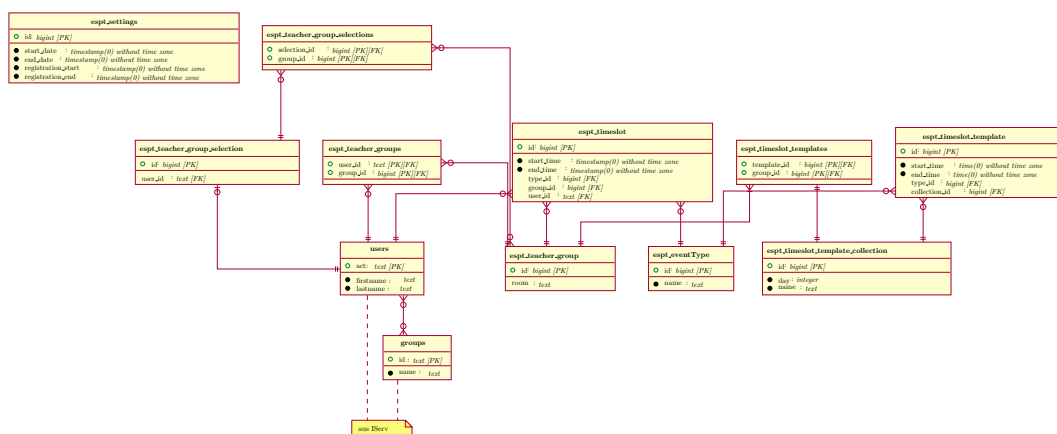


Abbildung 13: Entity Relationship Diagram

In den Tabellen *espt_timeslot_template*, *espt_timeslot_templates* und *espt_timeslot_template_collection* werden die Daten zu den vorhandenen Vorlagen gespeichert.

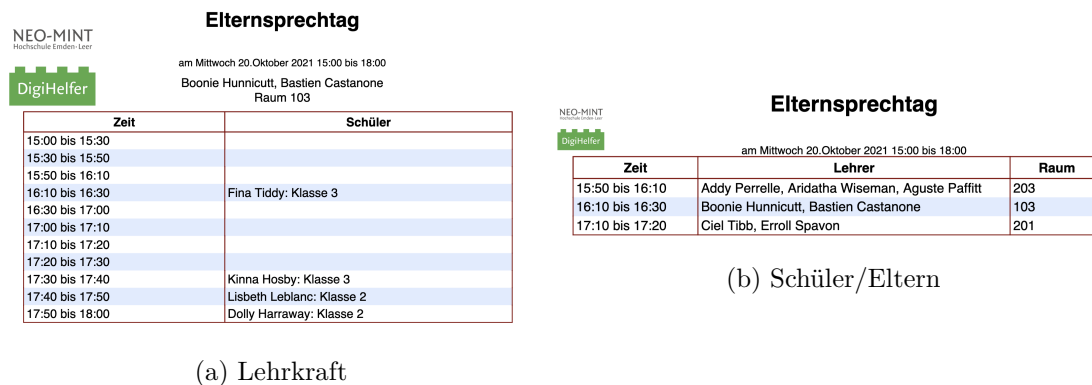
Die Tabellen *espt_teacher_group* und *espt_teacher_group* bilden die Lehrergruppen ab, wobei mehrere Lehrkräfte in mehreren Gruppen Mitglied sein können.

Für die Speicherung der Filterung in der Buchungsansicht(Abschnitt 4.3.2) sind die Tabellen *espt_teacher_group_selection*(Zuordnung zum Nutzer) & *espt_teacher_group_selections* verantwortlich. Die Tabelle *espt_eventType* ist die Datenbank Version des *EventType* Enums im PHP Code und enthält die vorhandenen Terminarten(Buchbar, nur Einladung, Pause, Blockiert).

4.3.5. Generierung von PDF Dateien

IServ liefert zwei PHP Bibliotheken zur Generierung von PDF Dateien mit, TCPDF²⁷ erstellt diese per API Methoden und dompdf²⁸ konvertiert vorhandenes HTML in PDF.

Aufgrund der bereits vorhandenen Beispiele auf der Webseite die nur leicht angepasst werden müssten wurde die TCPDF Bibliothek verwendet.



(a) Lehrkraft

Zeit	Schüler
15:00 bis 15:30	
15:30 bis 15:50	
15:50 bis 16:10	
16:10 bis 16:30	Fina Tiddy: Klasse 3
16:30 bis 17:00	
17:00 bis 17:10	
17:10 bis 17:20	
17:20 bis 17:30	
17:30 bis 17:40	Kinna Hosby: Klasse 3
17:40 bis 17:50	Lisbeth Leblanc: Klasse 2
17:50 bis 18:00	Dolly Harraway: Klasse 2

(b) Schüler/Eltern

Zeit	Lehrer	Raum
15:50 bis 16:10	Addy Perrelle, Aridatha Wiseman, Aguste Paffitt	203
16:10 bis 16:30	Boonie Hunnicutt, Bastien Castanone	103
17:10 bis 17:20	Ciel Tibb, Erroll Spavon	201

Abbildung 14: Ausgedruckte Liste

Während in der Liste für Lehrkräfte (Abbildung 14a) alle Zeitfenster angezeigt werden, auch leere um die Eintragung am Elternsprechtag selbst per Stift zu ermöglichen, werden in der Liste für Schüler/Eltern (Abbildung 14b) nur die zuvor gebuchten Termine angezeigt.

In beiden Dateien sind die Termine chronologisch sortiert. Ist in den IServ Einstellungen ein Logo definiert wird dieses auch in den PDF Dateien eingefügt.

Für den Aushang einer Raumliste zur besseren Orientierung am Sprechtag selbst (Abbildung 15, Implementierung in Listing 9, Abschnitt C.2) kann diese jederzeit in der Verwaltungsoberfläche generiert werden (siehe Abbildung 7).



Elternsprechtag

am Mittwoch 20. Oktober 2021 15:00 bis 18:00

Raum	Lehrer
301	Anthea Mewett, Anica Gurley
303	Johannes Theiner, Carleton Ciotto
201	Ciel Tibb, Erroll Spavon
103	Boonie Hunnicutt, Bastien Castanone
203	Addy Perrelle, Aridatha Wiseman, Aguste Paffitt

Abbildung 15: Generierte Raumliste

²⁷<https://tcpdf.org/>

²⁸<https://github.com/dompdf/dompdf>

Auf die Terminlisten sowohl für Lehrkräfte und Schüler/Lehrer kann nach Beendigung der Registrierungsperiode bis zum Ende des Elternsprechtags im Modul zugegriffen werden. In der Verwaltungsoberfläche kann die Liste für Gruppen(siehe Abbildung 8) jederzeit abgerufen werden.

5. Evaluation

Nach Abschluss der Implementierung des Moduls wurde den Verantwortlichen am Max-Windmüller Gymnasium ein Zugang zum Testsystem gegeben. Durch den fehlenden Online Zugang konnte das Modul nicht ausführlich von der IGS getestet werden, sondern nur an einem extra vereinbarten Termin.

Beiden Schulen fehlte die Möglichkeit die Anzahl angezeigter Lehrkräfte zu reduzieren, während am MWG dies über einen vom Nutzer definierten Filter geschehen sollte, wurde an der IGS eine Einstellung der Sichtbarkeit für Klassen in der Gruppenerstellung bevorzugt. Aufgrund der einfacheren Implementierung und der flexibleren Nutzung eines Filters wurde dieser für die Implementierung gewählt.

Weiterhin ist an beiden Schulen die Durchführung eines Sprechtags geplant gewesen, der Termine an mehreren Tagen anbieten soll. Zur Erfüllung dieser neuen Anforderung wurden die Vorlagen um einen neuen Parameter ergänzt, der dies abbildet.

Im produktiven Betrieb konnte das Modul noch nicht getestet werden, weil am dafür festgelegten Zeitpunkt noch Fehler im Modul vorhanden waren.

6. Zusammenfassung und Ausblick

Aufgrund von Unterschieden in der Organisation der beiden Schulen mussten einige Funktionen nur durch Optionen ermöglicht werden. So wird die Planung der Termine am MWG von einer Lehrkraft durchgeführt werden, die auch für die Planung des Stundenplanes zuständig ist. An der IGS soll dies von den Lehrkräften selbst gemacht werden. Ebenso sollte an der IGS auch die Bearbeitung der Buchungen während und nach der Registrierungsperiode durch die einzelnen Lehrkräfte möglich sein.

6.1. Mögliche Erweiterungen

Das Modul könnte besser in IServ integriert werden, so entspricht das Design für die Buchungsoberfläche und das der PDF-Dateien nicht dem der von IServ selbst erstellten Module. Ebenfalls könnte sowohl der Elternsprechtage selbst automatisch in den Schulöffentlichen Kalender und die gebuchten/eingeladenen Zeitfenster im Kalendermodul angezeigt werden.

Außerdem wäre eine automatische Erkennung von Pausen denkbar, aktuell muss manuell sichergestellt werden das keine Lücken zwischen einzelnen Zeitfenstern entstehen, um

keine fehlerhafte Darstellung zu erzeugen.

Auch das sich Zeitfenster innerhalb einer Vorlage und Vorlagen übergreifend nicht überschneiden muss aktuell manuell sichergestellt werden um, eine fehlerhafte Darstellung zu vermeiden.

6.2. Probleme während der Entwicklung

Eine bessere Dokumentation der vorhandenen API's seitens IServ wäre wünschenswert, aufgrund der häufig fehlerhaften oder nicht vorhandenen Dokumentation musste vielfach der Quellcode der vorhandenen Module durchsucht werden, um die korrekte Nutzung der gesuchten API zu erfahren.

Aufgrund der fehlenden Dokumentation wie ein Testsystem erstellt wird, wurde außerdem sämtliche Entwicklung per Fernzugriff durchgeführt,

Aufgrund der Abhängigkeit von IServ Code existiert nur ein Unit-Test der die Berechnung der aktuellen Phase (Einladung, Buchung oder Ausdrucken freigeschaltet) überprüft. Ein weiterer Unit-Test der die Generierung der JSON Datei für das Frontend(Abschnitt 4.3.2) konnte nicht verwendet werden, weil die zu testende Funktion zu viele IServ Klassen als Parameter verwendet um diese vernünftig per Mocking zu testen.

6.3. Ausblick

Am Max-Windmüller Gymnasium ist es geplant das Modul am nächsten Elternsprechtag zu verwenden, von der IGS gibt es hierhin gehend keine Rückmeldung.

Eine Antwort seitens der IServ GmbH zur Inklusion des Moduls in die Liste direkt installierbarer Module oder die Übernahme des Quellcodes steht noch aus.

A. Literatur

- [1] IServ GmbH. „IServ Broschüre.“ (4. März 2021), Adresse: https://iserv.de/downloads/flyer/032021_IServ_Broschuere_Auflage08.pdf (besucht am 08.05.2021).
- [2] IServ GmbH. „Backend-Server.“ (o. D.), Adresse: <https://iserv.de/doc/install/backend-server/> (besucht am 25.10.2021).
- [3] IServ GmbH. „Open Source-Software bei IServ.“ (o. D.), Adresse: <https://iserv.de/doc/development/opensource/> (besucht am 25.10.2021).
- [4] B. Paz. „An introduction to Symfony, The foundation of modern PHP applications.“ (24. Dez. 2018), Adresse: <https://dev.to/brpaz/an-introduction-to-symfony--the-foundation-of-modern-php-applications-ehj> (besucht am 10.11.2021).
- [5] „Symfony - Architecture,“ Tutorialspoint. (o. D.), Adresse: https://www.tutorialspoint.com/symfony/symfony_architecture.htm (besucht am 30.11.2021).
- [6] „Basic Mapping, Doctrine Object Relational Mapper,“ Doctrine Project. (o. D.), Adresse: <https://www.doctrine-project.org/projects/doctrine-orm/en/2.10/reference/basic-mapping.html> (besucht am 28.11.2021).
- [7] „Doctrine Query Language,“ Doctrine Project. (o. D.), Adresse: <https://www.doctrine-project.org/projects/doctrine-orm/en/2.10/reference/dql-doctrine-query-language.html#doctrine-query-language> (besucht am 28.11.2021).
- [8] G. Roden. „Was man über Vue.js wissen sollte,“ heise Developer. (30. Nov. 2020), Adresse: <https://www.heise.de/developer/artikel/Was-man-ueber-Vue-js-wissen-sollte-4969211.html> (besucht am 05.11.2021).
- [9] M. Geirhos, *IT-Projektmanagement was wirklich funktioniert - und was nicht*. Bonn: Galileo Press, 2011, ISBN: 978-3-8362-1773-6.
- [10] GGS Sandesneben. „Kurswahlen - Wahl erstellen für die Elternsprechtag.“ (16. Feb. 2021), Adresse: <https://www.youtube.com/watch?v=r6yoAa8U7Bo> (besucht am 23.09.2021).
- [11] E. Hlusko und S. Larkin. „Dependency Graph.“ (o. D.), Adresse: <https://webpack.js.org/concepts/dependency-graph/> (besucht am 25.10.2021).
- [12] D. Deimeke, S. Kania, C. Kühnast, D. van Soest und P. Heinlein, *Linux-Server, Das umfassende Handbuch*, 3. Aufl. Bonn: Galileo Press, 2014, ISBN: 978-3-8362-3020-9.
- [13] IServ GmbH. „Paketierung.“ (o. D.), Adresse: <https://iserv.de/doc/development/modules/packaging/> (besucht am 05.05.2021).
- [14] A. Evseev. „'all' arch apparently not honored by repo.“ (11. Feb. 2021), Adresse: <https://issues.sonatype.org/browse/NEXUS-26684> (besucht am 03.11.2021).
- [15] M. Fowler. „Continuous Integration.“ (1. Mai 2006), Adresse: <https://martinfowler.com/articles/continuousIntegration.html> (besucht am 10.11.2021).
- [16] S. Augsten. „Was ist Continuous Delivery?“ (8. Dez. 2017), Adresse: <https://www.dev-insider.de/was-ist-continuous-delivery-a-664429/> (besucht am 10.11.2021).

- [17] K. Spichale, *API-Design, Praxishandbuch für Java- und Webservice-Entwickler*, 2. Aufl. Heidelberg: dpunkt.verlag, 1. Mai 2019, ISBN: 978-3-86490-611-4.
- [18] „JSON.“ (o. D.), Adresse: <https://www.json.org/json-de.html> (besucht am 30.11.2021).
- [19] M. Geirhos, *Entwurfsmuster*. Rheinwerk Verlag GmbH, 30. Juni 2015, 643 S., ISBN: 978-3-8362-2762-9.
- [20] M. Fowler. „Data Mapper, Patterns of Enterprise Application Architecture Catalog.“ (o. D.), Adresse: <https://martinfowler.com/eaCatalog/dataMapper.html> (besucht am 28.11.2021).
- [21] M. Fowler. „Mocks Aren't Stubs.“ (2. Jan. 2007), Adresse: <https://martinfowler.com/articles/mocksArentStubs.html> (besucht am 30.11.2021).

B. Glossar

Application Programming Interface (API) Eine Programmierschnittstelle ermöglicht den Zugriff auf Daten und Funktionen externer Systeme . 3, 19, 24

Create, Read, Update, Delete (CRUD) Akronym für die vier fundamten Operatio-
nen des Datenmanagement: Neue Daten schreiben, Daten lesen, Daten aktualisie-
ren und Daten löschen . 14

Hypertext Transfer Protocol (HTTP) Protokoll zur Übertragung von Internetseiten .
19

JavaScript Object Notation (JSON) Einfaches Maschinen & Menschenlesbares Daten-
austauschformat. Basiert auf einer Teilmenge der Programmiersprache JavaScript[18]
. 17, 19, 20

Model View Controller (MVC) Architekturmuster/Entwurfsmuster/Designmuster bei
dem Datenhaltung, Präsentation und Steuerung getrennt werden[19, S. 520 f.] . 3

REpresentational State Transfer (REST) Softwarearchitekturstil für statuslose Kom-
munikation, bei der Ressourcen eindeutig identifiziert werden[17, S. 149 f.] . 3,
19

Virtual Private Network (VPN) Ein virtuell erstelltes Netzwerk welches entfernte Re-
sourcen zugreifbar macht. Die Verbindung ist üblicherweise verschlüsselt . 13

Adapter lat. "Anpassung". Strukturmuster das die Nutzung inkompatibler Schnittstellen
ermöglicht[17, S. 114].

Asset Externe Dateien die für die Nutzung der Webseite wichtig sind. 3

Data Mapper Ein Data Mapper bildet das relationele Datenbank Schema als Objekte
dar[20]. 4

Mock-up Attrappe der Nutzerschnittstelle verwendbar bei ersten Evaluationen. Häufig als simple Zeichnung auf Papier oder Computer. 10

Mocking eng. "etwas vortäuschen", Platzhalter für echte Objekte bei der Durchführung von Modultests[21]. 24

Wrapper eng. "Verpackung/Umschlag" , *siehe* Adapter. 3, 13, *siehe* Adapter

C. Anhang

C.1. Dateistruktur

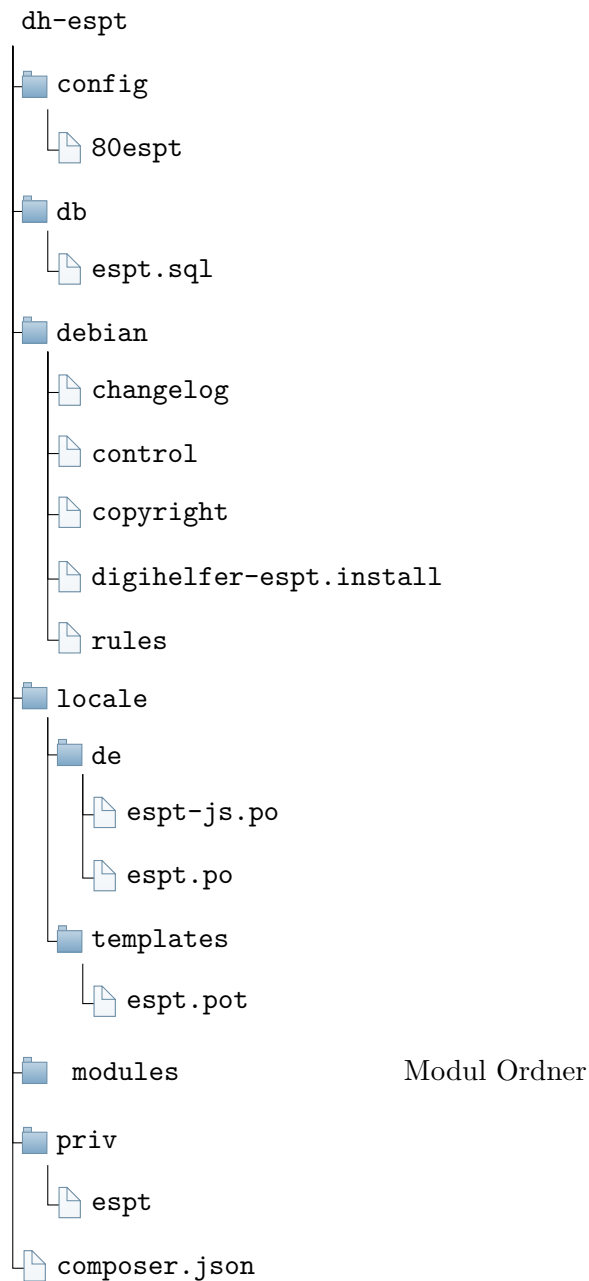


Abbildung 16: Ordnerstruktur

DigiHelfer/EspTBundle

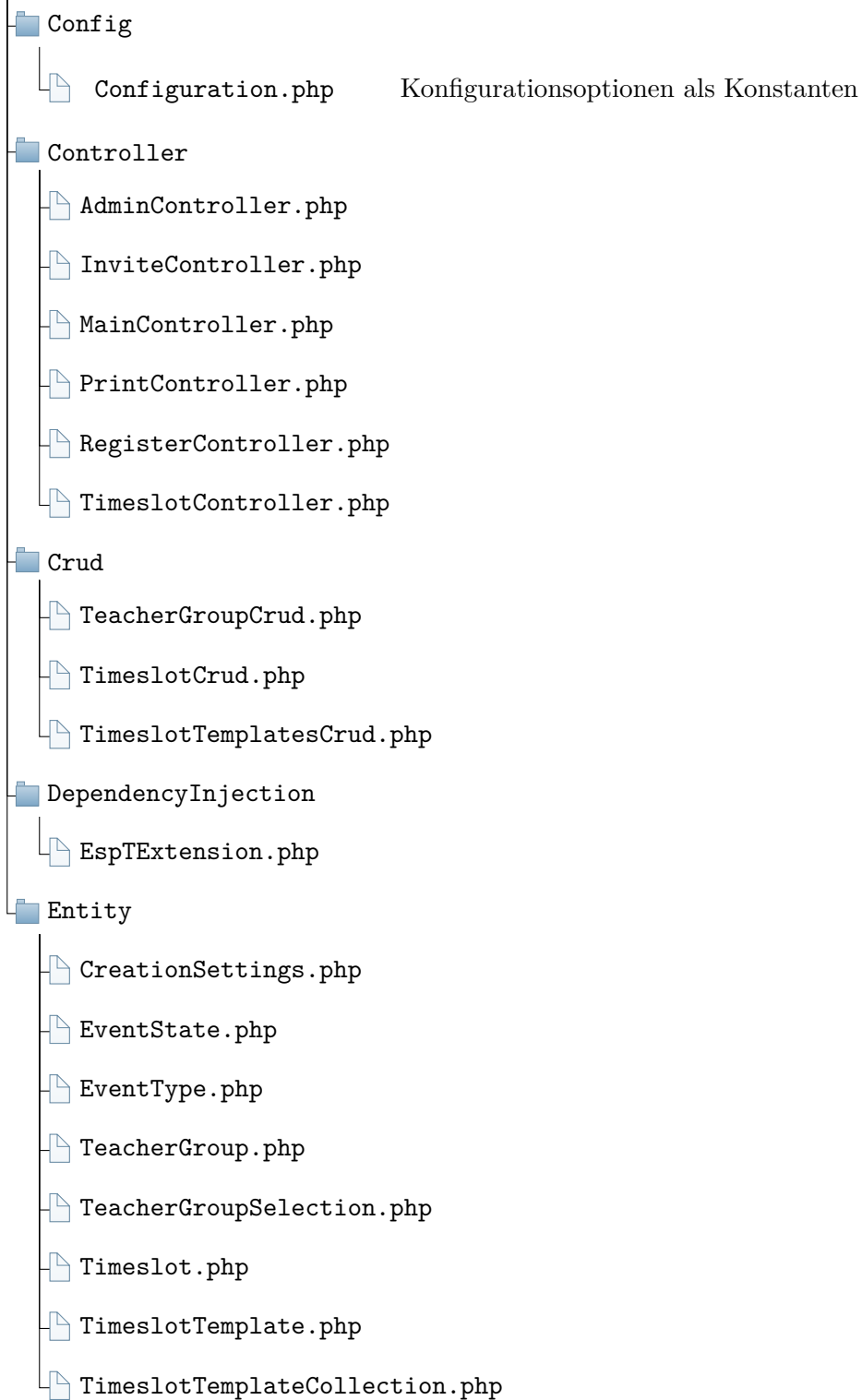


Abbildung 17: Ordnerstruktur

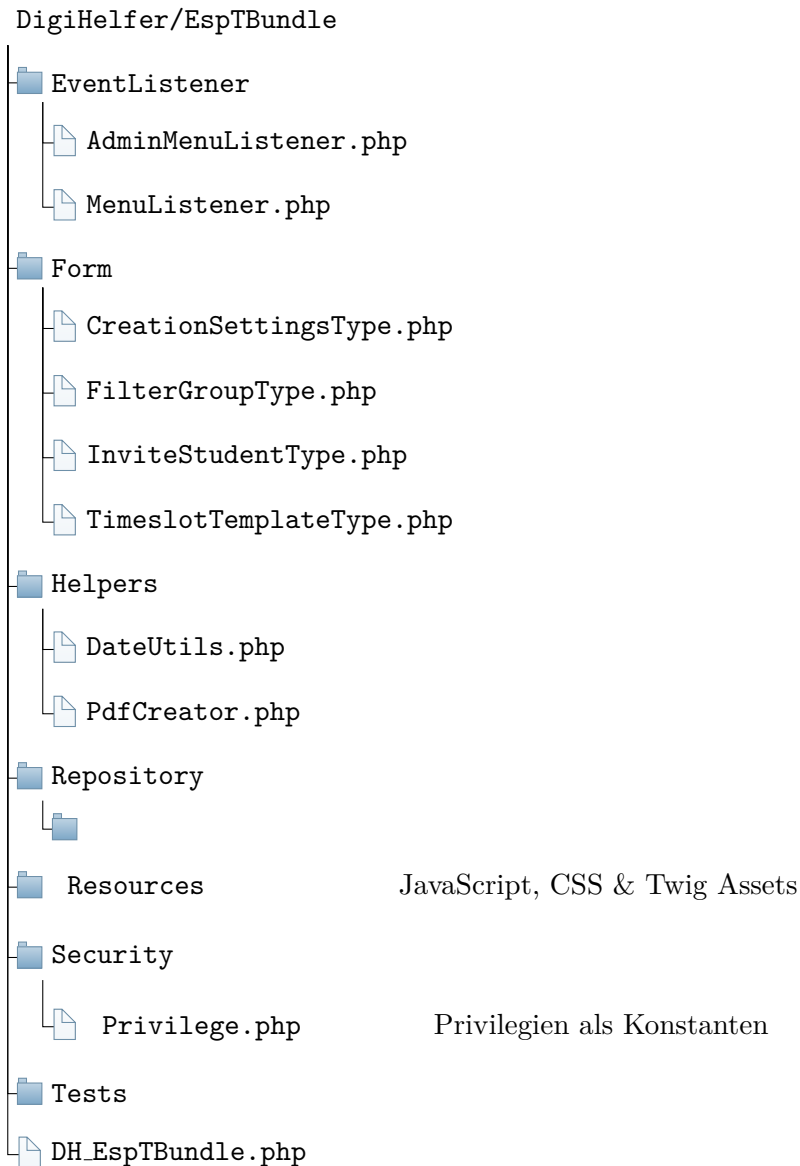


Abbildung 18: Ordnerstruktur

C.2. Ausschnitte aus dem Quellcode

```
1 <?php
2
3 declare(strict_types=1);
4
5 namespace DigiHelfer\EspTBundle\Crud;
6
7 use ...
8
9 class TeacherGroupCrud extends AdminServiceCrud {
```




```
10
11     protected static $entityClass = TeacherGroup::class;
12
13     protected function configure(): void {
14         $this->title = _('espt_groups');
15         $this->itemTitle = _('espt_group');
16         $this->templates['crud_index'] =
17             ↪ '@DH_EspT/AdminMenu.html.twig';
18     }
19
20     protected function configureListFields(ListMapper $listMapper):
21         ↪ void {
22         $listMapper
23             ->addIdentifier('id')
24             ->add('room', null, ['label' => _('espt_room')])
25             ->add('users', null, ['label' => _('espt_teachers')])
26             ->add('timeslotTemplates', null, ['label' =>
27                 ↪ _('espt_timeslot_template')]);
28     }
29
30     protected function configureShowFields(ShowMapper $showMapper):
31         ↪ void {
32         $showMapper
33             ->add('room', null, ['label' => _('espt_room')])
34             ->add('users', null, [
35                 'label' => _('espt_teachers'),
36                 'order_by' => $this->locator->get(NamesSortingDirectorInterface::class)-
37                 ↪ >getSortBy(),
38                 'entity_format' => EntityFormatter::FORMAT_USER
39             ])
40             ->add('timeslotTemplates', null, ['label' =>
41                 ↪ _('espt_timeslot_template')]);
42     }
43
44     protected function configureFormFields(FormMapper $formMapper):
45         ↪ void {
46         $formMapper
47             ->add('room', null, ['label' => _('espt_room'), 'required'
48                 ↪ => false,])
49             ->add('users', UserType::class, [
```



```
42         'label' => _('espt_teachers'),
43         'by_reference' => false,
44         'query_builder' => function(UserRepository $repository)
45             ↪ {
46                 //only show teachers in selection
47                 return $repository-
48                     ↪ >createPrivilegeQueryBuilder(Privilege::TEACHER);
49             }
50     ])
51     ->add('timeslotTemplates', EntityType::class, [
52         'class' => TimeslotTemplateCollection::class,
53         'label' => _('espt_timeslot_templates'),
54         'help' => _('espt_timeslot_templates_help'),
55         'choice_label' => 'name',
56         'multiple' => true,
57         'crud_create_remote' => $this->router()-
58             ↪ >generate('espt_admin_timeslottemplatecollection_add')
59     ]);
60
61     public function getShowActions(CrudInterface $item) :
62         ↪ OrderedCollection {
63         $actions = parent::getShowActions($item);
64         $actions['pdf'] =
65             ↪ Link::create($this->router()->generate('espt_print_group',
66             ↪ ['groupId' => $item->getId()]), _('espt_generate_print'),
67             ↪ 'list', 'btn-primary');
68         return $actions;
69     }
70
71     public function getIndexActions() : OrderedCollection {
72         $actions = parent::getIndexActions();
73         $actions['roomlist'] =
74             ↪ Link::create($this->router()->generate('espt_print_rooms'),
75             ↪ _('espt_room_list'), 'list', 'btn-primary');
76         return $actions;
77     }
78
79     public function isAuthorized(): bool {
80         return $this->isGranted(Privilege::ADMIN);
81     }
82 }
```



```
74     }
75
76     public function prepareBreadcrumbs(): array {
77         return [Breadcrumb::create(_('EspT'))];
78     }
79
80     public static function defineRoutes(): RoutingDefinition {
81         return parent::defineRoutes()
82             ->setNamePrefix('espt_admin_')
83             ->prependPathPrefix('espt/');
84     }
85
86     /**
87      * {@inheritdoc}
88      */
89     public static function getSubscribedServices(): array {
90         // Take all subscribed services from the parent classes.
91         $services = parent::getSubscribedServices();
92         // Add services we commonly use and don't want to inject in
93         // → each controller or action.
94         $services[] = NamesSortingDirectorInterface::class;
95         return $services;
96     }
97 }
```

Auffistung 7: Implementation CRUD Formulare am Beispiel Lehrergruppe

```
1 protected function configureFormFields(FormMapper $formMapper): void {
2     $formMapper
3     ->add('name', null, [
4         'label' => _('espt_name')
5     ])
6     ->add('day', null, [
7         'label' => _('Day'),
8     ])
9     ->add('timeslots', BootstrapCollectionType::class, [
10        'entry_type' => TimeslotTemplateType::class,
11        'by_reference' => false,
12        'label' => _('espt_timeslot'),
13        'allow_add' => true,
```



```
14 'allow_delete'      => true,
15 'add_button_text'  => _('espt_timeslot_add'),
16 'delete_button_text' => _('espt_timeslot_remove'),
17 'sub_widget_col'   => 9,
18 'button_col'       => 3,
19 ])
20 ->add('groups', EntityType::class, [
21 'class' => TeacherGroup::class,
22 'label' => _('espt_groups'),
23 'choice_label' => function(TeacherGroup $group) {
24 return $group;
25 },
26 'multiple' => true
27 ]
28 );
29 }
```

Aufistung 8: Implementation CRUD Zeitfenster Vorlage

```
1 /**
2  * @param CreationSettingsRepository $settingsRepository
3  * @param TeacherGroupRepository $teacherGroupRepository
4  * @return Response
5  * @throws NonUniqueResultException
6  * @Route("/rooms", name="espt_print_rooms")
7  */
8 public function printRooms(CreationSettingsRepository
9     ↪ $settingsRepository, TeacherGroupRepository
10     ↪ $teacherGroupRepository) : Response {
11
12     // column titles
13     $header = array(_('Room'), _('espt_teacher'));
14
15     $groups = $teacherGroupRepository->findAll();
16     $data = array();
17
18     /** @var TeacherGroup $group */
19     foreach ($groups as $group) {
20         $data[] = array($group->getRoom(), $group);
```



```
21 }
22
23 $pdf = $this->buildPdf($settings, 20);
24 $pdf->Ln(5);
25 $pdf->setCustomFooterText("");
26 $pdf->Table($header, $data, array(30, 130));
27
28 $filename = '/tmp/espt_temp' . $this->authenticatedUser()->getUuid() .
    ↪ '.pdf';
29 $pdf->Output($filename, 'F');
30
31 // display the file contents in the browser instead of downloading it
32 return $this->file($filename, _('EspT') .
    ↪ $settings->getStart()->format('j.n.Y') . '.pdf',
    ↪ ResponseHeaderBag::DISPOSITION_INLINE);
33 }
```

Auflistung 9: Implementation Generierung Raumliste

