



Duale Hochschule Baden-Württemberg
Mannheim

Dokumentation
**Projekt ExoPlan: Entwicklung eines
Vorlesungsverwaltungstools**

Studiengang Wirtschaftsinformatik
Studienrichtung Software Engineering

Verfassende:	Kurs WWI 17 SE B
Studiengangsleiter:	Prof. Dr. Sebastian Ritterbusch
Dozent:	Tarek Becker tarek@becker.ly
Bearbeitungszeitraum:	20.11.2019 – 31.07.2020

Inhaltsverzeichnis

Abbildungsverzeichnis	iv
Tabellenverzeichnis	v
Abkürzungsverzeichnis	vi
1 Einleitung	1
2 Projektmanagement	2
2.1 Projektdefinition	2
2.2 Projektbeteiligte	2
2.3 Zeitplanung	2
2.3.1 Meilensteinplan	2
2.3.2 Projektplanung mit Gantt	2
2.4 Teamorganisation	2
2.5 Interne Aufgabenverteilung	3
2.6 Projektabschluss	3
3 Anforderungsanalyse	4
3.1 Problemstellung und Zielsetzung	4
3.2 Ist-Analyse	5
3.3 Anforderungen	5
3.4 Priorisierung	9
4 Entwurf	11
4.1 Design Entwurf	11
4.2 Technischer Entwurf	13
4.2.1 Grundlegende Architektur	13
4.2.2 Datenmodell	13

5 Umsetzung	15
5.1 Infrastruktur	15
5.1.1 Github	16
5.1.2 VM	16
5.1.3 Docker	16
5.2 Backend	16
5.2.1 Grundlegender Aufbau	16
5.2.2 API-Schnittstellen	18
5.2.3 Sicherheit und Zugriffskontrolle	20
5.3 Frontend	22
5.3.1 Implementierung mit React	22
5.3.2 Styleguide für die Umsetzung	24
5.3.3 Login	26
5.3.4 Kursübersicht	26
5.3.5 Dozentenansicht	27
5.3.6 Modulkatalog	28
5.3.7 Datenverwaltung	29
5.3.8 Administrationsbereich	30
5.3.9 Allgemeine Einstellungen	31
5.3.10 Vorlesungskalender mit Google Calendar	32
5.4 Verknüpfung Backend und Frontend	37
5.5 Zugriffsberechtigungen	37
5.5.1 Zugriffsberechtigungen	37
5.6 Test	37
6 User Guide	38
6.1 Setup	38
6.2 Anwendungsanleitung	38
6.3 Funktionalitäten	38
6.4 Erweiterbarkeit	38
7 Evaluation	39
7.1 Bewertung Umsetzung	39
7.2 Lessons Learned	39

7.3 Nächste Schritte	39
8 Fazit und Ausblick	40
Quellenverzeichnis	41
A ER-Modell	42
B Styleguide	44

Abbildungsverzeichnis

Abbildung 4.1	Aufbau der Benutzeroberfläche	12
Abbildung 5.1	Übersicht der umgesetzten Infrastruktur	15
Abbildung 5.2	Schichtenmodell des Back-Ends	17
Abbildung 5.3	Übersicht der Ordnerstruktur des Front-Ends	23
Abbildung 5.4	Umsetzung der Benutzeroberfläche	24
Abbildung 5.5	Auszug Styleguide für das Logo	25
Abbildung 5.6	Auszug Styleguide für das Farbdesign	25
Abbildung 5.7	Login	26
Abbildung 5.8	Kursübersicht mit Vorlesungskalender	27
Abbildung 5.9	Anzeige der Dozenten	28
Abbildung 5.10	Anzeige der Modulkatalogen	29
Abbildung 5.11	Datenverwaltung	30
Abbildung 5.12	Administrationsbereich	31
Abbildung 5.13	Allgemeine Kontoeinstellungen	32
Abbildung 5.14	Übersicht des React Schedulers	33
Abbildung 5.15	Erstellen eines Ereignisses im React Scheduler	35
Abbildung 5.16	Popup eines Ereignisses im React Scheduler	35

Tabellenverzeichnis

Tabelle 3.1 Anforderungen des Auftraggebers	9
Tabelle 3.2 Priorisierung der Anforderungen	10
Tabelle 5.1 Übersicht Routen	19
Tabelle 5.2 Übersicht Admin-Routen	20

Abkürzungsverzeichnis

API	Application Programming Interface
CSS	Cascading Style Sheets
DHBW	Duale Hochschule Baden-Württemberg
DOM	Document Object Model
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ORM	Objektrelationales Mapping
REST	Representational State Transfer
SPA	Single-Page-Application

1 Einleitung

2 Projektmanagement

2.1 Projektdefinition

2.2 Projektbeteiligte

2.3 Zeitplanung

2.3.1 Meilensteinplan

2.3.2 Projektplanung mit Gantt

2.4 Teamorganisation

Zunächst wurde als Projektleiter Kay Wessel festgelegt, der die gesamte Verantwortung für das Projekt trägt und deshalb die oberste Entscheidungsgewalt inne hat. Zu seinen Aufgaben gehört die Gesamtplanung des Projekts und das Managen von organisatorischen Angelegenheiten, wie Aufsetzen und Durchführung von Meetings sowie das Klären von inhaltlichen Fragen.

Für das Projekt konnten die drei Aufgabenbereichen Produktmanagement, Front-End und Back-End identifiziert werden. Anhand dieser Aufgabenbereiche wurde der Kurs in drei dedizierte Teams unterteilt, welche jeweils der Führung eines Teamleiters unterliegen. Für das Produktmanagement ist Kay Wessel zuständig, Sandra Keller für das Front-End und Martin Sandig für das Back-End. Die Aufgabe der Teamleiter ist die Verteilung und Kontrolle von Aufgaben unter den jeweiligen Mitgliedern eines Teams. Das Planen und Erfassen der Aufwände fällt ebenfalls in ihren Tätigkeitsbereich.

Die Organisation und Realisierung des Projektes erfolgte mittels der Nutzung unterschiedlicher digitaler Medien. Als Unterstützung agiler Softwareentwicklung wurden

GitHub sowie Trello verwendet. Der netzbasierter Dienst *GitHub*¹ ermöglicht gemeinsames Arbeiten und die Versionsverwaltung bei Software-Entwicklungsprojekten. Für die Entwicklung des Front-End und Back-End sowie für die Erstellung der Dokumentation werden eigene Verzeichnisse (Repositories) verwendet.

Der Aufgaben-Verwaltungs-Onlinedienst *Trello*² ermöglicht das Verwalten von Aufgaben in sogenannten Boards. Die Aufgaben können beliebig bearbeitet und mit Checklisten, Anhängen, Terminen und vielem mehr versehen werden. Außerdem lassen sich die Bearbeiter zu einer Aufgabe hinzufügen, sodass diese über alle Änderungen und Fortschritte informiert werden. Für jedes Team wurde in Trello ein eigenständiges Board erstellt, sodass die Aufgaben im Verbund eines Teams entsprechend zugeordnet und anschließend bearbeitet werden können. Zur Klärung allgemeiner Fragen wurde ein gemeinsames Board „Organisation“ eingerichtet. Da für das Projekt eine Protokollierung der Stunden erforderlich ist, wurde ein geteiltes *Google-Sheet*³ erstellt, welches das aktuelle Stundenkontingent der Teammitglieder beinhaltet. Dieses und die Aufgabenverwaltung mit Trello gewährleisten die Transparenz über die aktuellen Aufwände und Aufgaben.

Die Kommunikation und Koordination erfolgte über den Instant-Messaging-Dienst *WhatsApp*⁴ sowie das Konferenzsystem *Discord*.⁵ In Letzterem sind verschiedene Sprachkanäle eingerichtet, in denen telefonische Absprachen durchgeführt werden können. Dabei besteht die Möglichkeit den Bildschirm und damit Inhalte zu teilen, welches unter anderem für das gemeinsame Bearbeiten von Aufgaben genutzt wurde.

2.5 Interne Aufgabenverteilung

2.6 Projektabschluss

¹<https://github.com/>

²<https://trello.com/>

³<https://www.google.com/sheets/about/>

⁴<https://www.whatsapp.com/>

⁵<https://discord.com/>

3 Anforderungsanalyse

In diesem Kapitel wird die Anforderungsanalyse erstellt. Zunächst wird die zu lösende Problemstellung sowie die Zielsetzung des Projekts beschrieben. Anschließend werden die Anforderungen des Auftraggebers identifiziert. Alle Anforderungen werden danach anhand ihrer Relevanz und dem gegebenen zeitlichen Budget priorisiert.

3.1 Problemstellung und Zielsetzung

Die Vorlesungspläne für die Kurse des Studiengangs Wirtschaftsinformatik an der Duale Hochschule Baden-Württemberg (DHBW) müssen aktuell vollständig manuell durch die jeweiligen Tutoren erstellt werden. Eine Excel-Vorlage dient dabei als Unterstützung, indem darin Informationen über den Kurs und die Module gesammelt werden. Für die zu haltenden Vorlesungen müssen passende Dozenten gesucht und angefragt werden, wobei keine zentrale Liste der Dozenten existiert, was den Prozess wiederum erschwert. Wurden zu allen Vorlesungen passende Dozenten sowie Zeitfester gefunden, werden diese in einem Google Calendar integriert.

Mit Hilfe eines Tools soll künftig das eben beschriebene Vorgehen zur Erstellung der Vorlesungspläne mit seinen manuellen Teilprozessen für die Studiengangsleiter erleichtert werden. Ziel dabei ist es, eine zentrale Dozentenverwaltung zu realisieren sowie den Studiengangsleitern ein einheitliches System zur Planung und Koordination ihrer Kurse zur Verfügung zu stellen.

3.2 Ist-Analyse

3.3 Anforderungen

Bevor der Lösungsansatz für die Problemstellung aus Kapitel 3.1 auf der vorherigen Seite konzipiert wird, müssen die Wünsche und Vorstellungen des Auftraggebers ermittelt werden. Bei dem Auftraggeber handelt es sich um einen Studiengangsleiter des Studiengangs Wirtschaftsinformatik an der DHBW. Durch verschiedene Interviews wurden die Anforderungen des Auftraggebers festgestellt. Die Anforderungen sind in Tabelle 3.1 aufgelistet.

Anforderung	Kurztitel	Beschreibung
A1	Dozentenpool	Alle Dozenten sollen in einem zentralen Dozentenpool verwaltet werden können.
A2	Stundenplan	Die Software soll Stundenpläne verwalten und erzeugen anhand diverser Parameter, die eingetragen werden wie z. B. welche Vorlesungen in welchem Semester für einen Kurs und Dozenten stattfinden.
A3	Google Calendar	Google Calendar soll vollständig in die Software integriert werden.
A4	Profilzuordnung	Die Profile der User, mit denen sie sich am System anmelden, sollen über deren Mail-Adresse zugeordnet werden.
<i>Fortsetzung nächste Seite</i>		

Anforderung	Kurztitel	Beschreibung
A5	Modulkataloge	Die Modulkataloge mit den Vorlesungen, Stundenanzahlen und Prüfungsmöglichkeiten (Klausur, Referat oder andere Ausarbeitungen) sollen verwaltet und über Templates hinzugefügt werden können.
A6	Kurskoordination	Ein Studiengangsleiter soll eine variable Anzahl an Kursen im System koordinieren können.
A7	Eindeutigkeit der Module	Module müssen eindeutig identifizierbar sein, da z. B. das Statistikmodul für Wirtschaftsinformatiker nicht gleich dem Statistikmodul für angewandte Informatiker ist.
A8	Suchen und Filtern im Dozentenpool	In dem Dozentenpool sollen Dozenten per Suche gefunden sowie nach sinnvollen Kriterien gefiltert werden können.
A9	Hinzufügen und Löschen im Dozentenpool	In dem Dozentenpool sollen neue Dozenten hinzugefügt und bestehende Dozenten gelöscht werden können.
<i>Fortsetzung nächste Seite</i>		

Anforderung	Kurztitel	Beschreibung
A10	Dozenteninformationen im Dozentenpool	Zu jedem Dozenten im Dozentenpool sollen Informationen wie Name, Mail, Handynummer verfügbar sein. Zudem sollen Felder für eine Bewertung, eine Information, ob der Dozent hauptamtlich tätig ist und der/-die jeweilige(n) Schwerpunkt(e) sowie ein Freitextfeld für Kommentare vorhanden sein.
A11	Automatisierte Dozentenanfrage	Nach Auswahl eines Dozenten soll diesem automatisch eine Anfrage via Mail geschickt werden, wobei ein personalisierbares Template als Basis dient.
A12	Zuordnung der Dozenten	Die Dozenten sollen jeweils einem Studiengangsleiter zugeordnet werden.
A13	Warnung bei maximaler Studenanzahl	Das Tool soll benachrichtigen, wenn ein Dozent seine maximale Stundenanzahl überschreiten würde bei der aktuellen Planung.
A14	Paralleler Zugriff	Parallele Zugriffe auf einen Kurs durch jeweils berechtigte Personen sollen möglich sein.
A15	Tooladministration	Die Administration des Tools soll über die Studiengangsleiter erfolgen.
A16	Kursübersicht	Es soll pro Kurs eine Übersicht mit Anzahl und Art der jeweiligen Prüfungsleistungen des Kurses existieren.
<i>Fortsetzung nächste Seite</i>		

Anforderung	Kurztitel	Beschreibung
A17	Farbiger Planungsstand	Der aktuelle Stand der Semesterplanung eines Kurses soll mithilfe von Farben verdeutlicht werden, z. B. grün – Termin der Vorlesung ist fix; gelb – Dozent hat zugesagt, aber noch kein fixer Termin; etc.
A18	Planungsexport	Existierende Planungen sollen exportiert werden können, z. B. für nachfolgende Kurse.
A19	Informationen der Veranstaltungen	Die detaillierten Informationen über die Veranstaltungen eines geplanten Kurses sollen mithilfe von angehängten Dokumenten einsehbar sein.
A20	Veranstaltungstitel	Der Titel einer Veranstaltung soll in folgender Form dargestellt werden: „Name der Veranstaltung - Name des Dozenten“.
A21	Automatisierte Benachrichtigungen	Kurse und Dozenten werden mit einer automatisch generierten E-Mail über Semesterbeginn, die finalisierte Planung oder die Festlegung von Prüfungsterminen und -arten benachrichtigt.
A22	Klausurenmaximum	Das Tool soll berücksichtigen, dass Studiengänge ab 2018 nur sechs schriftliche Klausuren pro Semester erlauben, indem eine entsprechende Warnung ausgegeben wird bei einer Überschreitung.
<i>Fortsetzung nächste Seite</i>		

Anforderung	Kurztitel	Beschreibung
A23	Korrekte Modul-durchführung	Die Anwendung soll berücksichtigen, dass Lehrveranstaltungen eines Moduls innerhalb eines Studienjahrs erfolgen müssen. Innerhalb eines Studienjahres sollen die Lehrveranstaltungen beliebig verschoben werden können.
A24	Eintragung von Wahlmodulen	Wahlmodule sollen in den Modulkatalog eingetragen werden können.
A25	Dozentenvorschläge	Bei neu hinzugefügten Modulen sollen Vorschläge für den Dozenten generiert werden.
A26	Weiterführbarkeit	Die Software soll später von nachkommenden Jahrgängen weitergeführt und modifiziert werden können.
A27	Usability	Die Bedienbarkeit der Software soll so intuitiv und einfach wie möglich gestaltet werden.

Tabelle 3.1: Anforderungen des Auftraggebers

Quelle: Interview mit Professor Matt

3.4 Priorisierung

Die Anforderungen werden, um sie zu priorisieren, in zwei Kategorien eingeteilt. Als *Muss* wird eine Anforderung kategorisiert, wenn sie die höchste Priorität hat. Eine Nicht-Erfüllung einer Anforderung, die mit einem *Muss*-Kriterium versehen wurde, kann die Ablehnung des gesamten Projektes oder Produktes bedeuten. Die *Kann*-Anforderungen sind Abstufungen der *Muss*-Anforderungen und „können“ erfüllt werden, soweit die Ressourcen dies ermöglichen. Die Anforderungen mit ihren jeweiligen

Priorisierungen sind in Tabelle 3.2 dargestellt.

Anforderung	Kurztitel	Priorität
A1	Dozentenpool	<i>Muss</i>
A2	Stundenplan	<i>Muss</i>
A3	Google Calendar	<i>Muss</i>
A4	Profilzuordnung	<i>Muss</i>
A5	Modulkataloge	<i>Muss</i>
A6	Kurskoordination	<i>Muss</i>
A7	Eindeutigkeit der Module	<i>Muss</i>
A8	Suchen und Filtern im Dozentenpool	<i>Muss</i>
A9	Hinzufügen und Löschen im Dozentenpool	<i>Muss</i>
A10	Dozenteninformationen im Dozentenpool	<i>Muss</i>
A11	Automatisierte Dozentenanfrage	<i>Kann</i>
A12	Zuordnung der Dozenten	<i>Kann</i>
A13	Warnung bei maximaler Stundenanzahl	<i>Kann</i>
A14	Paralleler Zugriff	<i>Kann</i>
A15	Tooladministration	<i>Kann</i>
A16	Kursübersicht	<i>Kann</i>
A17	Farbiger Planungsstand	<i>Kann</i>
A18	Planungsexport	<i>Kann</i>
A19	Informationen der Veranstaltungen	<i>Kann</i>
A20	Veranstaltungstitel	<i>Kann</i>
A21	Automatisierte Benachrichtigungen	<i>Kann</i>
A22	Klausurenmaximum	<i>Muss</i>
A23	Korrekte Moduldurchführung	<i>Muss</i>
A24	Eintragung von Wahlmodulen	<i>Muss</i>
A25	Dozentenvorschläge	<i>Kann</i>
A26	Weiterführbarkeit	<i>Kann</i>
A27	Usability	<i>Muss</i>

Tabelle 3.2: Priorisierung der Anforderungen

4 Entwurf

4.1 Design Entwurf

Für das Design sollte ein Prototyp erstellt werden, der den späteren Aufbau der Applikation verdeutlicht. Als technisches Hilfsmittel wurde das cloud-basierte Tool *Figma*⁶ verwendet. Dieses bietet die Möglichkeit mit mehreren Personen in Echtzeit an einem Prototypen zu arbeiten. Dabei ist es in seiner Bedienung leicht verständlich und bietet alle wichtigen Design-Werkzeuge. Außerdem können die Elemente der Benutzeroberfläche miteinander verbunden und mit Interaktionen sowie Animationen versehen werden, sodass ein interaktiver Prototyp entsteht. Nach dem Starten eines Prototyps kann das Design der Oberfläche sowie die Navigationen geprüft werden. Über einen generierten Link lässt sich der Prototyp mit Projektbeteiligten teilen.

Für das Erstellen des Designs wurde eine iterative Vorgehensweise gewählt. Gestartet ist der Prozess mit dem Einholen von Anforderungen, siehe Kapitel 3. Basierend darauf konnte die allgemeine Struktur der Benutzeroberfläche erstellt werden. Anschließend verlief der Prozess in vielen Iterationsschleifen, in denen bekannte Anforderungen zunächst statisch umgesetzt und bei Rücksprachen mit den Stakeholdern deren Feedback eingeholt wurde. Dabei wurden ebenfalls weitere Anforderungen besprochen, sodass das Design nachfolgend erweitert werden konnte. Während den Schleifen wurde der Prototyp jeweils interaktiv erstellt, d.h. mit Animationen versehen, und es wurden weitere Details hinzugefügt. Auf diese Weise sind während dem Projekt verschiedene Versionen des Prototyps entstanden, die jeweils mit den Stakeholdern besprochen wurden. Neben diesen Absprachen erfolgten zusätzliche mit dem Back-End-Team. Thema dieser Besprechungen waren die Umsetzbarkeit der Funktionalitäten, die vom Back-End bereitgestellt werden müssen. Damit aber auch das gesamte Projektteam über den Entwurf und die Fortschritte informiert blieb, wurden das jeweils aktuelle Design in verschiedenen wöchentlichen Meetings des gesamten Teams präsentiert. Damit je-

⁶<https://www.figma.com/>

des Team-Mitglied jederzeit auf den Prototyp zugreifen kann, wurde der Link in Trello eingetragen.

Die designte Benutzeroberfläche besteht, wie in Abbildung 4.1 dargestellt, jeweils aus drei Elementen. Sie beinhaltet eine Kopfzeile, eine Navigationsleiste und die gewählten Inhalte. Die Kopfzeile stellt am linken Rand das Logo der DHBW und am rechten Rand den Benutzernamen dar. Durch das Anklicken des Benutzernamens können allgemeine Kontoeinstellungen getätigter oder das Abmelden von der Applikation ausgelöst werden. Auf jeder Seite der Benutzeroberfläche befindet sich außerdem am linken Rand eine vertikale Navigationsleiste. Diese beinhaltet die drei Hauptseiten und ermöglicht das Navigieren zwischen diesen. Es können die Vorlesungen und Vorlesungspläne je Kurs angeschaut werden, eine Übersicht der Dozenten und verschiedene Modulkataloge.

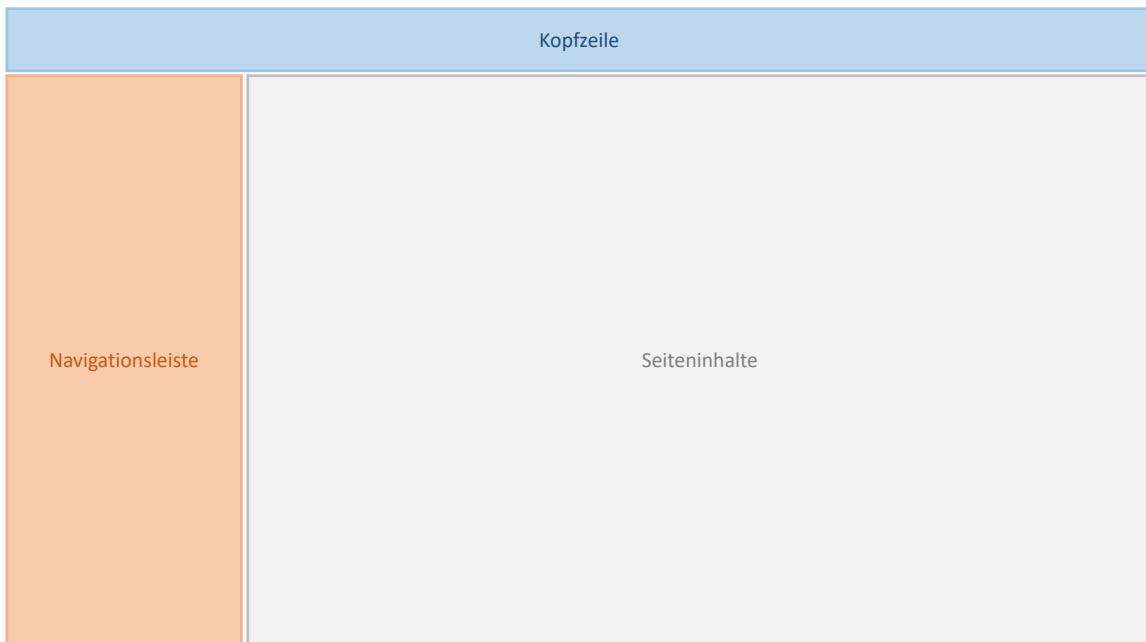


Abbildung 4.1: Aufbau der Benutzeroberfläche

Link finale Version Mockups

4.2 Technischer Entwurf

4.2.1 Grundlegende Architektur

4.2.2 Datenmodell

Für das serverseitige Speichern von Daten sind relationale Datenbanken weit verbreitet. Deshalb wird auch im Rahmen dieses Projekts eine solche Form der Datenhaltung genutzt. Im Folgenden wird nun das Datenmodell näher erläutert, welches im Anhang A dargestellt ist.

Über `directorOfStudies` werden die Nutzer der Anwendung, die Studiengangsleiter, dargestellt. Sie sind nicht automatisch als Dozenten hinterlegt, können jedoch als solche hinzugefügt werden. Ein Studiengangsleiter kann zusätzlich ein Administrator der Anwendung sein, wodurch er weitere Berechtigungen erhält, wie das Zurücksetzen von Passwörtern.

Die Entität `course` repräsentiert einen Kurs der DHBW, welcher genau einem Studiengang (`fieldOfStudy`) und einer Studienrichtung (`majorSubject`) zugeordnet ist. Damit die Eindeutigkeit der Module (Anforderung A7) gewährleistet ist, referenziert `majorSubject` auf die dazugehörigen Modulgruppen und entspricht dadurch dem gegebenen Aufbau des Modulkatalogs. Aus diesem Grund beinhaltet es auch das Jahr, ab welchem der Modulkatalog gültig ist.

Damit auch Wahlmodule abgebildet werden können (Anforderung A24), sind die Module (`module`) Teil einer Modulgruppe (`moduleGroup`). Besteht die Wahlmöglichkeit zwischen mehreren Modulen, beinhaltet die Modulgruppe alle wählbaren Module. Andernfalls beinhaltet sie nur das vorgesehene Modul. Damit neben den Informationen des aktuellen Modulkatalogs der DHBW auch die Prüfungsleistung je Modul gespeichert werden kann, wird die Entität `academicRecord` verwendet.

Kurse beinhalten neben ihrem Namen auch eine Referenz auf den für sie angelegten Google Calendar. Mithilfe von `semester` werden die kursspezifischen Semester festgelegt, welche den zeitlichen Rahmen des Studiums beschreiben. Die Kurse unterliegen

üblicherweise einem Studiengangsleiter, jedoch ist es auch möglich weitere Studiengangsleiter hinzuzufügen, bspw. für Übergangszeiten bei Positionswechseln.

Die Dozenten (**lecturer**) werden durch zahlreiche Attribute beschrieben (Anforderung A10) sowie ein Lebenslauf kann als PDF gesondert gespeichert werden. Gemäß Anforderung A1 findet die Verwaltung der Dozenten in einem zentralen Dozentenpool statt. Dabei ist ihnen jeweils genau ein Studiengangsleiter als Ersteller zugeordnet (Anforderung A12) und die Bearbeitbarkeit durch andere Studiengangsleiter kann eingeschränkt werden. Für die Verknüpfung zwischen Dozierenden und Vorlesung wird ein thematischer Schwerpunkt (**mainFocus**) eingesetzt. Dadurch wird es ermöglicht passende Dozenten für Vorlesungen vorzuschlagen (Anforderung A25).

Eine **lecture** ist eine abstrakte Lehr-/Lerneinheit. Sie beinhaltet lediglich allgemeine Beschreibungen und ist somit für alle Kurse einer Studienrichtung anwendbar. Planungseinheiten für konkrete Veranstaltungen werden über die Entität **presentations** abgebildet. Sie referenzieren auf abstrakte Vorlesungen, eine Prüfungsleistung sowie ein Semester und einen Kurs. Durch Planungseinheiten können gleiche Veranstaltungen mehrfach erzeugt werden, um mehrere Dozenten gleichzeitig zu kontaktieren und anzuhören. Der Status einer **presentation** indiziert den Fortschritt der Planung und die Rückmeldungen der jeweiligen Dozenten.

5 Umsetzung

5.1 Infrastruktur

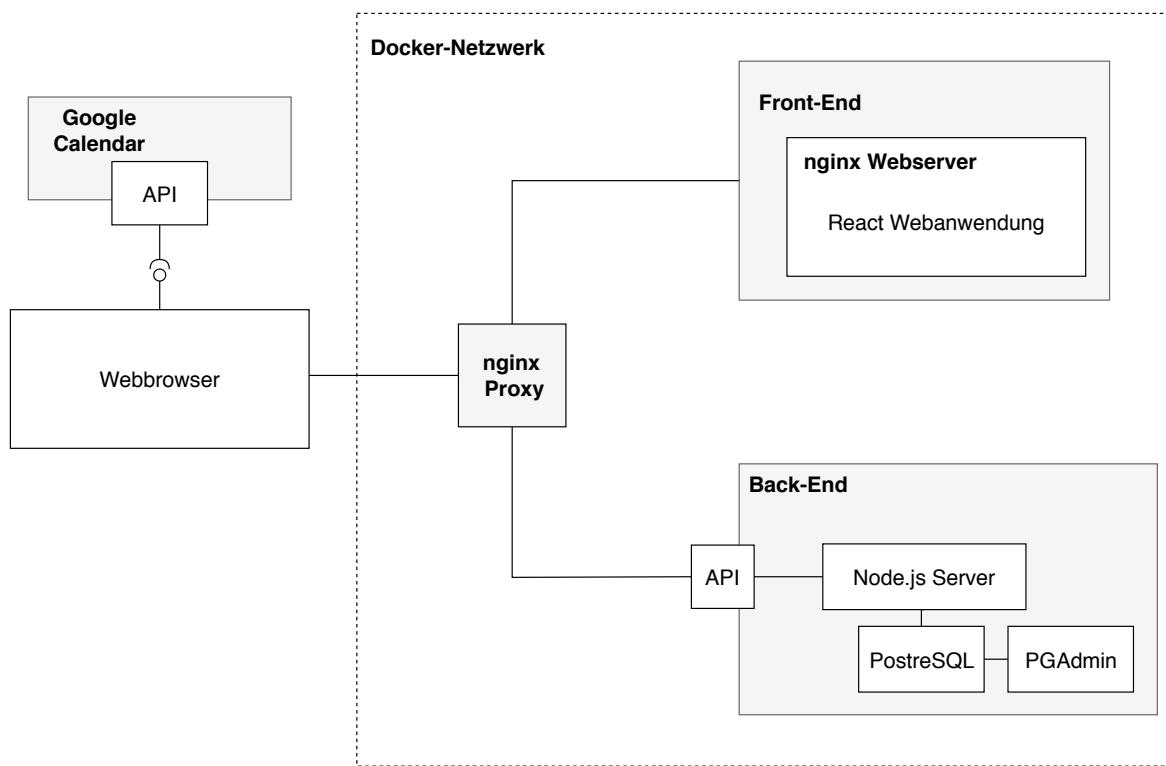


Abbildung 5.1: Übersicht der umgesetzten Infrastruktur

5.1.1 Github

5.1.2 VM

5.1.3 Docker

5.2 Backend

5.2.1 Grundlegender Aufbau

Server

Zur Implementierung der Geschäftslogik wird ein Webserver mit *Node.js*⁷ in Verbindung mit dem Framework *Express*⁸ verwendet. Das Framework Express dient der Entwicklung von Webanwendungen und Application Programming Interfaces (APIs). Diese Technologien sind weit verbreitet und vielen Teammitgliedern bereits bekannt. Besonders letzteres spricht für die Nutzung, da keine Einarbeitungsphase notwendig ist und direkt mit der Entwicklung gestartet werden kann.

Datenbank

Das Anbieten der API-Endpunkte und den dazugehörigen Funktionen erfordert das serverseitige Speichern von Daten, wie die Studiengangsleiter und Dozenten. Für die Datenhaltung wurde die Datenbank *PostgreSQL*⁹ ausgewählt, da diese relational aufgebaut ist und den Projektmitgliedern durch vorige Vorlesungen ebenfalls bereits bekannt ist. Außerdem ist sie ziemlich verbreitet und wird von verschiedenen Frameworks unterstützt. Ein Beispiel dafür ist *Sequelize*,¹⁰ ein Objektrelationales Mapping (ORM), welches in diesem Projekt für die Anbindung der Datenbank an das Back-End verwendet wird.

⁷<https://nodejs.org/>

⁸<https://expressjs.com/>

⁹<https://www.postgresql.org/>

¹⁰<https://sequelize.org/master/>

Architektur

Die Architektur des Back-Ends lässt sich durch das in Abbildung 5.2 dargestellte Schichtenmodell beschreiben. Wird eine Anfrage an das Back-End gestellt, erfolgt zunächst die Authentifizierung des Benutzers. Dazu wird geprüft, ob dieser bereits angelegt wurde und ob der mitgegebene Token dem hinterlegten Token entspricht. Dadurch kann entschieden werden, ob die angefragte Route benutzt werden darf.

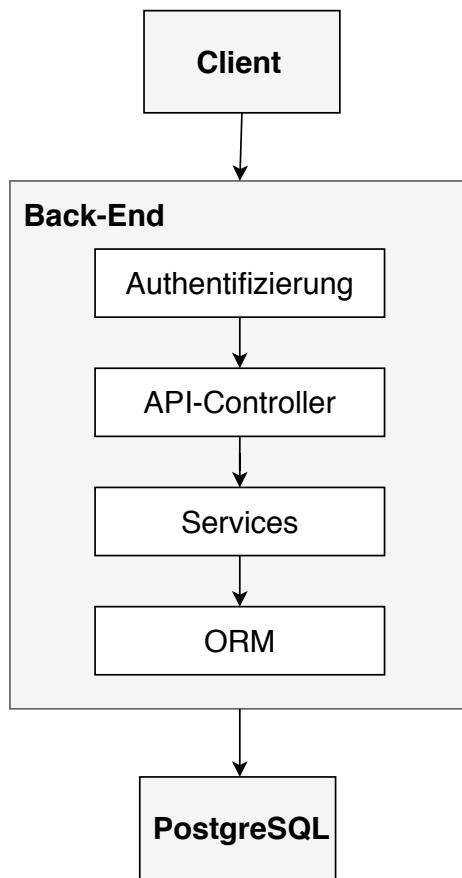


Abbildung 5.2: Schichtenmodell des Back-Ends

Ist dies der Fall wird die Anfrage an die nächste Schicht weitergegeben. Durch die API-Controller werden die Routen abgebildet. Für jede Route gibt es einen eigenen Controller, welcher die jeweiligen HTTP-Methoden einer Route anbietet. Außerdem enthält diese Schicht eine Steuerlogik, welche prüft, ob die gewünschte Aktion vom Benutzer durchgeführt werden darf.

Die Serviceschicht stellt eine Abstraktion von dem Inhalt der Datenbank bereit. Sie definiert außerdem die ORM-Aufrufe und stellt dadurch eine Verbindung zu dem Mapper bereit.

Mithilfe des ORM werden die objektorientierten Konzepte auf entsprechende relationale Konzepte abgebildet. Auf diese Weise können Entwickler in der objektorientierten Programmiersprache bleiben, um auf die Datenbank zuzugreifen. Dieser Zugriff erfolgt im nächsten Schritt, wenn die Daten entsprechend der gestellten Anfrage ausgelesen und an den Benutzer zurückgegeben werden.

5.2.2 API-Schnittstellen

Implementierung der Routen

Die Routen sind an dem Representational State Transfer (REST)-Paradigma orientiert, wonach nur die benötigten Informationen zurückgegeben werden. Bei den Modulgruppen wurde jedoch davon abgewichen, da ein semantischer Zusammenhang zwischen den einzelnen Modulgruppen besteht. Bei einem GET-Request werden immer alle Modulgruppen der angegebenen Studienrichtung zurückgegeben. Dies vereinfacht die Programmierung im Front-End, da mit einer Anfrage an das Back-End alle Informationen zum Anzeigen eines bestimmten Modulkatalogs zurückgegeben werden.

Durch die Routen wird der Zugriff auf die vorhandenen REST-Ressourcen gewährt, was unter anderem das Erstellen, Lesen, Bearbeiten und Löschen ermöglicht (CRUD-Operationen). Für den Zugriff auf die Routen wird das HTTP-Protokoll verwendet. Dadurch können entsprechende Statuscodes von HTTP verwendet werden, um eine Rückmeldung zum Erfolg oder Misserfolg einer Anfrage zu geben. Liegt eine fehlerhafte Anfrage vor, wird eine spezifische Fehlermeldung mit Erklärung in einem 4xx-Code an den Client gesendet. Der HTTP-Code 500 indiziert einen Fehler, welcher möglicherweise im Back-End liegt.

Übersicht der Routen

Die Tabellen 5.1 und 5.2 geben eine Übersicht der zur Verfügung stehenden Routen sowie die entsprechenden HTTP-Methoden, mit welchen auf die Routen zugegriffen werden kann. Tabelle 5.1 beschreibt dabei die Routen und ihre Aktionen, die von allen Nutzern der Anwendung verwendet werden können. Dazu zählt beispielsweise das nach Anforderung A9 geforderte Anlegen und Löschen von Dozenten, welches mithilfe der Route `\lecturers` durchgeführt wird.

Route	GET	POST	PUT	DELETE
<code>\register</code>		x		
<code>\login</code>		x		
<code>\changePassword</code>			x	
<code>\academicRecords</code>	x	x	x	x
<code>\courses</code>	x	x	x	x
<code>\directorOfStudies</code>	x		x	
<code>\fieldsOfStudy</code>	x	x	x	x
<code>\googleCalendarAPI</code>	x			
<code>\lecturerCV</code>	x		x	x
<code>\lecturers</code>	x	x	x	x
<code>\mainFocuses</code>	x	x	x	x
<code>\majorSubjects</code>	x		x	x
<code>\modulecatalog</code>	x			
<code>\moduleGroups</code>		x	x	x
<code>\presentations</code>	x	x	x	x
<code>\semesters</code>		x	x	x
<code>\transferOwnership</code>		x		

Tabelle 5.1: Übersicht Routen

In Tabelle 5.2 werden hingegen die Routen und Aktionen veranschaulicht, welche nur von Administratoren verwendet werden können. Diese umfassen unter anderem das Anlegen und Auslesen von Nutzern. Darüber hinaus kann ein Google Calendar aktualisiert sowie ein Nutzer zu einem Admin ernannt werden.

Admin-Route	GET	POST	PUT	DELETE
\createUser		x		
\googleCalendarAPI			x	
\registerKey	x		x	
\resetPassword			x	
\upgradeToAdmin			x	
\users	x			

Tabelle 5.2: Übersicht Admin-Routen

Während der Entwicklung wurden zunächst weitere Routen erstellt. Die Entscheidungen im Laufe des Projekts haben jedoch dazu geführt, dass diese nicht mehr notwendig sind und deshalb nicht weiter entwickelt wurden. Ein Beispiel ist die Route \signup. Diese wird nicht mehr angeboten, da eine selbstständige Registrierung nicht mehr ohne Weiteres möglich sein soll.

5.2.3 Sicherheit und Zugriffskontrolle

Da die Anwendung nur von bestimmten Personen verwendet werden soll, wird die Registrierung eingeschränkt. Dies erfolgt über einen Registrierungsschlüssel, welcher bei einer Registrierung angegeben werden muss. Ein solcher Schlüssel ist dem Administrator bekannt und dieser kann einen neuen Registrierungsschlüssel festlegen. Zusätzlich besteht die Möglichkeit, dass der Administrator selbst Benutzer zum System hinzufügt.

Des Weiteren sollen in der Anwendung nur benutzerspezifische Inhalte angezeigt werden und der Benutzer darf nur die Aktionen durchführen, zu denen er berechtigt ist. Dafür muss ein Nutzer identifiziert werden können, weshalb eine digitale Identität erstellt wird. Dies erfolgt durch die Registrierung mit Mail-Adresse und Passwort, entsprechend der Anforderung A4.

Für die Verifizierung des Passworts bei der Anmeldung eines Nutzers, muss das Passwort gespeichert werden. Damit dieses nicht ausgelesen und für unberechtigten Zugang zu der Software genutzt werden kann, wird das Passwort als Hash-Wert gespeichert.

Die Erstellung des Hash-Werts erfolgt mithilfe der Node.js-Bibliothek *bcrypt*.¹¹ Diese baut auf der kryptographische Hashfunktion bcrypt auf, welche speziell für das Hashen und Speichern von Passwörtern entwickelt wurde.

Um jedoch eine häufige Verwendung des Passworts zu vermeiden wird ein *JSON Web Token*¹² generiert, welcher ein genormtes Access-Token darstellt. Bei der Registrierung und Anmeldung eines Nutzers wird ein solcher Token ausgestellt, der bei anschließenden Anfragen an das Back-End mitgegeben wird. Dieses prüft daraufhin die Korrektheit des Tokens bevor eine Aktion durchgeführt wird. Ist der Nutzer gespeichert und der Token hinterlegt, ist dieser berechtigt Anfragen zu senden. Außerdem wird bei jedem Aufruf der Controllern geprüft, ob eine Aktion valide ist. Auf diese Weise wird bspw. verhindert, dass ein Studiengangsleiter die Kurse eines anderen ändern kann.

Ähnlich wie Passwörter nicht im Klartext gespeichert werden sollten, sollte auch die Kommunikation zwischen Client und Server nicht im Klartext erfolgen. Deshalb wird das sichere Kommunikationsprotokoll HTTPS verwendet. Dadurch werden die Nachrichten verschlüsselt übertragen, was insbesondere für die Übermittlung von Passwörtern und Token wichtig ist.

¹¹<https://www.npmjs.com/package/bcrypt>

¹²<https://jwt.io/>

5.3 Frontend

5.3.1 Implementierung mit React

Zur Erstellung des Front-Ends der Webanwendung wird *React*,¹³ eine JavaScript-Bibliothek, verwendet. React ist ein Framework, welches seit 2013 von Facebook als Open-Source-Lösung bereitgestellt wird.¹⁴

Alternativen zur Erstellung von interaktiven Webanwendungen sind *Angular*¹⁵ und *Vue.js*.¹⁶ In dem vorliegenden Projekt wurde entschieden React zu verwenden, da die Technologie als einsteigerfreundlich eingeschätzt wurde und sich somit für das kollaborative Entwickeln in dem großen Projektteam sehr gut eignet. Ein weiterer Grund für die Verwendung von React ist, dass der Großteil des Projektteams bereits Erfahrung mit React gesammelt hat und die Technologie somit von den meisten bevorzugt wurde. Darüber hinaus lässt sich der Entwurf in Kapitel 4 auf Seite 11 mit React gut umsetzen, sodass die Anforderungen erfüllt werden können.

Mit React können effizient sogenannte Single-Page-Application (SPA) entwickelt werden. Eine SPA ist eine Webanwendung, die lediglich aus einer HTML-Seite besteht und deren Inhalte dynamisch zur Laufzeit nachgeladen bzw. erweitert werden. Der Kern von React liegt auf einem komponentenbasierten Aufbau der Webanwendungen sowie der jeweiligen Anordnung der Komponenten.¹⁷

Komponenten sind unabhängige, wiederverwendbare Entitäten, die durch eine hierarchische Komposition die Inhalte der SPA bestimmen und beliebig tief geschachtelt werden können. Hierbei werden die Struktur mit HTML, das Aussehen mit Cascading Style Sheets (CSS) und die Logik mit JavaScript gekapselt. In React werden zwei Typen von Komponenten unterschieden. Zum einen können funktionale Komponenten erzeugt werden, die JavaScript-Funktionen umfassen. Zum anderen werden klassenbasierte Komponenten verwendet. Diese sind JavaScript-Klassen, die eine höhere Komplexität aufweisen, jedoch auch mehr Funktionalitäten besitzen.

¹³<https://reactjs.org>

¹⁴Vgl. Hartmann und Zeigermann 2019, S. 3.

¹⁵<https://angular.io>

¹⁶<https://vuejs.org>

¹⁷Vgl. Hartmann und Zeigermann 2019, S. 3.

In Abbildung 5.3 ist die Ordnerstruktur des Front-Ends abgebildet, die unter anderem in dem Unterordner `src/Components` eine Übersicht über die Komponenten liefert.

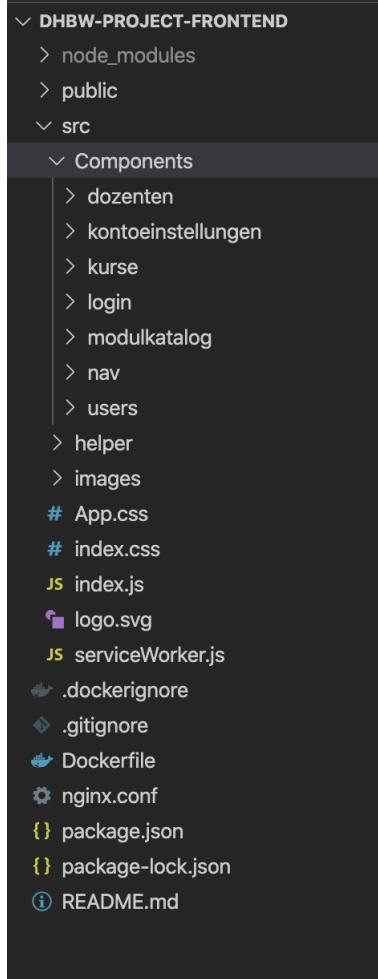


Abbildung 5.3: Übersicht der Ordnerstruktur des Front-Ends¹⁸

Sobald Änderungen in der Webanwendung vorgenommen werden, wird bei React nicht der komplette Seiteninhalt angepasst, sondern lediglich die tatsächlichen Veränderungen bei dem entsprechenden Document Object Model (DOM)-Element angepasst. Der Vorteil hierbei ist, dass die Benutzeroberfläche schneller auf Veränderungen reagieren kann.

Für eine einfachere Webentwicklung in React wurde *Material-UI*¹⁹ verwendet. Material-

¹⁸Verwendete Entwicklungsumgebung: Visual Studio Code (<https://code.visualstudio.com>)

¹⁹<https://material-ui.com>

UI ist ein Open-Source-Framework, das React Komponenten bereitstellt und dadurch eine schnelle Entwicklung ermöglicht. Ein einheitliches Design System für React und die vielen nützlichen UI-Komponenten waren ausschlaggebend für die Entscheidung für Material-UI.

Entsprechend dem Design Entwurf aus Kapitel ?? auf Seite ?? wurden die Komponenten so zusammengesetzt, dass sich der Seitenaufbau aus den Hauptkomponenten Kopfzeile, Navigationsleiste und Seiteninhalt zusammensetzt. In Abbildung 5.4 ist die Umsetzung entsprechend dem Design Entwurf abgebildet.

The screenshot shows the ExoPlan software interface. The top navigation bar (Kopfzeile) includes the logo 'ExoPlan DHBW Mannheim', a search bar with placeholder 'ABC17DEF KURS HINZUFÜGEN', and a user icon. Below the header, a sidebar on the left lists 'Vorlesungspläne', 'Dozenten', 'Modulkataloge', 'Datenverwaltung', and 'Administrationsbereich'. The main content area (Seiteninhalt) displays a table for 'Vorlesung' with rows for 'Marketing' and 'BWL'. The table columns are 'Vorlesung', 'SWS', 'Kursleistung', 'Dozent', and 'Status'. The status for Marketing is 'angeschrieben' and for BWL is 'bestätigt'. At the bottom is a weekly calendar view from Sunday to Saturday, with Monday the 6th highlighted in red. The sidebar has an orange background and is labeled 'Navigationsleiste'.

Abbildung 5.4: Umsetzung der Benutzeroberfläche

5.3.2 Styleguide für die Umsetzung

Für ein einheitliches Erscheinungsbild der Webanwendung wurde in einem Styleguide definiert, welche Gestaltungsrichtlinien für die Entwicklung des Front-Ends gelten. Der Guide wurde den Front-End-Entwicklern zur Verfügung gestellt, damit sie diesen bei ihrer Entwicklung berücksichtigen können. In Abbildung 5.5 auf der nächsten Seite

und 5.6 ist ein Auszug des Styleguides hinsichtlich des Logos und den einheitlichen Farben abgebildet. Der vollständige Styleguide ist in Anhang B auf Seite 44 einzusehen.



Abbildung 5.5: Auszug Styleguide für das Logo

Color		
Primary	#e2001a	rgb(226, 0, 26)
Primary- Light	#ff5444	rgb(255, 84, 68)
Primary - Dark	#a70000	rgb(167, 0, 0)
Secondary	#5c6971	rgb(92, 105, 13)
Secondary - Light	#89979f	rgb(137, 151, 159)
Secondary - Dark	#323e49	rgb(50, 62, 73)

Abbildung 5.6: Auszug Styleguide für das Farbdesign

5.3.3 Login

Die Webanwendung startet gemäß der Anforderung A4 aus Tabelle 3.1 auf Seite 5 mit einem Login, sodass sich der Benutzer anmelden oder registrieren kann. In Abbildung 5.7 ist die Login-Seite der Software zu sehen.

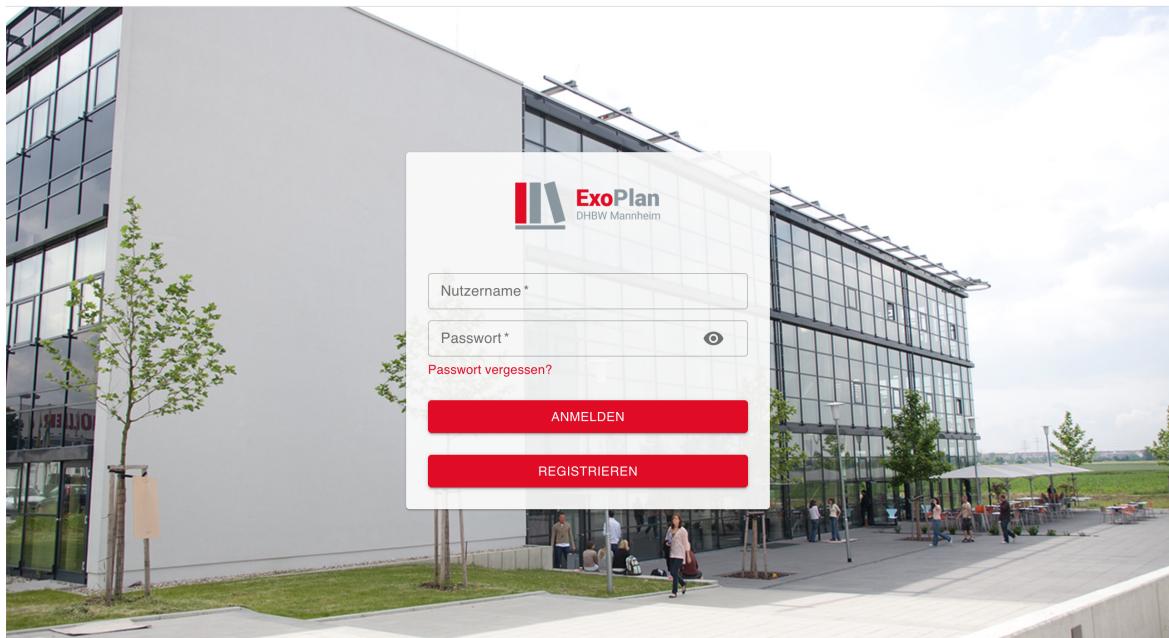


Abbildung 5.7: Login

5.3.4 Kursübersicht

Die Hauptansicht der Webanwendung ist die in Abbildung 5.8 auf der nächsten Seite dargestellte Seite. Hier werden die unterschiedlichen Kurse mit den jeweiligen Semestern, den Vorlesungen und dem eingebundenen Kurskalender angezeigt. Es können bereits angelegte Kurse verwaltet sowie neue Kurse angelegt werden. In Unterkapitel 5.3.10 auf Seite 32 wird die Einbindung des Kalendars sowie die Verbindung mit dem Google Calendar näher erläutert, da dies eine zentrale Komponente des Front-Ends darstellt.

The screenshot shows the ExPlan software interface. At the top, there is a navigation bar with links for 'Vorlesungspläne', 'Dozenten', and 'Modulkataloge'. Below this is a search bar with the placeholder 'ABC17DEF' and a button 'KURS HINZUFÜGEN'. A table header 'SEMESTER 1' through 'SEMESTER 6' is visible. Under 'Vorlesung', there are entries for 'Marketing' (SWS 20, Kursleistung K, Dozent Prof. Dr. Max Mustermann) and 'BWL' (SWS 40, SE, Dozent Willi Willisch). A status column shows 'angeschrieben' and 'bestätigt'. Below the table is a calendar for '6-12 JULY 2020'. The calendar grid shows days from Monday to Sunday. On Monday (6), there is a block labeled 'Wissenschaftliches Arbeiten...' from 1:30 PM to 5:30 PM. On Tuesday (7), there is a block labeled 'Fortgeschrittenes Programm...' from 1:00 PM to 5:00 PM. On Thursday (9), there is a block labeled 'Kosten- und Leistungsrechnen...' from 2:00 PM to 5:00 PM. The bottom right of the calendar has a red button 'VORLESUNG IM KALENDER EINTRÄGEN'.

Abbildung 5.8: Kursübersicht mit Vorlesungskalender

5.3.5 Dozentenansicht

Unter dem Navigationspunkt *Dozenten* wird der Dozentenpool dargestellt sowie die Möglichkeit zum Hinzufügen von neuen Dozenten gegeben. Die Umsetzung der Anforderungen A1, A8, A9 und A12 aus Tabelle 3.1 auf Seite 5 ist in Abbildung 5.9 auf der nächsten Seite zu sehen. Wird ein Dozent ausgewählt, erscheinen weitere Informationen in der Detail-Ansicht. Diese beinhaltet außerdem Funktionen, wie zum Beispiel das Hinterlegen eines Lebenslaufs.

Name	Vorlesungen	Studiengangsleiter
Elliot Alderson (extern) Tel.: (212) 555-0179 elliottalderson@protonmail.ch	IT-Sicherheit Machine Learning / AI / Data Science	jreichwald
Dr. Emmett L. Brown (extern) Tel.: +1 1210 1985 2204 emmett.l@brown.edu	Projektmanagement Software-Entwicklung	jreichwald
B. Sc. Tony Maldonary (intern) Tel.: 0123456789 Tony.Maldonary@bachelor-of-science.com	Projektmanagement Webentwicklung Recht Datenbanken	görmert
Quirinus Quirrell (extern) Tel.: +44 578 2108 quirrell@hogwarts.co.uk		jreichwald

Abbildung 5.9: Anzeige der Dozenten

5.3.6 Modulkatalog

Mit dem Navigationspunkt *Modulkatalog* wird die Anforderungen A4 aus Tabelle 3.1 auf Seite 6 umgesetzt. Es können Modulkataloge hinzugefügt, angezeigt sowie verwaltet werden.

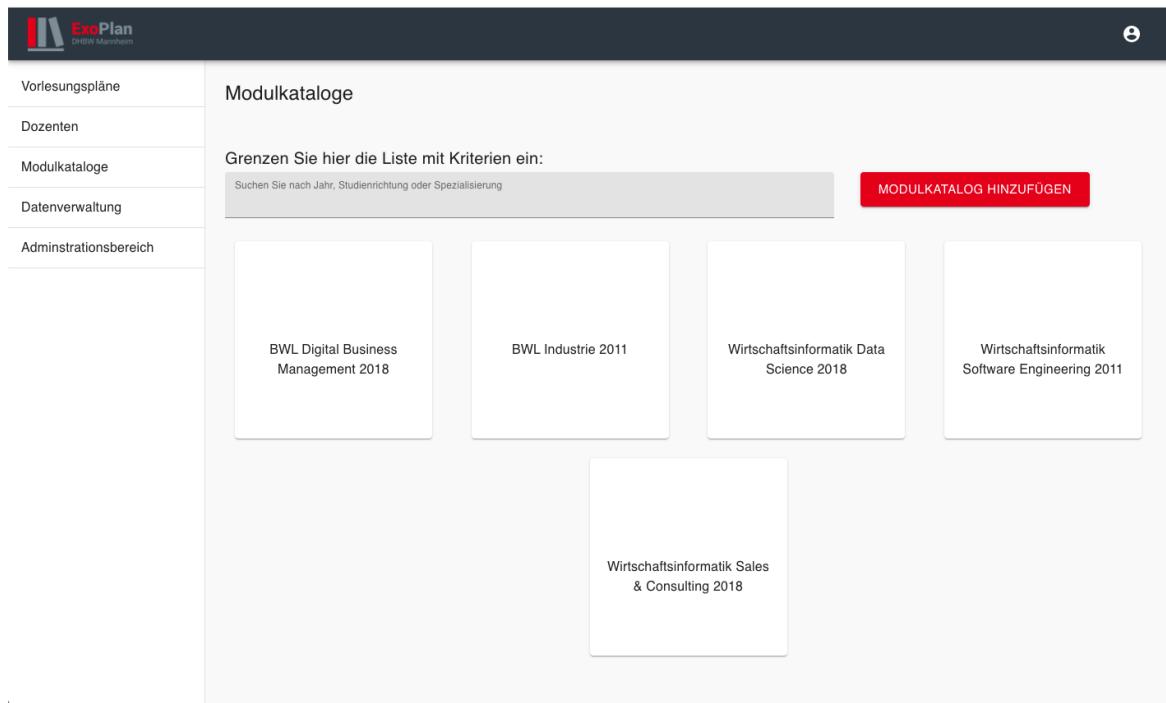


Abbildung 5.10: Anzeige der Modulkatalogen

5.3.7 Datenverwaltung

In der Datenverwaltung können die Studiengänge, die Schwerpunkte sowie die Prüfungsleistungen verwaltet werden. In Abbildung 5.11 auf der nächsten Seite ist das Hinzufügen, Bearbeiten und Löschen von Studiengängen abgebildet.

The screenshot shows the ExoPlan software interface. On the left, there is a sidebar with links: 'Vorlesungspläne', 'Dozenten', 'Modulkataloge', 'Datenverwaltung', and 'Administrationsbereich'. The main area has tabs at the top: 'STUDIENGÄNGE' (which is selected and highlighted in red), 'SCHWERPUNKTE', and 'PRÜFUNGSLEISTUNGEN'. A red button labeled 'Studiengang hinzufügen' is located in the top right. Below these, a table lists 'Studiengänge' with columns 'Aktionen' (Actions) and three entries: 'BWL', 'Digitale Medien', and 'Wirtschaftsinformatik', each with edit and delete icons.

Abbildung 5.11: Datenverwaltung

5.3.8 Administrationsbereich

Der Administrationsbereich erlaubt es Nutzern mit der Benutzerart *Administrator* andere Benutzer zu verwalten. Neben dieser Funktionalität, in Abbildung 5.12 auf der nächsten Seite dargestellt, können Registrierungsschlüssel sowie der Zugang zu dem Google Calendar verwaltet werden.

The screenshot shows the ExoPlan software interface. On the left is a sidebar with links: 'Vorlesungspläne', 'Dozenten', 'Modulkataloge', 'Datenverwaltung', and 'Administrationsbereich'. The main area has tabs at the top: 'BENUTZER' (which is selected and highlighted in red), 'REGISTRIERUNGSSCHLÜSSEL', and 'GOOGLE CALENDAR'. A red button labeled 'Studiengangsleiter hinzufügen' is visible. Below these tabs is a table with the following data:

ID	Benutzername/E-Mail	Benutzerart	Passwortstatus
1	admin	Administrator	OK
5	görmert	Benutzer +	OK :)
2	jreichwald	Benutzer +	OK :)
4	matt	Benutzer +	OK :)
3	ritterbusch	Benutzer +	OK :)

Abbildung 5.12: Administrationsbereich

5.3.9 Allgemeine Einstellungen

Zusätzlich können allgemeine Einstellungen für den Benutzer des Profils verändert werden, die in Abbildung 5.13 auf der nächsten Seite dargestellt sind. So kann der Benutzer unter anderem sein Passwort ändern.

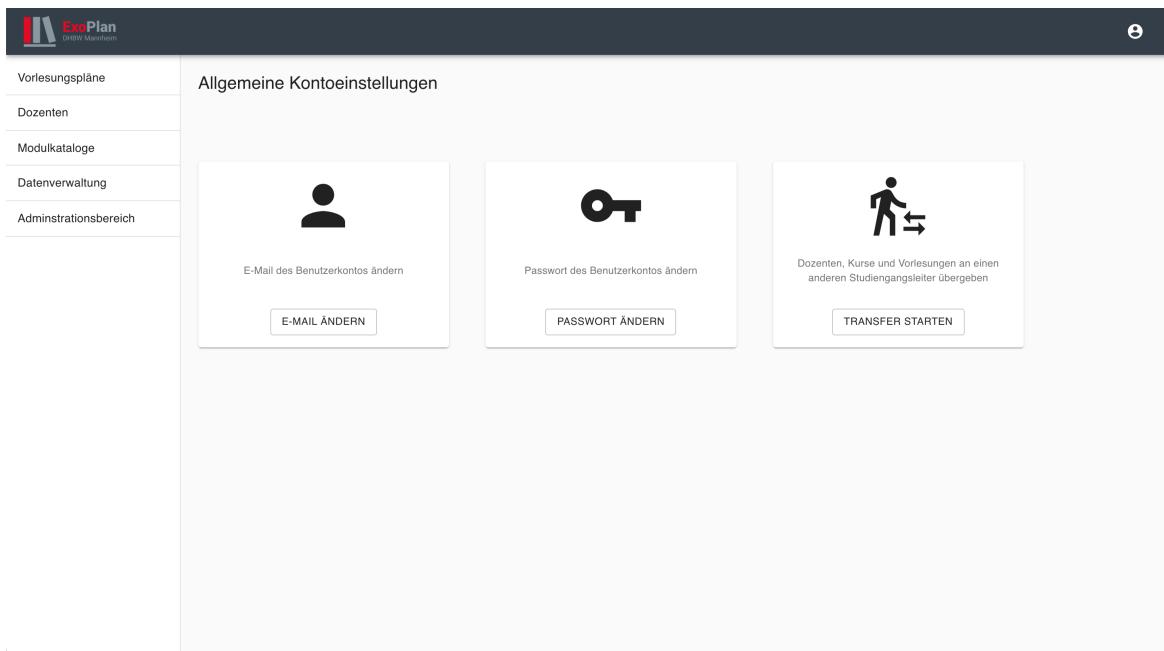


Abbildung 5.13: Allgemeine Kontoeinstellungen

5.3.10 Vorlesungskalender mit Google Calendar

Damit der Studiengangsleiter die Vorlesungen organisieren kann, wird der Google Calendar gemäß der Anforderung A2 und A3 aus der Tabelle 3.1 auf Seite 5 vollständig in die Software integriert. Damit die Ereignisse der jeweiligen Google Calendar angezeigt sowie verwaltet werden können, muss ein Kalender in React implementiert werden.

Einbindung des DevExtreme React Scheduler

Der *DevExtreme React Scheduler*²⁰ ist eine Komponente für Material-UI, die einen Kalender für React bereitstellt. Das Erscheinungsbild des React Schedulers ist von dem Google Calendar inspiriert und benutzerfreundlich gestaltet.²¹ Neben Funktionalitäten, wie beispielsweise Drag-and-Drop-Operationen sowie unterschiedlichen Anzeigemöglichkeiten, ist insbesondere die Anbindung und Integration eines Google Calendars

²⁰<https://devexpress.github.io/devextreme-reactive/react/scheduler/>

²¹Vgl. Developer Express Inc. 2020.

über geeignete Schnittstellen ausschlaggebend für die Entscheidung zur Verwendung der Komponente.

In Abbildung 5.14 ist eine Übersicht mit erklärenden Ergänzungen über den implementierten und konfigurierten React Scheduler dargestellt. Zur Verwendung des React Schedulers wird ein Package und entsprechende Abhängigkeiten von NPM eingebunden.²² Weitere Informationen zur Einbindung sowie Konfiguration des Kalenders sind in der Dokumentation von DevExtreme gegeben.²³

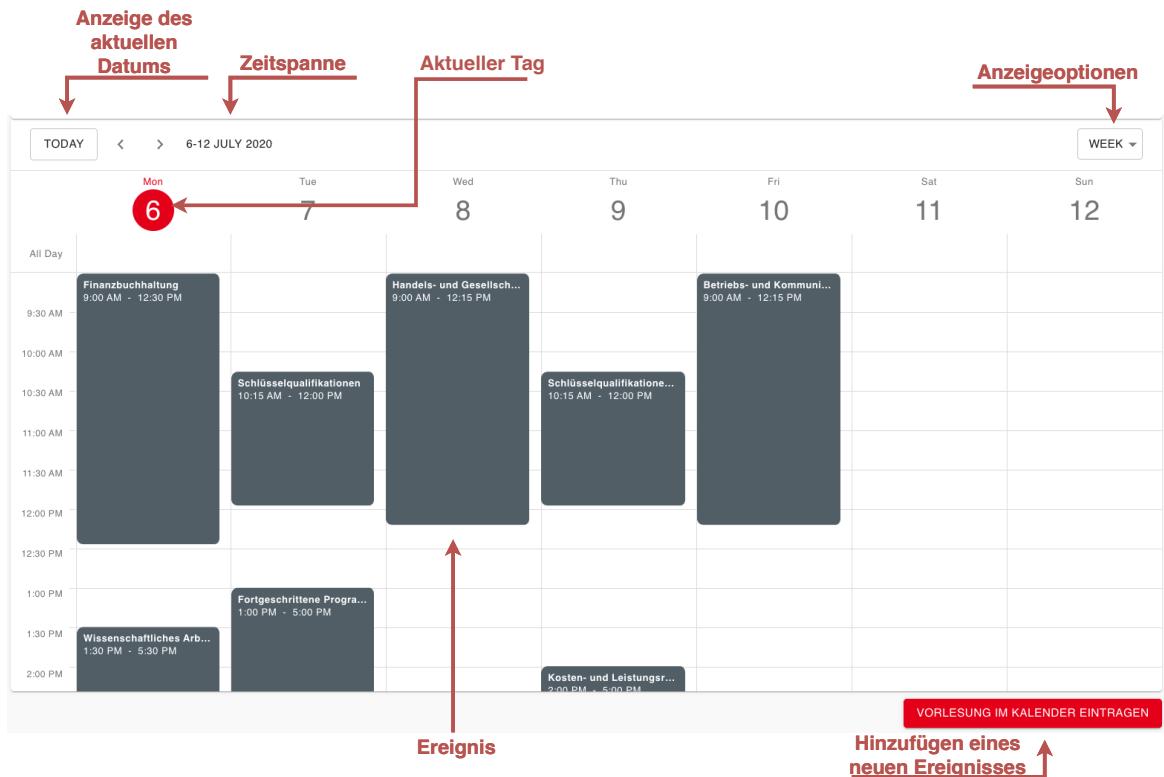


Abbildung 5.14: Übersicht des React Schedulers

²²<https://www.npmjs.com/package/@devexpress/dx-react-scheduler>

²³<https://devexpress.github.io/devextreme-reactive/react/scheduler/docs/guides/getting-started/>

Initialisierung und Verbindung des Google Calendars

Der Google Calendar ermöglicht eine Anbindung an das System über eine API. Damit die Inhalte eines Google Calendars über die Google Calendar-API-Schnittstelle²⁴ angefragt und in dem React Scheduler angezeigt werden können, müssen die Anfragen über einen Token authentifiziert werden. Hierfür wird OAuth 2.0 verwendet, um die Authentifizierung durchzuführen.²⁵

In einer eigenen JavaScript-Datei `apiHandlerGoogleCalendar.js` wird der Token bei den Anfragen mitgegeben. Dieser Token wird nach einem einmaligen Bestätigen des Zugriffs auf den Google Calendar erstellt und hinterlegt. Mit einem validen Token werden die Anfragen an die Google Calendar-API genehmigt und die Inhalte des jeweiligen Kalenders an die Anwendung übertragen. Die erhaltenen Inhalte werden formatiert und in dem React Scheduler angezeigt.

Interaktion mit dem Kalender

Zur Funktionalität des Kalenders werden drei unterschiedliche Interaktionen mit dem Google Calendar benötigt: das Hinzufügen, das Ändern sowie das Löschen von Ereignissen. Im Folgenden werden die Anfragen an die REST-API des Google Calendars näher beschrieben.

Hinzufügen von Ereignissen

In dem React Scheduler können neue Ereignisse dem Kalender hinzugefügt werden. In Abbildung 5.15 auf der nächsten Seite ist das Dialog-Fenster abgebildet, welches nach dem Drücken des Hinzufügebuttons erscheint.

²⁴<https://developers.google.com/calendar/v3/reference>

²⁵Vgl. o.V. 2020.

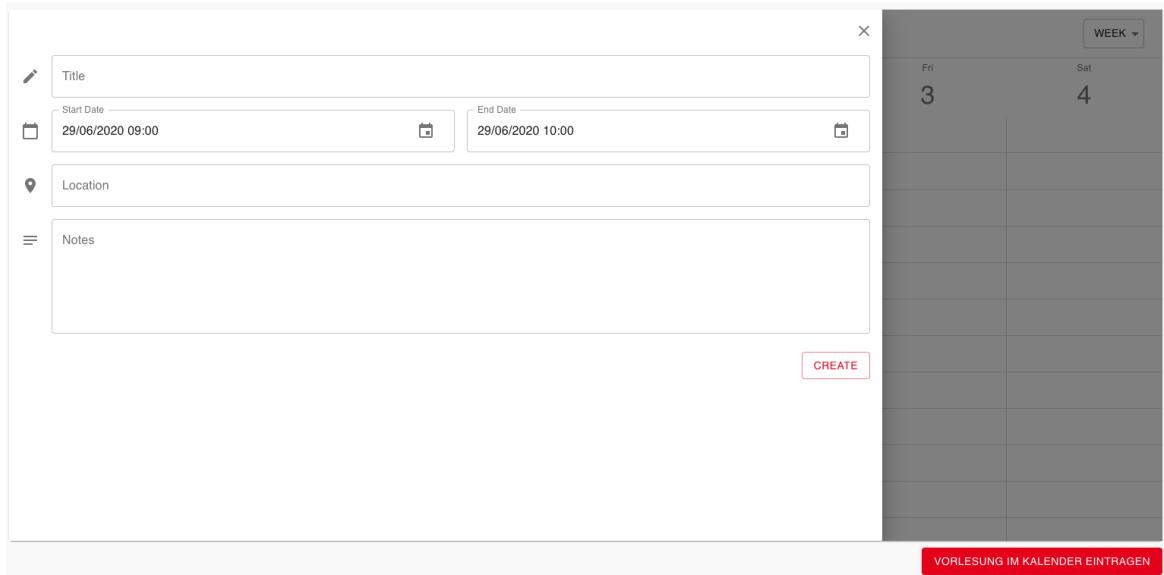


Abbildung 5.15: Erstellen eines Ereignisses im React Scheduler

Zusätzlich zu den abgebildeten Inhalten werden die Kalender-ID und der Token bei der PUSH-Anfrage übermittelt. Hierbei wird die Bibliothek *GAPI* von Google verwendet, die eine einfache Verbindung mit der API über browserseitiges JavaScript ermöglicht. Eine detaillierte Dokumentation dieser Anfrage ist in der Schnittstellenbeschreibung der Google Calendar API gegeben.²⁶

Ändern von Ereignissen

Vorhandene Ereignisse können ebenfalls bearbeitet werden, indem die entsprechende Veranstaltung ausgewählt und der Bearbeiten-Button auf dem erscheinenden Popup geklickt wird, welches in Abbildung 5.16 dargestellt ist.

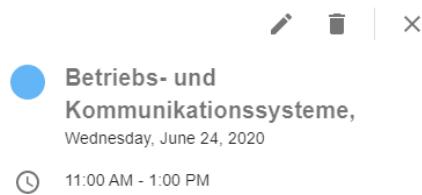


Abbildung 5.16: Popup eines Ereignisses im React Scheduler

²⁶<https://developers.google.com/calendar/v3/reference/events/insert>

Zur Änderung der Daten wird eine API-Anfrage mit der Methode *update* verwendet.²⁷ Diese ist vom Aufbau ähnlich zu der zuvor vorgestellten *insert*-Methode.

Löschen von Ereignissen

Zusätzlich können Ereignisse in dem Google Calendar gelöscht werden, indem die Funktionalität, ebenfalls in Abbildung 5.16 auf der vorherigen Seite abgebildet, genutzt wird. Zur Durchführung dieser Operation wird das Event *delete* verwendet, wobei lediglich die Calendar-ID sowie die Event-ID übermittelt werden.²⁸ In dem Code-Ausschnitt 5.1 ist die Erstellung der Delete-Anfrage abgebildet.

```
1 function handleAppointmentDelete(deleteAppointmentId, gapi) {  
2     var request = gapi.client.calendar.events.delete({  
3         'calendarId': creds.calenderID,  
4         'eventId': deleteAppointmentId  
5     });  
6  
7     request.execute(function (response) {  
8         if (response.error || response == false) {  
9             alert('Error');  
10        } else {  
11            alert('Success');  
12        }  
13    });  
14}
```

Quelltext 5.1: Anfrage zum Löschen eines Ereignisses

²⁷<https://developers.google.com/calendar/v3/reference/events/update>

²⁸<https://developers.google.com/calendar/v3/reference/events/delete>

5.4 Verknüpfung Backend und Frontend

5.5 Zugriffsberechtigungen

5.5.1 Zugriffsberechtigungen

5.6 Test

6 User Guide

6.1 Setup

6.2 Anwendungsanleitung

6.3 Funktionalitäten

6.4 Erweiterbarkeit

7 Evaluation

7.1 Bewertung Umsetzung

- Abgleich Anforderungen
- Derzeitiger Stand der Software

7.2 Lessons Learned

7.3 Nächste Schritte

- Erweiterbarkeit

8 Fazit und Ausblick

Anmerkungen Martin (01.07.20)

- Nicht alle Routen komplett nach REST-Paradigma erstellt. Ist bei dem Weiterführen des Projekts eine Möglichkeit zur Verbesserung
- in Entwicklung Verwendung von selbstsignierten Zertifikaten, muss vor dem deployment geändert werden
- Aktuell ist Datenbank nur so lange persistent wie Docker-Container nicht weggeworfen wird
 - > in laufendem Betrieb: regelmäßige Backups durchführen und/oder Container auf eine persistente Schicht (z.B. Docker-Volume) mounten

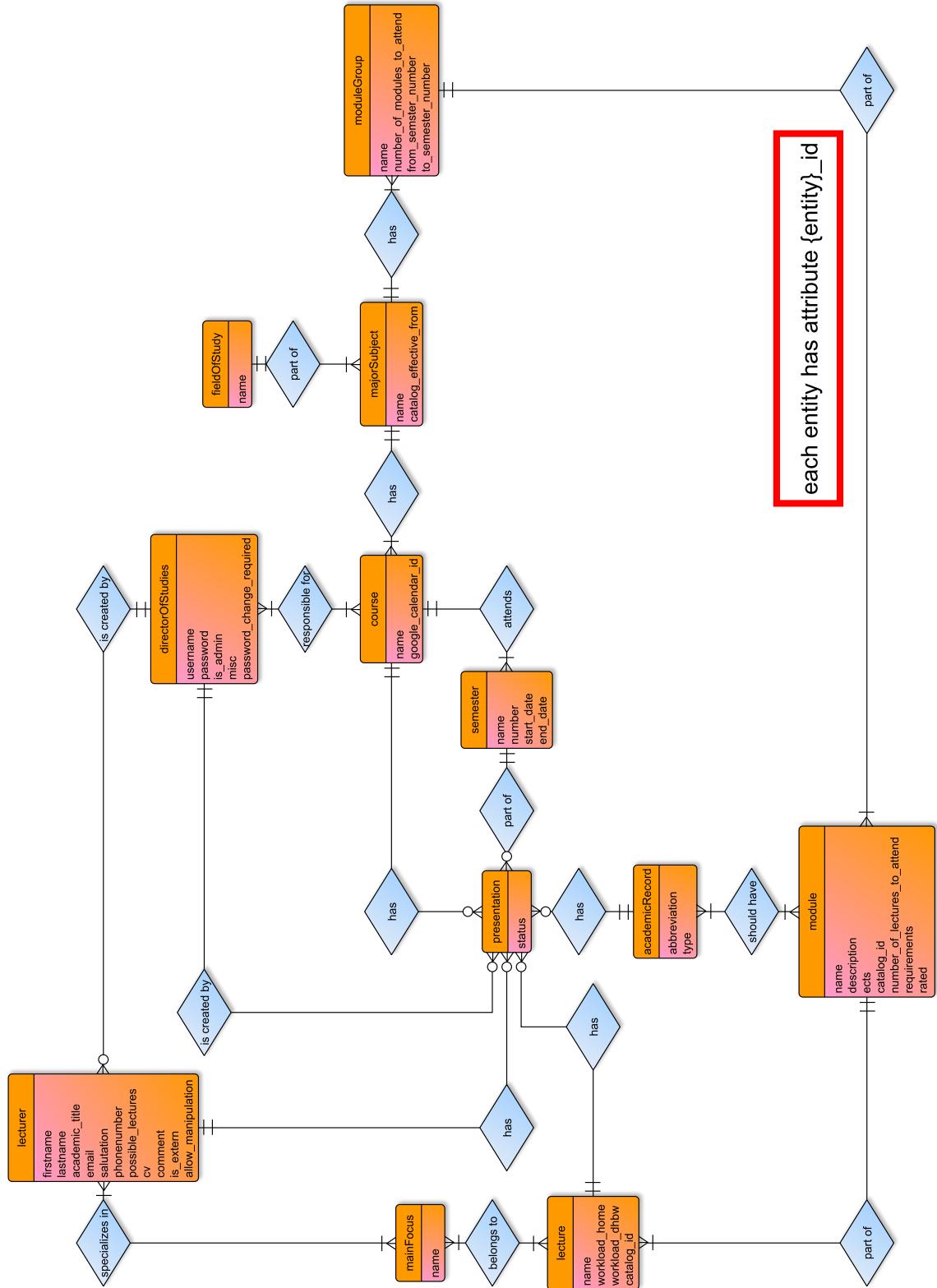
Quellenverzeichnis

Developer Express Inc. (2020). *React Scheduler for Material-UI*. URL: <https://devexpress.github.io/devextreme-reactive/react/scheduler/> (besucht am 27.06.2020).

Hartmann, Nils und Oliver Zeigermann (2019). *React: Grundlagen, fortgeschrittene Techniken und Praxistipps – mit TypeScript und Redux*. 2. Aufl. Heidelberg: dpunkt.verlag. ISBN: 978-3-86490-552-0.

o.V. (2020). *Authorizing Requests to the Google Calendar API*. URL: <https://developers.google.com/calendar/auth> (besucht am 27.06.2020).

A ER-Modell



B Styleguide

Logo	 The logo consists of a stylized 'E' icon followed by the word 'ExoPlan' in a bold sans-serif font, with 'DHBW Mannheim' in smaller letters below it.	 The logo consists of a stylized 'E' icon followed by the word 'ExoPlan' in a bold sans-serif font, with 'DHBW Mannheim' in smaller letters below it.
Typo	<p>Roboto Regular</p> <p>ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789</p>	<p>Roboto Medium</p> <p>ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789</p>
	<p>Roboto Black</p> <p>ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789</p>	
Color	<p>Primary #e2001a rgb(226, 0, 26)</p> <p>Primary- Light #ff5444 rgb(255, 84, 68)</p> <p>Primary - Dark #a70000 rgb(167, 0, 0)</p>	
	<p>Secondary #5c6971 rgb(92, 105, 13)</p> <p>Secondary - Light #89979f rgb(137, 151, 159)</p> <p>Secondary - Dark #323e49 rgb(50, 62, 73)</p>	
Headlines	<h1>Headline 1</h1> <h2>Headline 2</h2> <h3>Headline 3</h3> <h4>Headline 4</h4> <h5>Headline 5</h5> <h5>Headline 5</h5>	

