

**Technische Hochschule Deggendorf**  
**Fakultät Angewandte Informatik**

Studiengang Bachelor Angewandte Informatik

**ENTWURF UND IMPLEMENTIERUNG EINER APP FÜR DAS  
INTERNATIONAL OFFICE DER TECHNISCHEN HOCHSCHULE  
DEGGENDORF**

**DESIGN AND IMPLEMENTIERUNG OF AN APP FOR THE  
INTERNATIONAL OFFICE OF THE TECHNICAL UNIVERSITY  
DEGGENDORF**

Bachelorarbeit zur Erlangung des akademischen Grades:

*Bachelor of Engineering (B.Eng.)*

an der Technischen Hochschule Deggendorf

Vorgelegt von:

Therese Grace Bondoum

Matrikelnummer: 00692309

Am: 02. April 2024

Prüfer:

Prof. Dr. -Ing. Udo Garmann

Zweitprüfer:

Prof. Dr. Jüttner

## Erklärung

Name des Studierenden: Therese Grace Bondoum

Name des Betreuenden: Prof. Dr. -Ing. Udo Garmann

Thema der Abschlussarbeit:

Entwurf Und Implementierung einer App für das International Office der Technischen Hochschule Deggen-  
dorf .....


.....

.....

.....

1. Ich erkläre hiermit, dass ich die Abschlussarbeit gemäß § 35 Abs. 7 RaPO (Rahmenprüfungsordnung für die Fachhochschulen in Bayern, BayRS 2210-4-1-4-1-WFK) selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Deggendorf, 02.04.2024  
Datum

  
Unterschrift des Studierenden

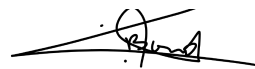
2. Ich bin damit einverstanden, dass die von mir angefertigte Abschlussarbeit über die Bibliothek der Hochschule einer breiteren Öffentlichkeit zugänglich gemacht wird:

☒ Nein

☐ Ja, nach Abschluss des Prüfungsverfahrens

☐ Ja, nach Ablauf einer Sperrfrist von ...Jahren.

Deggendorf, 02.04.2024  
Datum

  
Unterschrift des Studierenden

---

Bei Einverständnis des Verfassenden vom Betreuenden auszufüllen:

Eine Aufnahme eines Exemplars der Abschlussarbeit in den Bestand der Bibliothek und die Ausleihe des Exemplars wird:

☐ Befürwortet

☐ Nicht befürwortet

Deggendorf, .....  
Datum

.....  
Unterschrift des Betreuenden

# Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit der Konzeption und Umsetzung einer benutzerfreundlichen und funktionalen mobilen Anwendung für das International Office der Technischen Hochschule Deggendorf. Das Hauptziel dieser App besteht darin, den Austauschstudierenden sowie den Mitarbeitenden des International Office eine effiziente Plattform zu bieten, um Informationen zu Austauschprogrammen, Veranstaltungen, Unterstützungsleistungen und anderen relevanten Themen schnell und einfach zugänglich zu machen.

Der Entwurfsprozess umfasste eine gründliche Analyse der Anforderungen des International Office und der Bedürfnisse der Zielgruppe. Basierend auf diesen Erkenntnissen wurde ein umfassendes Designkonzept entwickelt, das eine intuitive Benutzeroberfläche und eine klare Informationsstruktur gewährleistet.

Die Implementierung der Anwendung erfolgte unter Verwendung modernster Technologien und Frameworks wie Ionic-Angular für das Frontend und Node.js in Kombination mit dem Strapi Content-Management-System für das Backend. Besonderes Augenmerk wurde auf Sicherheit, Skalierbarkeit und Benutzerfreundlichkeit gelegt. Die App wurde um Funktionen wie mehrsprachige Unterstützung, die nahtlose Integration von Kalendern sowie die Möglichkeit zum Herunterladen wichtiger Dokumente erweitert, um eine reibungslose Interaktion zwischen den Nutzern und dem International Office zu gewährleisten.

# Inhaltsverzeichnis

<b>Zusammenfassung</b>	<b>iii</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Hintergrund und Motivation . . . . .	1
1.1.1 Hintergrund . . . . .	1
1.1.2 Motivation . . . . .	1
1.2 Problemstellung . . . . .	2
1.3 Zielsetzung . . . . .	2
1.4 Gliederung der Arbeit . . . . .	2
<b>2 Technischer Grundlagen</b>	<b>4</b>
2.1 Frameworks . . . . .	4
2.1.1 Node Js . . . . .	4
2.1.2 Angular . . . . .	5
2.1.3 Ionic . . . . .	7
2.1.4 Strapi . . . . .	9
2.2 Technologie und Sprachen . . . . .	12
2.2.1 HTML . . . . .	12
2.2.2 TypeScript . . . . .	12
2.2.3 SASS . . . . .	13
2.2.4 MySQL . . . . .	13
2.3 Bibliotheken,Software,Betriebssysteme,IDE und Plugins . . . . .	14
2.3.1 Visual-Studio-Code . . . . .	14
2.3.2 Angular-Material . . . . .	14
<b>3 Theoretischer Grundlagen</b>	<b>16</b>
3.1 Internationales Hochschulmanagement . . . . .	16
3.1.1 Prinzipien . . . . .	16
3.1.2 Aufgaben des International Office . . . . .	16
3.1.3 Orientierungswoche für Austauschstudenten . . . . .	17
3.1.4 Herausforderungen bei der Betreuung von Austauschstudenten . . . . .	17
<b>4 Anforderungen</b>	<b>18</b>
4.1 Darstellung der Orientierungsweek Zeitplan . . . . .	18
4.2 Bereitstellung von Dokumenten zum Herunterladen . . . . .	18
4.3 Das Abholservice . . . . .	18
4.4 Sprachenvielfalt . . . . .	18
4.5 Präsentation der häufig gestellten Fragen (FAQ) . . . . .	18
4.6 Darstellung einer Checkliste . . . . .	18
4.7 Die Benachrichtigung für den Kurs-Deadline-Reminder . . . . .	18
4.8 Zielplattformen . . . . .	19
<b>5 Konzeption und Design</b>	<b>20</b>
5.1 Architektur der Anwendung . . . . .	20
5.1.1 Mobile Anwendung . . . . .	20
Hybrid Mobile App . . . . .	20
Native Mobile App . . . . .	21

Mobile Web App . . . . .	22
5.2 Cross-Platform-Entwicklung . . . . .	23
5.2.1 Aktuelle Cross-Platform-Frameworks . . . . .	24
Apache Cordova . . . . .	24
Xamarin . . . . .	26
Ionic . . . . .	26
React Native . . . . .	26
5.3 Benutzeroberfläche und Interaktionsdesign . . . . .	27
5.4 Sicherheitsaspekte . . . . .	28
<b>6 Implementierung</b>	<b>30</b>
6.1 Umsetzung der Funktionalitäten . . . . .	30
6.1.1 Header-Design . . . . .	30
6.1.2 Orientierungswoche Zeitplan . . . . .	31
6.1.3 Dokumente Herunterladen . . . . .	32
6.1.4 Pickup-Service . . . . .	33
6.1.5 Faq . . . . .	34
6.1.6 Checkliste . . . . .	35
6.2 Backend-Implementierung mit Strapi . . . . .	36
6.2.1 Collections Type . . . . .	37
6.2.2 Media Library . . . . .	37
6.3 Frontend-Implementierung mit Ionic Angular . . . . .	38
6.3.1 Orientierungswoche im Blick . . . . .	38
6.3.2 Überblick zum Herunterladen von Dokumenten . . . . .	40
6.3.3 Einblick in den Pickup-Service . . . . .	42
6.3.4 Darstellung der Faq . . . . .	45
6.3.5 Implementierung der Checkliste . . . . .	46
6.4 Internationalisierung . . . . .	47
<b>7 Evaluation und Verbesserung</b>	<b>48</b>
7.1 Usability-Tests . . . . .	48
7.1.1 Benutzerfreundlichkeit . . . . .	48
7.1.2 Leistung . . . . .	48
7.1.3 Visuel Erscheinungsbild und Design . . . . .	48
7.1.4 Nützlichkeit der Funktion . . . . .	48
7.1.5 Nicht Erfüllte Anforderungen . . . . .	49
7.2 Verbesserungsvorschläge und Anpassung . . . . .	49
<b>8 Schlussfolgerungen</b>	<b>50</b>
8.1 Zusammenfassung . . . . .	50
8.2 Ausblick . . . . .	50
<b>Abbildungsverzeichnis</b>	<b>52</b>
<b>Literaturverzeichnis</b>	<b>53</b>

# 1 Einleitung

Die Entwicklung einer mobilen Webanwendung wird von verschiedenen Personen unterschiedlich interpretiert. Während einige sie als eine Anwendung betrachten, die im Web läuft und so gestaltet wurde, dass sie auf einem mobilen Gerät korrekt angezeigt wird, sehen andere darin eine native Anwendung, die speziell für ein bestimmtes mobiles Betriebssystem entwickelt wurde und über das Web mit Daten kommuniziert[1].

Die Bedeutung und Verwendung von mobilen Anwendungen haben in den letzten Jahren stark zugenommen. Ihr Fortschritt ist unübersehbar. Historisch gesehen lag der Schwerpunkt der Arbeit an Webapplikationen hauptsächlich auf der Serverseite. Die meisten Berechnungen und Datenverarbeitungen wurden vom Server durchgeführt, da nur dieser über ausreichende Rechenleistung verfügte, um komplexe Anwendungen zu erstellen. Der Browser war in erster Linie dafür da, Benutzerdaten an den Server zu senden und die Ergebnisse zu präsentieren. Im Laufe der Zeit haben sich jedoch die Leistungsfähigkeit der Client-Computer und die Technologien von Webbrowsers stetig verbessert, was zu einer Verschiebung von Teilen der Verarbeitung auf der Clientseite geführt hat[2].

Diese Entwicklung hat nicht nur die Möglichkeiten für die Erstellung und Nutzung mobiler Anwendungen erweitert, sondern auch neue Herausforderungen und Chancen geschaffen. Die Integration von server- und clientseitiger Verarbeitung sowie die zunehmende Bedeutung von plattformübergreifenden Technologien haben die Landschaft der mobilen Entwicklung grundlegend verändert. Es ist daher wichtig, diese Entwicklungen zu verstehen und angemessen in die Gestaltung und Implementierung mobiler Anwendungen einzubeziehen.

## 1.1 Hintergrund und Motivation

### 1.1.1 Hintergrund

Der Ursprung dieses Projekts liegt in dem wachsenden Bedarf von Bildungseinrichtungen, ihre Abläufe durch den Einsatz effizienter und anwendungsfreundlicher Technologien zu optimieren. Das International Office der Technischen Hochschule Deggendorf (THD) spielt hierbei eine zentrale Rolle als Schnittstelle zwischen der Hochschule und internationalen Studierenden. Sein Ziel ist es, die Integration und Erfahrung der Studierenden während ihres Aufenthalts zu erleichtern. Vor diesem Hintergrund stellt die Entwicklung einer maßgeschneiderten mobilen Anwendung für das International Office eine innovative und zugleich notwendige Lösung dar.

Die THD, als angesehene Bildungseinrichtung, verfolgt aktiv das Ziel, ihre internationalen Bemühungen zu verstärken. Das International Office fungiert als Anlaufstelle für Studierende und Mitarbeiter, die an internationalen Austauschprogrammen teilnehmen oder aus dem Ausland kommen. Innerhalb dieses Rahmens entsteht der Bedarf nach einer individuell gestalteten App, die die Verwaltungs- und Kommunikationsprozesse entscheidend verbessern kann.

### 1.1.2 Motivation

Die Entwicklung einer mobilen Anwendung für Studierende bringt zahlreiche Vorteile mit sich. Sie verbessert die Gesamterfahrung und Effizienz des studentischen Alltags erheblich. Mit einer solchen Anwendung können Studierende bequem von ihren mobilen Geräten aus auf verschiedene Dienste und Informationen zugreifen. Die App fördert eine effektive Kommunikation zwischen Studierenden und Hochschule Mitarbeiter. Außerdem bietet sie die Möglichkeit, sich über Veranstaltungen, Clubs und andere außerschulische Aktivitäten zu informieren und zu vernetzen. Durch die Integration von Funktionen wie Kalender, Erinnerungen und To-Do-Listen unterstützt die App auch eine effektive Zeitplanung und verbessert so die organisatori-

schen Fähigkeiten der Studierenden. Die Entwicklung einer mobilen Anwendung trägt insgesamt dazu bei, das Studium effizienter, zugänglicher und interaktiver zu gestalten. Dadurch wird das Studenumfeld für alle Beteiligten verbessert.

### 1.2 Problemstellung

Die Technische Hochschule Deggendorf hat eine wichtige Rolle im internationalen Studentenaustausch. Um den wachsenden Bedürfnissen und Herausforderungen des Internationalen Büros gerecht zu werden, ist es entscheidend, eine spezielle Anwendung für den Studentenaustausch zu entwickeln und zu implementieren. Die Anwendung soll den Austauschprozess erleichtern und verbessern. Die Anwendung soll den Informationsaustausch und die Kommunikation zwischen dem internationalen Büro und den Austauschschülern verbessern.

Sie muss benutzerfreundlich und einfach zu verwalten sein, auch für Mitarbeiter ohne Programmierkenntnisse. Um den Inhalt der App ohne technische Kenntnisse zu verwalten, ist die Implementierung eines einfachen Datenmanagementsystems erforderlich.

### 1.3 Zielsetzung

Das Ziel der Entwicklung und Implementierung der Anwendung für das International Office der Technischen Hochschule Deggendorf umfasst mehrere Schlüsselaspekte, um die Effizienz, Kommunikation und Zufriedenheit der Austauschstudierenden zu verbessern.

In Bezug auf die technischen Aspekte muss diese Anwendung die modernsten und passendsten Technologien verwenden, um eine optimale Benutzererfahrung und eine zuverlässige Leistung zu gewährleisten. Dies beinhaltet die Nutzung robuster und skalierbarer Entwicklungsframeworks, die es ermöglichen, plattformübergreifende mobile Anwendungen mit maximaler Effizienz zu entwickeln.

### 1.4 Gliederung der Arbeit

Die vorliegende Arbeit wurde sorgfältig strukturiert, um systematisch eine Vielzahl von Aspekten des behandelten Themas einzubeziehen. Beginnend mit dem ersten Kapitel der Einleitung, das als Orientierungshilfe dient, indem es nicht nur den Hintergrund und die Motivation der Arbeit erläutert, sondern auch die Kernproblematik, die angestrebten Ziele sowie die Methodik und den Arbeitsaufbau vorstellt. Das nächste Kapitel widmet sich den technischen Grundlagen und gibt einen detaillierten Überblick über die verwendeten Frameworks und Technologien wie Node.js, Angular, Ionic, Strapi, HTML, TypeScript, SASS und MySQL sowie die genutzten Bibliotheken und Software. Das dritte Kapitel beleuchtet darüber hinaus den theoretischen Rahmen mit weiterführenden Konzepten, darunter die Prinzipien, die Aufgaben des internationalen Hochschulmanagements, das Ziel der Orientierungswochen sowie die spezifischen Herausforderungen bei der Betreuung von Austauschstudierenden.

Im folgenden Abschnitt „Anforderungen“ werden verschiedene Bedürfnisse und Funktionen definiert. Dazu gehören unter anderem die übersichtliche Darstellung des Zeitplans der Orientierungswoche, die Bereitstellung von Download-Dokumenten, der sogenannte Pickup-Service, die Berücksichtigung der Sprachenvielfalt sowie die Integration eines FAQ-Bereichs zur Beantwortung häufig gestellter Fragen. Des Weiteren wird neben der Bereitstellung einer Checkliste in Form eines Formulars und der Sicherstellung der Kompatibilität mit verschiedenen Zielplattformen auch der Deadline-Reminder in diesem Kapitel erläutert. Das fünfte Kapitel bildet die Grundlagen für die Konzeption und Entwicklung der Anwendung. Diese beschreibt nicht nur die Kategorien mobiler Anwendungen, sondern stellt auch einige aktuelle Cross-Platform-Entwicklungen einer Anwendung vor. Ebenso wird eine detaillierte Diskussion über Architektur, Design, Benutzerschnittstelle und Sicherheitsaspekte eröffnet.

Weiterhin erläutert das Kapitel sechs die Implementierungsphase. Diese umfasst die Entwicklung der Funktionalitäten der Anwendung, angefangen beim Header-Design bis hin zur Integration von Features wie dem

Zeitplan für die Orientierungswoche, dem Herunterladen von Dokumenten, dem Pickup-Service, der Darstellung der häufig gestellten Fragen und schließlich der Darstellung des Checklistenformulars. Zusätzlich wird die Backend- und Frontend-Entwicklung mit den genannten Technologien Strapi und Ionic Angular behandelt. Gleichzeitig wird auch ein sehr wichtiges Thema, die so genannte Internationalisierung, erläutert. Das siebte Kapitel widmet sich der Bewertung der entwickelten Anwendung und den Verbesserungsmaßnahmen auf der Grundlage des erhaltenen Feedbacks, insbesondere der Benutzertests.

Im letzten Kapitel werden schließlich die Schlussfolgerungen gezogen, die die wichtigsten Erkenntnisse zusammenfassen, und ein Ausblick auf mögliche zukünftige Entwicklungen gegeben. Dabei wird auch auf die Auswirkungen der Forschung und potenzielle Anwendungen in der Praxis eingegangen. Zusätzlich wird die Arbeit durch ein Abbildungsverzeichnis und Literaturangaben abgerundet, wodurch eine gut strukturierte und leicht verständliche Darstellung des Themas gewährleistet wird, die es dem Leser erleichtert, die Inhalte zu erfassen und zu verstehen.



## 2 Technischer Grundlagen

### 2.1 Frameworks

#### 2.1.1 Node Js

Node.js repräsentiert eine plattformübergreifende Open-Source-JavaScript-Laufzeitumgebung und ist als vielseitiges Werkzeug für nahezu jede Projektart äußerst beliebt. Diese Umgebung nutzt die V8-JavaScript-Engine, das Herzstück von Google Chrome, außerhalb des Browsers, wodurch Node.js eine beeindruckende Leistung erzielen kann. Im Gegensatz zu traditionellen Ansätzen führt eine Node.js-Anwendung sämtliche Operationen in einem einzigen Prozess aus, ohne für jede Anfrage einen neuen Thread zu erzeugen. Die Standardbibliothek von Node.js bietet eine Palette von asynchronen I/O-Grundelementen, um das Blockieren von JavaScript-Code zu verhindern. Typischerweise sind Node.js-Bibliotheken nach dem nicht blockierenden Paradigma konzipiert, wodurch Blockadeverhalten eher die Ausnahme als die Regel ist[3].

Node.js setzt in E/A-Operationen auf eine fortlaufende Betriebsweise nach Abschluss der Anfrage, anstatt den Thread zu blockieren und auf das Ergebnis zu warten. Dadurch kann Node.js Tausende gleichzeitige Verbindungen mit einem einzigen Server verwalten, ohne die potenziellen Fehlerquellen der Thread-Parallelität in Kauf nehmen zu müssen. Ein bemerkenswerter Vorteil von Node.js ist, dass Frontend-Entwickler, die bereits JavaScript für browserseitigen Code verwenden, nun auch serverseitigen Code schreiben können. Sie müssen keine neue Programmiersprache erlernen. Millionen von Entwicklern profitieren von dieser Möglichkeit.[3].

Node.js basiert auf einer ereignisgesteuerten Architektur, die im Wesentlichen aus drei Schritten besteht:

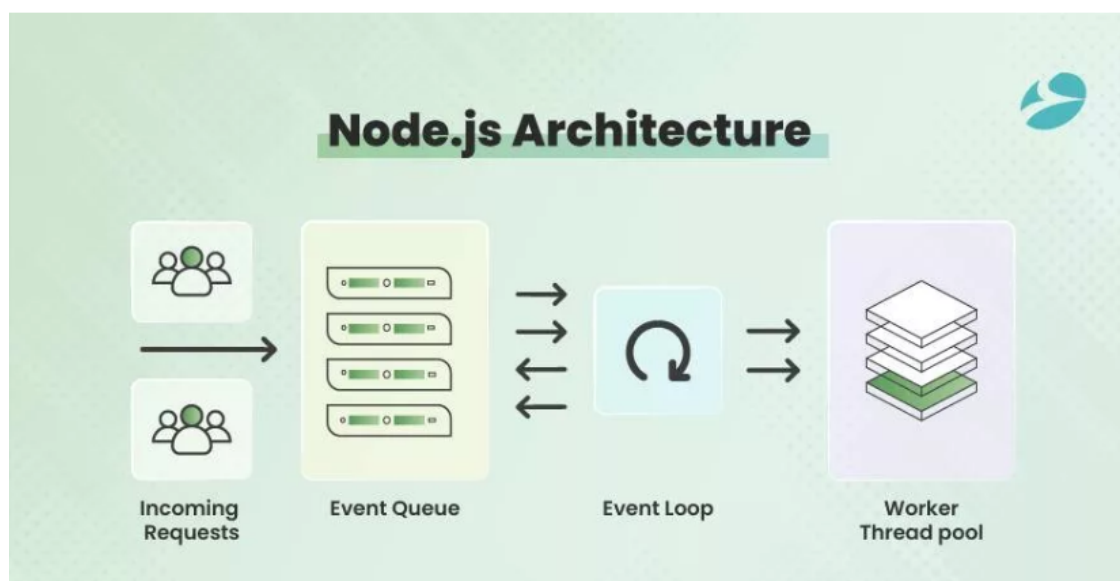


Abbildung 2.1: Node.js Architektur[4]

- Die Ausgabe von Ereignissen in die Ereignis-Queue
- Die Verarbeitung der generierten Ereignisse durch die Ereignisschleife

- Der Aufruf der Callbacks

Zwar handelt es sich um einen Single-Task, aber es kann einen Thread-Pool verwalten, um sowohl blockierende als auch nicht-blockierende I/O-Anforderungen zu erhalten [4].

Mit den vielseitigen Kernmodulen von Node.js ist es möglich, eine breite Palette von Serversystemen zu implementieren, sei es für beliebige TCP- oder UDP-Protokolle wie Domain Name System (DNS), HTTP, Internet Relay Chat (IRC) oder FTP. Obwohl Node.js in der Lage ist, Internet-Server oder -Clients zu entwickeln, liegt sein größter Anwendungsfall oft in der normalen Website-Entwicklung, insbesondere in Ergänzung zu bestehenden Technologien wie Apache/PHP- oder Rails-Stacks. Dies kann von der Implementierung von Echtzeit-Chats bis hin zur Überwachung bestehender Websites reichen. Aufgrund seiner Leichtigkeit und hohen Leistung wird Node.js oft als „Glue Service“ bezeichnet, der unterschiedliche Technologien nahtlos verbindet. Besonders interessant ist die Kombination von Node.js mit modernen Cloud-Infrastrukturen unter Verwendung von Tools wie Docker und Kubernetes oder plattformübergreifenden Funktionen wie AWS Lambda. Node.js eignet sich hervorragend, um umfangreiche Anwendungen in leicht implementierbare Microservices aufzuteilen, was zu einer effizienten und skalierbaren Architektur führt [5].

Um die Funktionsweise von Node.js besser zu verstehen, ist es wichtig, auch die negativen Aspekte zu betrachten. So führt das asynchrone Programmiermodell zu Problemen bei der Wartung des Codes. Obwohl das asynchrone Programmiermodell nützlich ist, um schnell auf Rückrufe zu reagieren, kann es problematisch sein, wenn mehrere Tasks im Hintergrund in der Warteschlange stehen. Dies kann sich auf die Qualität des Codes auswirken und seine Wartung erheblich erschweren. Ebenso ist Node.js nicht für rechenintensive Aufgaben geeignet. Bei einer schweren CPU-gesteuerten Rechenaufgabe in der Ereignisschleife wird der Großteil der CPU-Leistung für den Prozess verwendet. Dies verlangsamt die gesamte Ereignisschleife. Außerdem ist die Node.js-API aufgrund regelmäßiger Updates nicht stabil genug. Dies stellt ein Problem für die Kompatibilität des Codes dar, daher ist es zwingend notwendig, den Code an die neue Version anzupassen [4].

### 2.1.2 Angular

Angular ist das bevorzugte Framework für die Entwicklung reaktionsfähiger, einseitiger Webanwendungen und ästhetischer Website-Komponenten. Es ist eines der führenden JavaScript-Frameworks und wurde als umfassendes Framework mit Schwerpunkt auf Klarheit und Robustheit entwickelt. Entwickler schätzen die hervorragende Leistung von Angular bei der Erstellung dynamischer einseitiger Webanwendungen und die Unterstützung der MVC-Struktur. Angular hat sich aufgrund seiner Reaktionsfähigkeit und seines großen Funktionsumfangs schnell zu einer Schlüsseltechnologie in anspruchsvollen Unternehmensumgebungen entwickelt. Im Gegensatz zu anderen Frameworks, die einfach mit bestehenden Technologien gebündelt werden, was die Anwendungsentwicklung verkomplizieren kann, wurde Angular sorgfältig entwickelt. Dadurch wird sichergestellt, dass jede Funktion reibungslos funktioniert und zuverlässige Ergebnisse liefert [6].

Es ist wichtig zu wissen, dass Angular-Anwendungen aus NgModules bestehen, die als Container für einen Codeblock fungieren, der eine Anwendungsdomäne, einen Workflow oder einen eng verwandten Satz von Fähigkeiten abdeckt. NgModules stellen den Kompilierungskontext für ihren Inhalt bereit und können Komponenten, Dienstleister und andere Code-Dateien enthalten. Zudem können die Funktionalitäten von anderen Modulen exportiert und importiert werden [6].

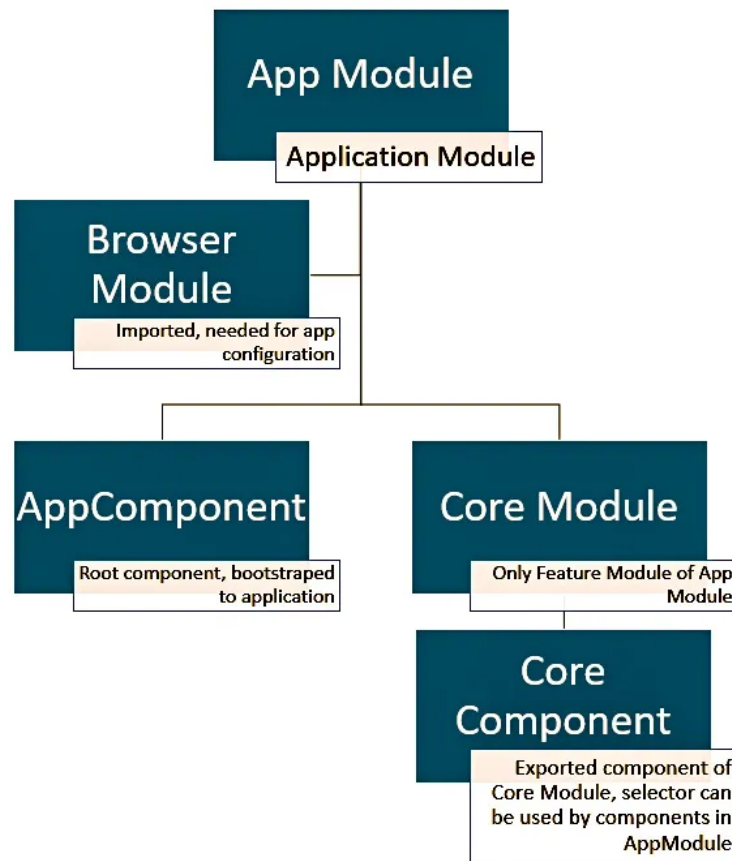


Abbildung 2.2: Angular-Modul Struktur[7]

Die oben gezeigte Abbildung gibt einen Überblick über die Komponenten der Angular-Modulstruktur. Es ist wichtig, die Architektur von Angular zu verstehen, die aus Feature-Modulen, Shared-Modulen und Core-Modulen besteht, da diese die wesentlichen Bausteine der Angular-Struktur darstellen.

- Feature Module

Die Strukturierung von Angular-Anwendungen mithilfe von Feature-Modulen ist ein effektiver Weg, um den Code übersichtlich zu halten und die Wartbarkeit zu verbessern. Jedes Feature-Modul fungiert als unabhängige Einheit, die eine bestimmte Funktionalität oder einen bestimmten Teil der Anwendung. Diese Module können als "Mini-Anwendungen" betrachtet werden, die innerhalb des größeren Rahmens der Programm funktionieren. Dank dieser modularen Struktur wird das Projekt in logische Abschnitte unterteilt, was die Entwicklung, das Testen und die Skalierung erleichtert. Darüber hinaus fördert die Verwendung von Funktionsmodulen die Wiederverwendung von Code, da Module und ihre Komponenten leicht in andere Projekte integriert werden können. Die klare Trennung der Funktionen in einzelne Module erleichtert auch die Zusammenarbeit im Team, da verschiedene Entwickler gleichzeitig an verschiedenen Teilen der Anwendung arbeiten können, ohne sich gegenseitig zu behindern. Insgesamt bieten die Funktionsmodule einen strukturierten und flexiblen Ansatz für die Entwicklung von Angular-Anwendungen, der es den Entwicklern ermöglicht, effektiv auf komplexe Anforderungen zu reagieren[7].

- Shared Module

Shared Module in Angular dienen als Sammlung von Elementen der Anwendung, die in verschiedenen Bereichen oder Funktionen verwendet und wiederverwendet werden sollen. Wenn eine Komponente, ein Service oder eine Pipe in verschiedenen Teilen der Anwendung wiederverwendet werden soll, empfiehlt es sich, diese in einem Shared Module zu deklarieren. Diese Zentralisierung gewährleistet Konsistenz und reduziert Redundanzen im Code. Im Gegensatz zu Feature-Modulen, die bestimmte Funktionsbereiche abdecken, sind Shared-Module nicht streng dem Bereichskonzept. Stattdessen die-

nen sie als Mittel zur Verteilung von gemeinsamen Elementen, die innerhalb von Feature-Modulen verwendet werden können, um deren Funktionalität zu erweitern. In der Regel werden in Shared Module neben wiederverwendbaren Komponenten auch Services und Pipes deklariert. Durch die Verwendung von Shared Module können Entwickler gemeinsame Funktionalitäten in einer Angular-Anwendung effektiv verwalten und pflegen. Dies fördert die Wiederverwendbarkeit von Code, Modularität und Skalierbarkeit in verschiedenen Teilen des Projekts. Dieser modulare Ansatz vereinfacht die Entwicklung, fördert die Konsistenz des Codes und erleichtert die Zusammenarbeit zwischen Teammitgliedern, die an verschiedenen Funktionen oder Bereichen der Anwendung arbeiten[7].

- Core Module

Das Core-Modul ist eine wichtige Komponente einer Angular-Anwendung, die anwendungsübergreifende Dienste und Komponenten bereitstellt. Seine Hauptaufgabe besteht darin, Singleton-Dienste auf Anwendungsebene bereitzustellen, die von verschiedenen Teilen der Anwendung benötigt werden. Dies können z.B. HTTP-Interceptoren zum Abfangen von HTTP-Anfragen, ein Logging-Service oder ein Service zur Verwaltung der angemeldeten Benutzer sein. In einigen Fällen kann das Core-Modul auch auf Funktionen oder Pipes aus dem Shared Module zugreifen, insbesondere wenn es sich um gemeinsame Funktionalitäten wie Formatierung handelt. Diese Verbindung wird in den Diagrammen durch gestrichelte Linien dargestellt, da sie nicht immer notwendig ist. Die Services im Core-Modul sind als Singleton konzipiert, d.h. sie werden nur einmal initialisiert und verwendet während der gesamten Laufzeit der Anwendung dieselbe Instanz. Um sicherzustellen, dass alle Module der Anwendung auf dieselben Instanzen dieser Services zugreifen, sollte das Core-Modul nur vom Root-App-Modul importiert werden. Würde ein lazy-loaded Modul das Core-Modul ebenfalls importieren, würde es seine eigenen Instanzen der Services erzeugen, was zu Inkonsistenzen führen könnte[8].

Ausschlaggebend für die Wahl von Angular für dieses Projekt war die Besonderheit, dass seine verschiedenen Elemente wie Komponenten, Direktiven, Services usw. eine standardisierte Verwendung erleichtern und in Module unterteilt werden können. Ebenso ermöglicht die Verwendung von Typescript die Verwendung von Datentypen, was Fehler reduziert und die Struktur übersichtlicher und leichter zu pflegen macht. Außerdem benötigt Angular mehr Code als die meisten Frameworks[9].

### 2.1.3 Ionic

Das Ionic-Framework ist ein quelloffenes, plattformübergreifendes Werkzeug, das verwendet werden kann, um Hybridanwendungen zu entwickeln, die problemlos auf verschiedenen mobilen Plattformen wie Android, iOS und Windows funktionieren. Das Ionic-Framework erleichtert die Entwicklung erheblich, indem es die Benutzeroberfläche der Hybridanwendung durch die Verwendung von HTML5 und CSS verbessert. Ionic stellt eine deutlich bessere Plattform für die Gestaltung und Entwicklung von Anwendungen dar, im Vergleich zu nativen Anwendungen. Letztere sind plattformspezifisch, erfordern jeweils spezifische Entwicklungswerkzeuge und sind sowohl zeit- als auch kostenintensiv. Im Gegensatz dazu ist Ionic plattformunabhängig und setzt auf die Verwendung von HTML5, CSS und Javascript. Der Entwicklungsprozess gestaltet sich zügig, und Entwickler haben einen direkten Zugang zu APIs über Cordova. Apache Cordova ist ein hybrides Open-Source-Framework, mit dem Webentwickler aus ihren HTML-, CSS- und JavaScript-Inhalten eine native Anwendung für eine Vielzahl mobiler Plattformen erstellen können[10]. Ionic macht zudem Gebrauch von den CSS-Eigenschaften der Syntactically Awesome Style Sheets (SASS).

Bei der Erkundung der Landschaft der Frameworks für die Entwicklung mobiler Anwendungen ist es wichtig, die tatsächliche Anwendung und Nutzung dieser Werkzeuge zu untersuchen. In diesem Zusammenhang wurde auf das Ionic Framework, einen wichtigen Akteur in diesem Bereich, hingewiesen. Durch die Analyse seiner Nutzungsstatistiken konnten wertvolle Erkenntnisse über seine Verbreitung und Relevanz im Ökosystem der mobilen Anwendungsentwicklung gewonnen werden.

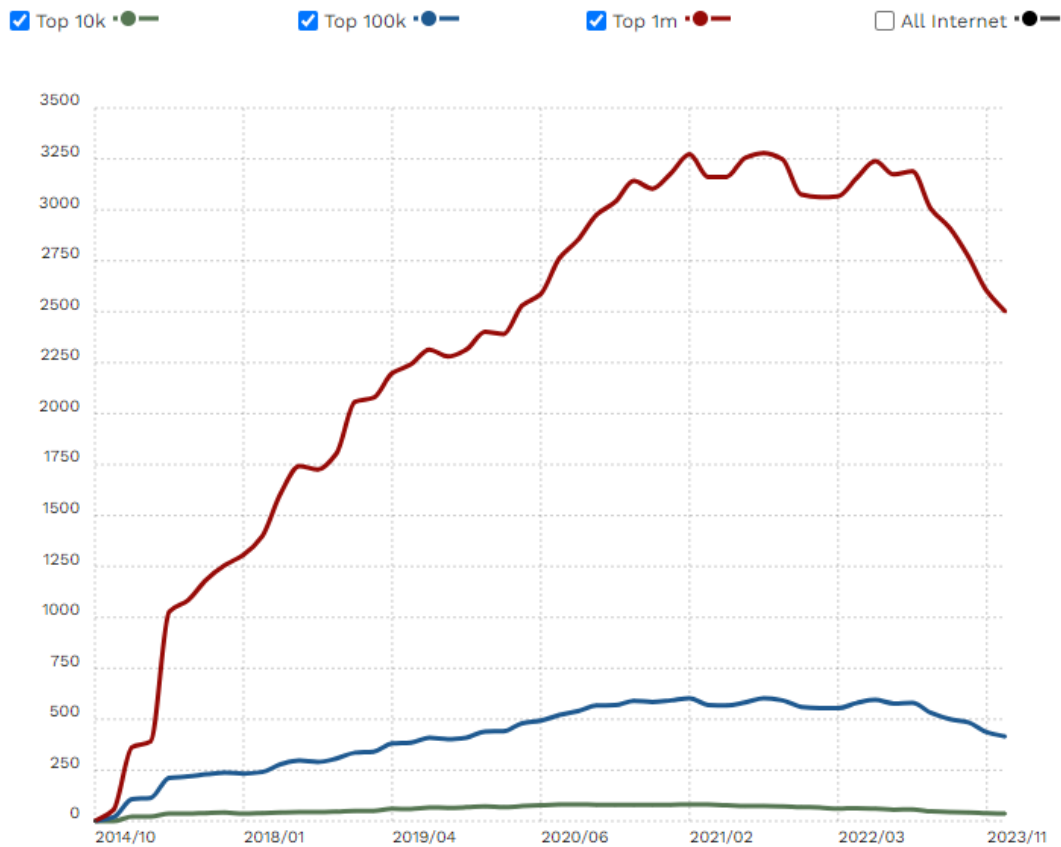


Abbildung 2.3: Ionic-Nutzungsstatistiken[11]

Aus der obigen Grafik geht hervor, dass die Nutzung des Ionic-Frameworks für die Entwicklung nativer und progressiver Webanwendungen seit seiner Einführung zugenommen hat. Dieser anhaltende Wachstumstrend spiegelt die Beliebtheit und Leistungsfähigkeit des Frameworks in der Entwicklergemeinschaft wider und wird nicht nur durch die kontinuierliche Weiterentwicklung und Verbesserung des Frameworks, sondern auch durch die steigende Nachfrage nach mobilen Lösungen in verschiedenen Branchen vorangetrieben.

Ionic bietet seinen Benutzern sämtliche Komponenten und Funktionen zur Verfügung, die auch bei der Entwicklung nativer mobiler Anwendungen, mithilfe von Software Development Kits (SDKs), genutzt werden. Entwickler können ihre Anwendungen mit Hilfe von Tools und Beispielcode erstellen, die in der Ionic-Framework-Dokumentation und auf der Support-Website zur Verfügung stehen. Die Installation erfordert NodeJS und npm, den Standard-Paketmanager für NodeJS, unabhängig von der Betriebsumgebung (Windows, Linux oder Macintosh). Ionic Creator ist eine Plattform zur Erstellung von Benutzeroberflächen durch einfaches Ziehen und Ablegen vorgefertigter Komponenten. Sie wurde implementiert, um das endgültige Erscheinungsbild der Anwendung zu gestalten. Ionic View ermöglicht es den Benutzern, die entwickelten Apps vor dem Hochladen in den offiziellen App Store einfach auf mehreren Geräten auszuführen und gemeinsam zu nutzen. Vor dem Hochladen in den offiziellen App Store kann Ionic Market verwendet werden, um mobile Anwendungen zu teilen, wobei für den Download eine Gebühr erhoben wird. Darüber hinaus bietet Ionic einen zuverlässigen Benutzersupport und eine Dokumentation mit nützlichen Informationen. Mit Ionic Lab können neu entwickelte Anwendungen auf Android- und iOS-Geräten getestet werden[12].

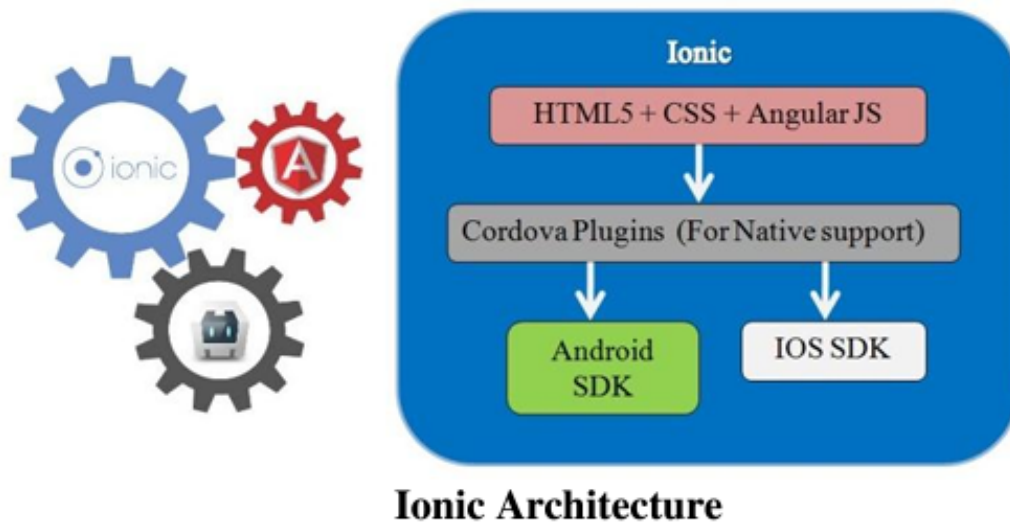


Abbildung 2.4: Ionic Architektur[12]

#### 2.1.4 Strapi

Strapi ist ein flexibles, kopfloses Content-Management-System (CMS), das mit Node.js entwickelt wurde. Es soll es Entwicklern ermöglichen, schnell APIs zu erstellen und anpassbare Content-Management-Lösungen zu entwickeln. Die Funktionen von Strapi bestehen aus einem Administrationsbereich, Inhaltstypen, Authentifizierung und rollenbasierter Zugriffskontrolle. Die Dokumentation behandelt verschiedene Aspekte der Erstellung von Anwendungen mit Strapi, einschließlich Installation, Konfiguration und Anpassung[13].

Hinter dem Wortspiel „Strapi“ verbirgt sich die Abkürzung für Bootstrap API. Wie der Name bereits verrät, hat es das Ziel, Entwicklern dabei zu helfen, eine API schnell aufzubauen. Strapi spart Zeit bei der Entwicklung von APIs durch sein integriertes, benutzerfreundliches Admin-Panel und bietet von Anfang an eine solide Auswahl an Kernfunktionen. Ob Backend-, Full-Stack- oder Frontend-Entwickler – mit Strapi gelingt der Einstieg in die API-Entwicklung mühelos, ohne das Rad neu erfinden oder Zeit mit der Implementierung grundlegender Funktionen wie Create, Read, Update und Delete (CRUD) oder Authentifizierung und Autorisierung verschwenden zu müssen[14].

Im Gegensatz zu traditionellen CMS-Anwendungen wie WordPress, bei denen Frontend und Backend eng miteinander verbunden sind und die Struktur der Inhalte sowie deren Präsentation festgelegt sind, ist ein Headless CMS vollständig von der Präsentationsschicht entkoppelt. Der Begriff „Headless“ bezieht sich auf die Trennung von Kopf (Frontend) und Körper (Backend). Das bedeutet, dass sich ein Headless CMS nicht darum kümmert, wie Inhalte präsentiert werden. Stattdessen bietet es einen inhaltszentrierten Ansatz mit einer API, um auf die Daten zuzugreifen und sie in jedem beliebigen Format anzuzeigen. Ein traditionelles CMS ist stark an die Darstellung gebunden, während ein Headless CMS mehr Flexibilität bietet[14]. Die genaue Darstellung der Unterschiede zwischen einem traditionellen CMS und einem Headless CMS ist der folgenden Abbildung zu entnehmen:

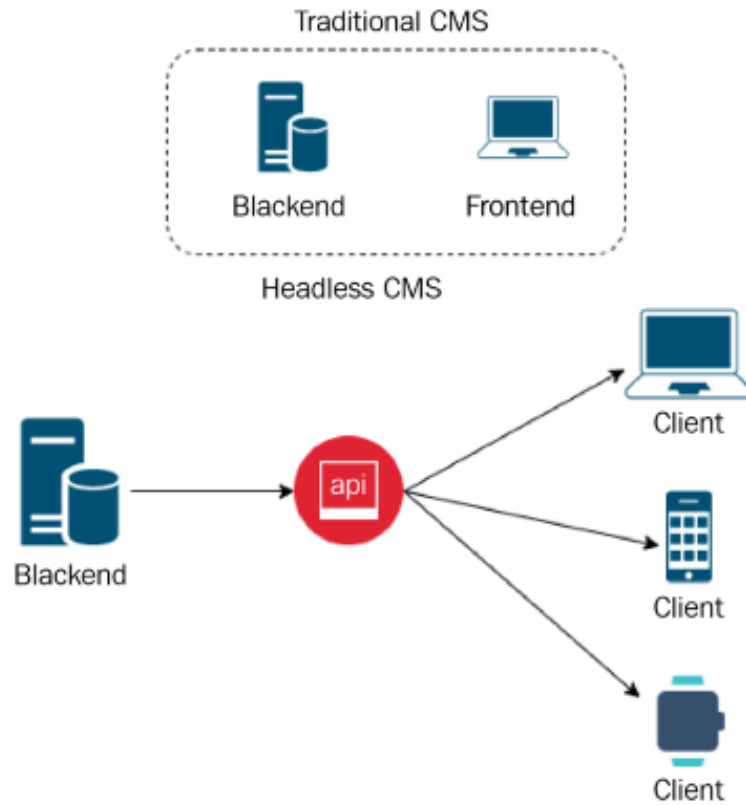


Abbildung 2.5: Traditional CMS versus headless CMS[14]

Strapi ist ein sogenanntes „Headless Content Management System“ (CMS), eine Softwareanwendung in der Webentwicklung, die es Benutzern ermöglicht, Inhalte zu erstellen, zu bearbeiten und zu veröffentlichen. Im Gegensatz zu herkömmlichen CMS-Systemen ist ein Headless CMS unabhängig von der Darstellung der Inhalte und verfolgt stattdessen einen inhaltsorientierten Ansatz. Es bietet eine API zum Zugriff und zur Anzeige von Daten in beliebigen Formaten[14].

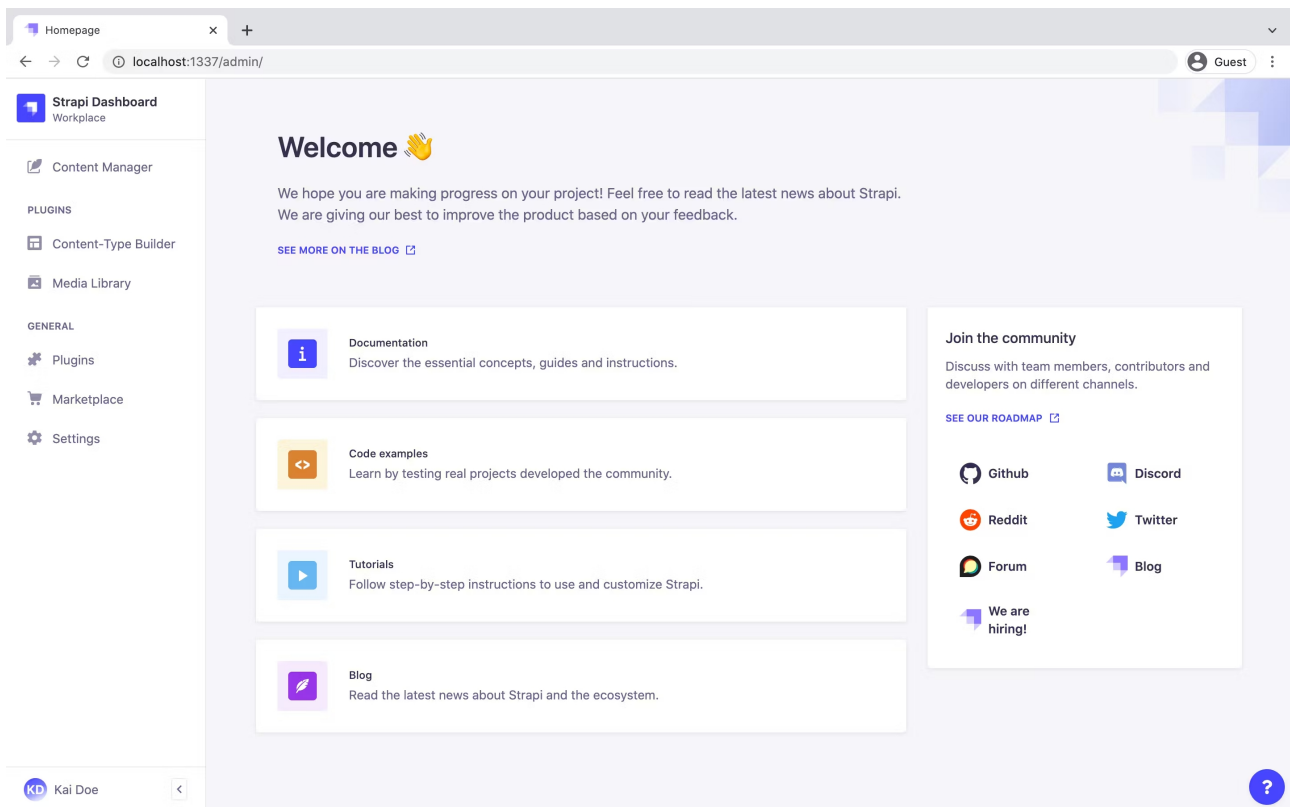


Abbildung 2.6: Strapi Content Management System[15]

Zudem gehören API-Stile RESTful und GraphQL, die von Strapi unterstützt werden. Strapi bietet eine sofort einsatzbereite RESTful-API, die auf dem Prinzip der Representational State Transfer (REST) basiert. Diese API kann von verschiedenen Web-Clients wie React, Angular, Vue.js, sowie von mobilen Anwendungen und sogar Internet of Things (IoT)-Anwendungen verwendet werden. Darüber hinaus unterstützt Strapi auch GraphQL über ein Plugin. Dies ermöglicht Entwicklern die Flexibilität, zwischen RESTful und GraphQL je nach den Anforderungen ihres Projekts zu wählen[14].

Die Entscheidung für die Verwendung der REST-API basiert auf der Tatsache, dass RESTful in Strapi einen transparenten Zugriff auf Inhaltstypen über dedizierte API-Endpunkte bietet und somit Entwicklern eine einfache Möglichkeit zur Interaktion mit Daten bereitstellt. Ein wesentlicher Vorteil von Strapi besteht darin, dass bei der Erstellung von Inhaltstypen automatisch API-Endpunkte generiert werden, was die manuelle Konfiguration von Endpunkten überflüssig macht. Dieser automatisierte Prozess vereinfacht die Entwicklung und beschleunigt die Markteinführung von Projekten. Darüber hinaus unterstützt die REST-API die Verwendung von API-Parametern, wodurch Entwickler API-Abfragen verfeinern und an spezifische Anforderungen anpassen können. Diese Parameter dienen Entwicklern dazu, Ergebnisse zu filtern, zu sortieren und zu paginieren, was eine fein abgestimmte Steuerung über die aus den API-Endpunkten abgerufenen Daten bietet. Insgesamt vereinfacht die REST-API in Strapi den Prozess des Aufbaus robuster und skalierbarer Anwendungen, indem sie einen intuitiven Zugriff auf Inhaltstypen bietet und Entwickler mit leistungsstarken Abfragefähigkeiten ausstattet[16].

Die GraphQL-API in Strapi bietet eine leistungsstarke Möglichkeit, mit Inhaltstypen über das GraphQL-Plugin von Strapi zu interagieren, wodurch Entwickler Abfragen und Mutationen mühelos durchführen können. Diese API erlaubt Entwicklern, die Struktur ihrer Abfragen zu definieren, um genau die benötigten Daten abzurufen und damit das Über- oder Unterabfragen von Daten zu reduzieren, das häufig bei traditionellen REST-APIs auftritt. Mit GraphQL können Programmierer komplexe Abfragen ausführen, um verschachtelte Daten abzurufen, Ergebnisse aus mehreren Quellen zusammenzuführen und Mutationen auszuführen, um Daten auf flexible und intuitive Weise zu ändern. Darüber hinaus unterstützt die GraphQL-API in Strapi die Filterung, Sortierung und Paginierung von Ergebnissen, wodurch Entwickler eine feine abgestimmte Steue-



rung über die aus API-Anfragen zurückgegebenen Daten erhalten. Diese Funktion gewährleistet eine optimale Leistung und Effizienz bei der Arbeit mit großen Datensätzen. Insgesamt bietet die GraphQL-API in Strapi einen modernen und effizienten Ansatz für das Abrufen und Manipulieren von Daten, was die Produktivität der Entwickler steigert und die Erstellung leistungsstarker Anwendungen ermöglicht[17].

## 2.2 Technologie und Sprachen

### 2.2.1 HTML

HTML, auch bekannt als Hypertext Markup Language, dient der Strukturierung von Webseiten. Eine typische HTML-Seite besteht aus zwei Hauptteilen: dem Kopfbereich (Head) und dem Inhaltsbereich (Body). Der Kopfbereich enthält Metadaten, also zusätzliche Informationen, die für den Benutzer nicht direkt sichtbar sind. Hier können z.B. der Zeichensatz oder die Viewport-Einstellungen definiert werden. Der Viewport ist besonders wichtig, um sicherzustellen, dass die Inhalte auf verschiedenen Bildschirmgrößen gut lesbar und korrekt dargestellt werden[18].

```
1  <!doctype html>
2  <html lang="de">
3
4      <head>
5          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6          <title>Titre de la page</title>
7          <link rel="stylesheet" type="text/css" href= "chemin/feuille.css">
8      </head>
9      <body>
10     </body>
11 </html>
12
```

Abbildung 2.7: HTML Architektur [19]

Für einen detaillierten Überblick über die grundlegenden Prinzipien und Strukturen von HTML zeigt die untenstehende Abbildung die Architektur von HTML. Zu den wesentlichen Komponenten gehört unter anderem:

- <!DOCTYPE>

Dieses Element gibt den Dokumententyp und die HTML-Version an. Es hilft dem Browser, die Regeln und Standards zu verstehen, die er beim Rendern der Seite befolgen muss[19].

- <head>

Dieser Bereich enthält Metainformationen über das Dokument, die nicht direkt auf der Webseite angezeigt werden. Sie können hier den Seitentitel, Meta-Tags für die Suchmaschinenoptimierung (SEO), externe Stylesheets, Skripte usw. unterbringen[19].

- <body>

Hier wird der eigentliche Inhalt der Webseite abgelegt, der vom Browser gerendert und angezeigt wird.

### 2.2.2 TypeScript

TypeScript ist eine Erweiterung von JavaScript, die darauf abzielt, die Entwicklung umfangreicher JavaScript-Anwendungen zu vereinfachen. In Bezug auf die TypeScript-Syntax handelt es sich um eine Weiterentwicklung von EcmaScript 5, die es jedem JavaScript-Programm ermöglicht, gleichzeitig als TypeScript-Programm

zu fungieren. TypeScript bereichert JavaScript um ein Modulsystem, Klassen, Schnittstellen und ein statisches Typsystem. Das Modulsystem und das Typensystem von TypeScript sind darauf ausgelegt, Programmierern eine leichte Unterstützung zu bieten und zeichnen sich durch ihre Flexibilität und Benutzerfreundlichkeit aus. Der TypeScript-Compiler überwacht TypeScript-Programme und konvertiert sie in JavaScript. Dadurch können Programme sofort in verschiedenen Ausführungsumgebungen ausgeführt werden. Microsoft verwendet den Compiler häufig zur Entwicklung umfangreicher JavaScript-Anwendungen. Ein nettes Feature von TypeScript ist, dass sein Typsystem nicht von Natur aus statisch ist. Das bedeutet, dass es von Grund auf darauf ausgelegt ist, sich flexibel an unterschiedliche Anforderungen anzupassen[20].

### 2.2.3 SASS

Sass, ein Akronym für „Syntactically Awesome Style Sheets“, ist eine leistungsstarke Skriptsprache, die entwickelt wurde, um die Fähigkeiten von Cascading Style Sheets (CSS) zu erweitern. Als CSS-Präprozessor führt Sass erweiterte Funktionen und Konstrukte ein, die über die nativen Fähigkeiten von Standard-CSS hinausgehen. Der Hauptzweck von Sass besteht darin, den Prozess des Schreibens und Pflegens von Stylesheets für die Webentwicklung zu rationalisieren und zu vereinfachen. Um Sass in einem Webentwicklungsprojekt zu verwenden, verwenden Entwickler normalerweise einen Compiler, um Sass-Code in Standard-CSS zu übersetzen, das der Browser interpretieren und anzeigen können. Diese zusätzliche Abstraktionsebene vereinfacht den Styling-Prozess und trägt zu besser wartbarem und effizienterem Code bei[21].

### 2.2.4 MySQL

Die Abkürzung „MySQL“ steht für „Structured Query Language“, die standardisierte Sprache für den Datenbankzugriff. MySQL ist eine neue Version der Open-Source-SQL-Datenbank, die Teil der Oracle Corporation ist und in einer Version verfügbar ist. Dabei handelt es sich um die Struktur der Daten, die letzten Hinweise, die Kanten und die Richtigkeit der Peilungen. Mithilfe der tabellenbasierten Speicherung organisierter MySQL-Datenbankstrukturen und optimierter physischer Dateien wurde die Datenverarbeitungsgeschwindigkeit verbessert. Das Logistikkmodell von MySQL umfasst Elemente mit Datumsangaben, Tabulatoren, Ansichten, Zeilen und Spalten und verfügt über eine flexible Programmierdatenbank. MySQL steht unter der GPL (GNU General Public License) und ist eine weitere Open-Source-Software, die auf transparente und gemeinschaftliche Weise unter den Leistungsbeschreibungen veröffentlicht wurde[22].

Standardversionen werden vom SQLite-Datenbank-Client bereitgestellt. Für dieses Projekt wurde jedoch entschieden, stattdessen MySQL zu verwenden. Daher wurde Strapi so konfiguriert, dass MySQL als Datenbankclient verwendet wird. Diese Entscheidung wurde aufgrund der spezifischen Anforderungen des Projekts getroffen, da MySQL im Vergleich zu SQLite Vorteile in Bezug auf Skalierbarkeit, Leistung und Datenverwaltung bietet. Die Integration von MySQL in Strapi ermöglichte eine effiziente Datenverwaltung in einem robusten relationalen Datenbank-Framework, was das Projekt weitgehend erfüllte.

## 2.3 Bibliotheken, Software, Betriebssysteme, IDE und Plugins

### 2.3.1 Visual-Studio-Code

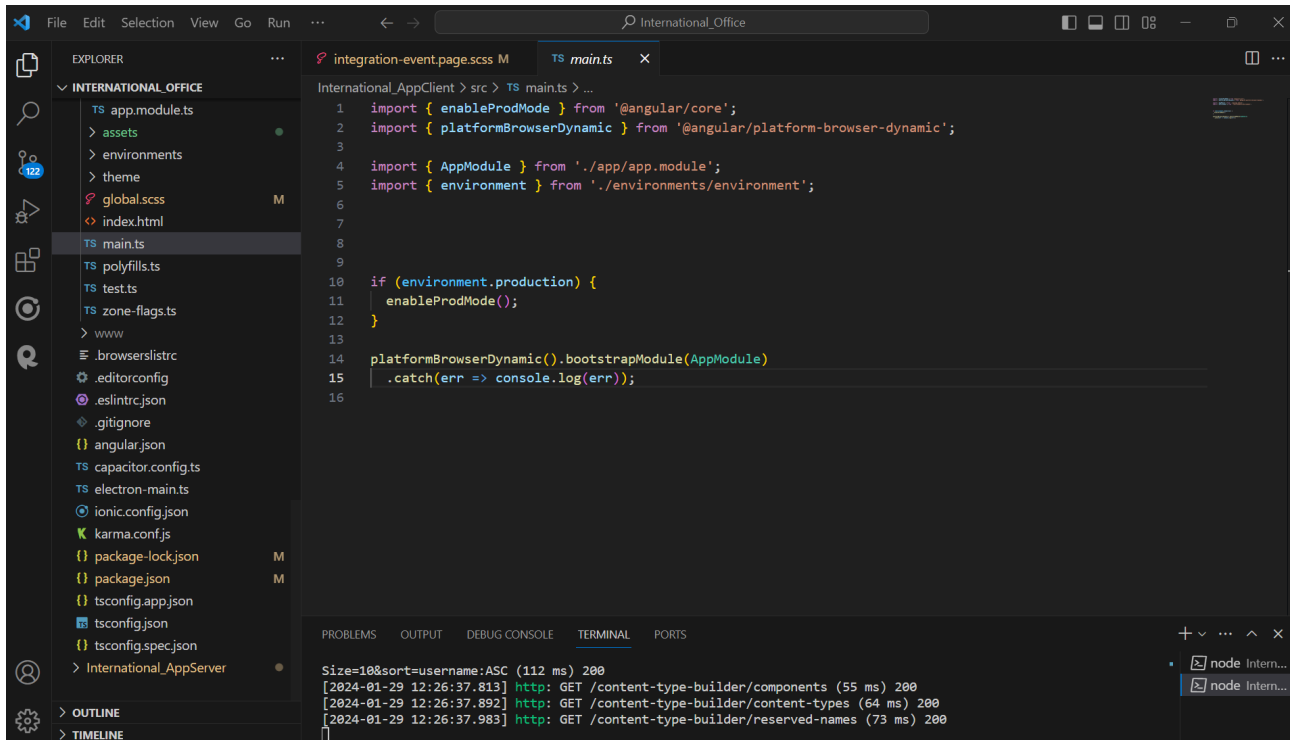


Abbildung 2.8: Visual Studio Code  
Quelle: Eigene Grafik

Visual Studio Code (VS Code) ist das erste plattformübergreifende Entwicklungswerkzeug innerhalb der Microsoft Visual Studio-Familie, das Windows-, Linux- und macOS-Nutzer erreicht. Dieses quelloffene und kostenlose Tool ist in erster Linie auf Code ausgerichtet und erleichtert die Bearbeitung von Code-Dateien und die Verwaltung von ordnerbasierten Projektstrukturen. Neben der Vereinfachung dieser grundlegenden Aufgaben dient Visual Studio Code als Werkzeug für die Erstellung cross-plattform Anwendungen für Web-, Mobil- und Cloud-Umgebungen und unterstützt beliebte Plattformen wie Node.js und .NET 5 (einschließlich früherer Versionen wie .NET Core). Seine Vielseitigkeit wird durch die integrierte Unterstützung für eine Vielzahl von Programmiersprachen und eine Reihe von erweiterten Bearbeitungsfunktionen unterstrichen, darunter IntelliSense, Symbolreferenzidentifizierung und schnelle Typdefinitionssuche [23].

Visual Studio Code basiert auf dem Electron-Framework und macht sich die Leistungsfähigkeit nativer Technologien zunutze, um plattformunabhängige Anwendungen zu erstellen. Das Ergebnis ist ein Tool, das die Einfachheit eines robusten Code-Editors nahtlos mit den wesentlichen Tools verbindet, die für die Unterstützung des gesamten Lebenszyklus der Anwendungsentwicklung entscheidend sind. Visual Studio Code enthält Debugger und bietet eine Versionskontrollintegration auf der Basis von Git, was es zu einem ganzheitlichen Entwicklungstool und nicht zu einem reinen Code-Editor macht. Für Entwickler, die eine umfassendere Entwicklungserfahrung suchen, wäre es ratsam, sich für Microsoft Visual Studio 2019 für Windows-Benutzer oder Visual Studio 2019 für Mac-Benutzer zu entscheiden. Visual Studio Code erweist sich in verschiedenen Szenarien als unschätzbar wertvoll und bietet einen agilen, fokussierten Ansatz für Codierung und Projektmanagement. [23].

### 2.3.2 Angular-Material

Angular Material ist nicht nur eine Sammlung von hochwertigen Komponenten für Angular-Webanwendungen, sondern auch ein umfassendes Toolkit, das eine Reihe von leistungsstarken Funktionen bietet. Neben der Bereitstellung von benutzerfreundlichen und barrierefreien Komponenten legt Angular Material auch einen

starken Fokus auf Internationalisierung und ermöglicht so die Erstellung von Anwendungen, die problemlos in verschiedenen Sprachen und Regionen eingesetzt werden können[24]. Darüber hinaus bietet Angular Material Tools zur einfachen Anpassung und Thematisierung von Material-Design-Komponenten, sodass Entwickler die visuelle Darstellung ihrer Anwendungen ganz nach ihren Anforderungen gestalten können. Dies ermöglicht eine konsistente und ansprechende Benutzererfahrung über die gesamte Anwendung hinweg. Ein weiterer wichtiger Aspekt von Angular Material ist seine Fähigkeit, universelles Verhalten zu unterstützen. Dies bedeutet, dass Anwendungen, die mit Angular Material entwickelt wurden, auf verschiedenen Geräten und Betriebssystemen einheitlich und zuverlässig funktionieren, was die Entwicklung und Wartung erleichtert[24].

## 3 Theoretischer Grundlagen

### 3.1 Internationales Hochschulmanagement

Das International Office ist die zentrale Anlaufstelle für Studierende, die sich für ein Auslandssemester interessieren. Hier erhalten sie umfassende Beratung und Unterstützung bei der Planung und Organisation Ihres Studienaufenthaltes im Ausland. Das Büro bietet Unterstützung bei administrativen Abläufen sowie in den verschiedenen Fachbereichen, um eine reibungslose Integration und Orientierung der Studierenden zu gewährleisten.

#### 3.1.1 Prinzipien

Ein zentrales Prinzip ist die Nutzerorientierung. Das bedeutet, dass die Bedürfnisse, Erwartungen und Anforderungen der Nutzer im Mittelpunkt des Designs stehen müssen. Die Anwendung sollte intuitiv und benutzerfreundlich gestaltet sein, um eine reibungslose Interaktion und ein positives Nutzererlebnis zu gewährleisten.

Ein weiteres wichtiges Prinzip ist die Skalierbarkeit und Flexibilität. Die Anwendung muss so konzipiert sein, dass sie mit den wachsenden Anforderungen des International Office mitwächst und gleichzeitig flexibel genug ist, um auf Änderungen und neue Anforderungen reagieren zu können. Dies erfordert eine modulare und gut strukturierte Architektur. Sicherheit und Datenschutz sind ebenfalls wichtige Prinzipien, die bei der Entwicklung einer Anwendung für ein International Office berücksichtigt werden müssen. Aufgrund der Sensibilität der bereitgestellten Informationen ist es wichtig, robuste Sicherheitsmaßnahmen zu implementieren, um die Vertraulichkeit, Integrität und Verfügbarkeit der Daten zu gewährleisten. Dazu gehören Aspekte wie sichere Datenübertragung, Benutzerauthentifizierung und -autorisierung sowie der Schutz vor gängigen Sicherheitsbedrohungen wie Datenlecks oder unberechtigt Zugriff.

Neben diesen Prinzipien ist auch die Barrierefreiheit ein wichtiger Aspekt, der beim Design der Anwendung berücksichtigt werden muss. Die Anwendung muss für die Zielgruppe unabhängig von ihren individuellen Fähigkeiten oder Einschränkungen zugänglich sein. Dies erfordert die Einhaltung von Zugänglichkeitsstandards und -richtlinien sowie die Implementierung von Funktionen wie Alternativtext für Bilder, Tastaturzugänglichkeit und Unterstützung beim Lesen von Bildschirminhalten.

#### 3.1.2 Aufgaben des International Office

Das International Office bietet umfassende Unterstützung für internationale Studierende, Stipendiaten und Gäste. Dazu gehören nicht nur administrative Fragen wie Visum und Unterkunft, sondern auch die Orientierung im neuen Studien- und Lebensumfeld. Ziel ist es, den internationalen Gästen einen guten Start an der Hochschule und in der Stadt zu ermöglichen.

Ein weiterer wichtiger Aufgabenbereich des International Office ist die Verwaltung und Pflege der europäischen Partnerschaften im Rahmen des ERASMUS-Programms. Dazu gehören die Koordination von Austauschprogrammen, die Förderung der Mobilität zwischen europäischen Hochschulen und die Sicherstellung einer effektiven Zusammenarbeit.

Darüber hinaus spielt das International Office eine entscheidende Rolle bei der Koordination von Hochschulpartnerschaften weltweit. Dazu gehören die Planung internationaler Projekte, die Förderung von Forschungskooperationen und der Aufbau strategischer Allianzen mit Hochschulen in verschiedenen Ländern.

### 3.1.3 Orientierungswoche für Austauschstudenten

Die Orientierungswochen markieren den Beginn einer erfolgreichen akademischen und persönlichen Erfahrung für internationale Studierende in Deggendorf. Die zwei Orientierungswochen wurden sorgfältig konzipiert, um den Studierenden einen reibungslosen Übergang in ihre neue Umgebung zu gewährleisten und die vielen Herausforderungen der Studienzeit zu meistern.

Die Orientierungswochen dienen nicht nur als formaler Auftakt, sondern schaffen auch eine ermutigende und herzliche Umgebung für internationale Studierende. Dadurch ist es möglich, die akademische Laufbahn in Deggendorf nicht nur erfolgreich zu starten, sondern auch in einer Atmosphäre der Unterstützung und Gastfreundschaft zu erleben.

### 3.1.4 Herausforderungen bei der Betreuung von Austauschstudenten

Die Betreuung von Austauschstudenten ist eine anspruchsvolle Aufgabe, die vielfältige Herausforderungen mit sich bringt. Einer der zentralen Aspekte ist die kulturelle Anpassung der Studierenden an ihre neue Umgebung. Unterschiede in der Sprache, im Lebensstil und in sozialen Normen können zu Anfangsschwierigkeiten führen, die einer sorgfältigen Betreuung bedürfen.

## **4 Anforderungen**

In diesem Kapitel wird das Ziel der Arbeit näher erläutert, um das Konzept entsprechend anzupassen und den Erfolg der Umsetzung nach Abschluss des Projekts zu bewerten.

### **4.1 Darstellung der Orientierungsweek Zeitplan**

Die Funktionalität soll die Bereitstellung von Daten, Zeitplänen und andere wichtige Informationen zu den Orientierungswochen für die Studenten umfassen.

### **4.2 Bereitstellung von Dokumenten zum Herunterladen**

Austauschstudierenden sollen über die Anwendung die Möglichkeit haben, wichtige Dokumente herunterzuladen, welche für ihren Studienaufenthalt relevant sind.

### **4.3 Das Abholservice**

Ein Service-Formular für die Eingabe von Reiseinformationen der Studierenden sollte bereitgestellt werden, um Abholungen vom Flughafen oder Bahnhof zu organisieren.

### **4.4 Sprachenvielfalt**

Die Anwendung soll zweisprachig konzipiert und in Englisch und Deutsch verfügbar sein, um sicherzustellen, dass Benutzer die Anwendung unabhängig von ihrer Muttersprache problemlos nutzen können.

### **4.5 Präsentation der häufig gestellten Fragen (FAQ)**

Es soll eine übersichtliche Darstellung der häufig gestellten Fragen (FAQ) gestellt, um Nutzern schnell und einfach Antworten auf ihre Anliegen zu liefern.

### **4.6 Darstellung einer Checkliste**

Es sollte ein Formular in Form einer Checkliste geben, damit Studenten wissen, welche Dokumente sie für jegliche administrative Dienstleistungen in der Stadt Deggendorf benötigen.

### **4.7 Die Benachrichtigung für den Kurs-Deadline-Reminder**

Die Applikation soll eine Nachricht an Studierende senden, um sie an die Registrierungsfrist für Kurse zu erinnern.

## 4.8 Zielplattformen

Das Projekt soll als mobile Anwendung konzipiert, die auf verschiedenen Zielplattformen verfügbar sein soll.



## 5 Konzeption und Design

Angesichts der Anforderungen liegt nun der Fokus darauf, ein Konzept zu erstellen, das sämtliche erforderlichen Funktionen und Qualitätsstandards des Systems umsetzt. Die angestrebte Anwendung sollte als hybride App realisiert werden, welche von einer einzigen Codebasis aus sowohl auf Android- und iOS-Geräten als auch als Progressive Web-App und Desktop-Web-Anwendung genutzt werden kann.

Der gegenwärtige Trend in der Anwendungsentwicklung tendiert klar zu mobilen Geräten. Daher steigt die Nachfrage nach Multiplattform-Lösungen, die nicht nur im Web, sondern auch auf mobilen Endgeräten als iOS- und Android-Apps funktionieren können[25].

### 5.1 Architektur der Anwendung

#### 5.1.1 Mobile Anwendung

Eine mobile Anwendung ist eine Software, die für die Ausführung auf tragbaren Geräten wie Smartphones oder Tablets entwickelt wurde und die Möglichkeit bietet, eine Verbindung zu drahtlosen Netzwerken herzustellen. Viele davon können im App Store des Geräts vorinstalliert oder aus dem Internet heruntergeladen oder von Benutzern selbst erstellt werden[26]. Derzeit stehen für die Entwicklung mobiler Apps drei Optionen zur Verfügung: native Apps, Web-Apps und Hybrid-Apps. Die Abbildung zeigt die verschiedenen Arten von Tools, Techniken und Programmiersprachen für die Entwicklung mobiler Anwendungen.

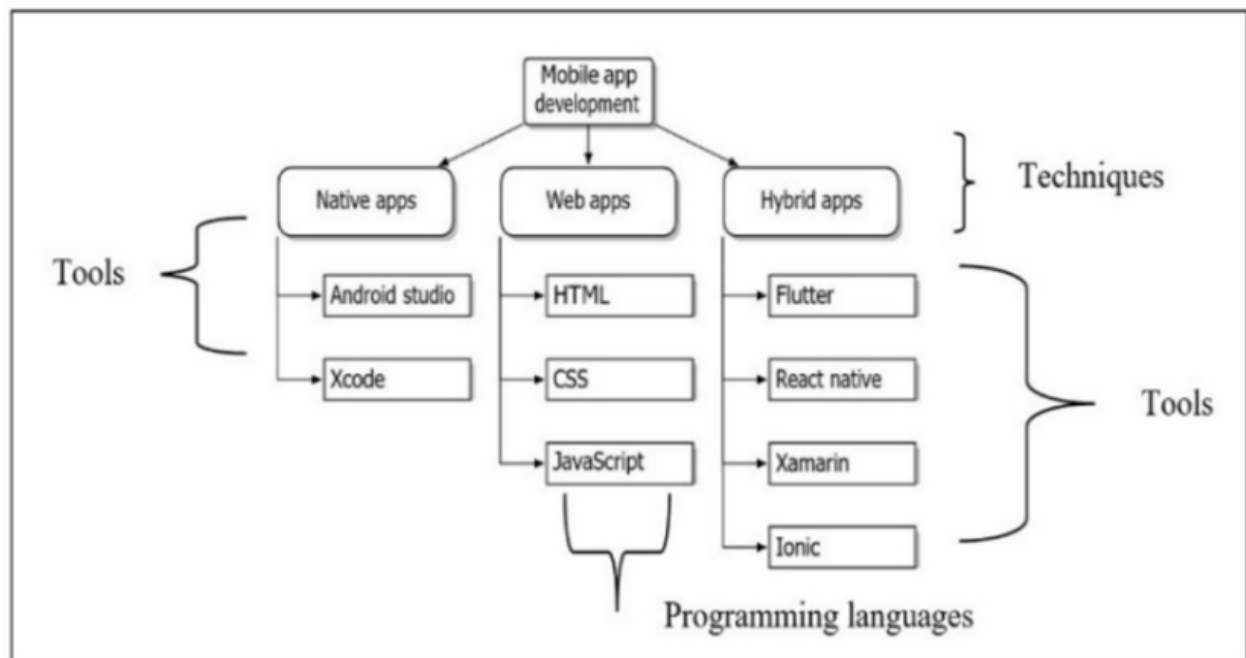


Abbildung 5.1: Mobile Anwendungsentwicklung[27]

#### Hybrid Mobile App

Eine hybride Anwendung kombiniert die Flexibilität von Webtechnologien mit der Funktionalität nativer mobiler Anwendungen. Sie läuft innerhalb eines nativen Wrappers, der oft als WebView bezeichnet wird und die

Interaktion zwischen den webbasierten Komponenten und den nativen Funktionen des Geräts erleichtert[28].

Hybrid-Apps basieren auf Webtechnologien wie HTML5, CSS3 und JavaScript sowie Plugins, um auf die API-Plattform des Geräts zuzugreifen. Sie verwenden einen nativen Wrapper, um Webcode zu kapseln und eine generische JavaScript-API bereitzustellen, um Serviceanfragen zwischen Webcode und der Plattform-API zu überbrücken. Dieser native Wrapper ermöglicht die Verpackung, Bereitstellung und plattformübergreifende Verteilung hybrider mobiler Anwendungen[29].

Die Entwicklung hybrider mobiler Apps erfordert Frameworks wie Apache Cordova, Ionic und ähnliches, um plattformübergreifende Apps mit nativer Funktionalität zu erstellen. Hybride App-Entwicklungsplattformen werden immer beliebter, da sie eine geräteunabhängige App-Entwicklung ermöglichen. Hybride Apps können auf verschiedenen Plattformen ausgeführt werden, sei es auf einem Desktop (Windows, Linux, Mac OS) oder in einem mobilen Browser. Sie können auch die von uns verwendeten mobilen Apps und Progressive Web Apps (PWA) verwenden[29].

Hybride Anwendungen bieten weitere Vorteile. Da der Code nur einmal geschrieben und auf mehreren Plattformen bereitgestellt werden muss, reduzieren sich Entwicklungszeit und -aufwand. Außerdem können Entwickler ihre vorhandenen Web-Entwicklungskenntnisse nutzen, ohne eine neue Programmiersprache erlernen zu müssen. Im Gegensatz zu reinen Web-Apps greifen hybride Apps über Plugins auf native Gerätefunktionen zu und erweitern so ihren Funktionsumfang.

Dieser Ansatz hat jedoch auch Nachteile. Die Performance kann je nach Browsereinstellungen variieren. Die Verbindung zwischen WebView und der nativen Funktionalität, die durch die Plugins bereitgestellt wird, wird durch Abhängigkeiten und mögliche Einschränkungen beim Zugriff auf bestimmte Geräte-APIs ermöglicht. Darüber hinaus fehlt es hybriden Anwendungen an nativen UI-Komponenten, was Entwickler dazu zwingt, die gesamte UI mit Webtechnologien zu entwerfen.

<b>Platform</b>	<b>Native Programming languages</b>
iOS	Objective-C, Swift
Android	Java
Windows Phone	C#, C++
Symbian	C++
BlackBerry	C#

Abbildung 5.2: Hybride App[29]

### Native Mobile App

Native Anwendungen werden mit den spezifischen Programmiersprachen des jeweiligen Betriebssystems entwickelt, für das sie bestimmt sind. Die Entwicklung mobiler Anwendungen für verschiedene Geräte mit unterschiedlichen Betriebssystemen (OS) ist mit zusätzlichen Ressourcenkosten und erhöhten Entwicklungsherausforderungen verbunden. Dies ist darauf zurückzuführen, dass Programmierer Kenntnisse in verschiedenen Programmiersprachen erwerben müssen, um den Anforderungen der verschiedenen Plattformen gerecht zu werden[29].

Diese Art der Entwicklung bietet einige klare Vorteile gegenüber anderen Ansätzen. Erstens sind die Entwicklungswerkzeuge eng in die Geräteplattform integriert. Softwareentwickler können in IDEs arbeiten, die speziell für die jeweilige Plattform konfiguriert sind, wie zum Beispiel Xcode für iOS oder Android Studio für Android. Andererseits stehen alle nativen APIs und Funktionen zur Verfügung, da direkt im nativen Framework entwickelt wird. Dies garantiert eine optimale Performance ohne zusätzliche Umwege, da die App direkt auf dem Gerät läuft, ohne Zwischenschichten im Code, die die Performance beeinträchtigen könnten [28].

Der größte Nachteil der nativen App-Entwicklung sind jedoch die sprachlichen Herausforderungen. Eine App

muss oft für iOS, Android (und möglicherweise auch Windows) entwickelt werden, was umfangreiche Kenntnisse in verschiedenen Programmiersprachen und APIs erfordert. Da der Clientcode nicht wiederverwendet werden kann, muss er für jede Plattform neu geschrieben werden. Hinzu kommt der technische Aufwand für die Pflege mehrerer Codebasen.[28].

Die Entwicklung solcher Anwendungen ist sehr anspruchsvoll. Diese Softwareentwicklungsplattformen umfassen Software Development Kits (SDKs), die Softwareprogramme einschließlich Softwareentwicklungsanleitungen enthalten.

	<b>iOS</b>	<b>Android</b>	<b>Windows</b>
<b>Development Tool</b>	X-code	Eclipse	Visual Studio
<b>Programming Language</b>	Objective-C Swift	Java Code-Base	C#, C++

Abbildung 5.3: Native App[29]

### Mobile Web App

Bei mobilen Webanwendungen handelt es sich um Anwendungen, die über den Webbrowser auf mobile Geräte geladen werden können. Im Gegensatz zu nativen Apps, die speziell für eine bestimmte Plattform wie iOS oder Android entwickelt wurden, basieren mobile Web-Apps auf Webtechnologien wie HTML, CSS und JavaScript. Diese Anwendungen bieten dem Nutzer ein ähnliches Benutzererlebnis wie native Anwendungen, können aber über den Browser auf verschiedenen Geräten und Plattformen ausgeführt werden[28].

Ein großer Vorteil mobiler Webanwendungen ist ihre breite Verfügbarkeit. Sie können auf nahezu jedem Gerät mit einem Webbrowser ausgeführt werden, unabhängig vom Betriebssystem. Dies erhöht die Reichweite Ihrer App, da Sie nicht auf bestimmte Plattformen beschränkt sind. Darüber hinaus sind mobile Web-App oft einfacher und kostengünstiger zu entwickeln und zu aktualisieren als native Apps. Da sie auf Webtechnologien basieren, können Entwickler ihre vorhandenen Fähigkeiten nutzen und müssen nicht verschiedene Programmiersprachen oder Entwicklungsplattformen beherrschen[28].

Mobile Web Apps haben aber auch einige Nachteile. Sie können nicht wie native Apps auf alle Funktionen und die Hardware des Geräts zugreifen. Darüber hinaus kann die Leistung einer mobilen Web-App von der Leistung des Webbrowsers und des Geräts abhängen, was zu einer weniger reibungslosen Nutzererfahrung führen kann[28].

Um sich in der Landschaft der Softwareentwicklung zurechtzufinden, ist es unerlässlich, eine Vielzahl von technischen Kriterien zu berücksichtigen, um den Entscheidungsprozess effizient zu steuern. Diese Faktoren dienen als unverzichtbarer Kompass bei der Suche nach dem am besten geeigneten Entwicklungsansatz unter den vielen verfügbaren Optionen. Durch die sorgfältige Bewertung von Elementen wie Plattform- und Versionsunterstützung, Gerätekapazitäten, Benutzererfahrung, Leistung und Aufrüstbarkeit kann jede Alternative systematisch bewertet werden. Dieser methodische Ansatz ermöglicht fundierte Entscheidungen und stellt sicher, dass die getroffene Wahl genau den spezifischen Anforderungen und Zielen des jeweiligen Projekts entspricht.

Zunächst ist es sinnvoll, die zu unterstützenden Plattformen und Versionen sowie die Bandbreite der Geräte und Browserfähigkeiten zu evaluieren. Ein hybrider oder mobiler Web-Ansatz kann für die Entwicklung von Anwendungen für mehrere Plattformen besser geeignet sein, während native Anwendungen eine stärker personalisierte Erfahrung für jede Plattform bieten. Darüber hinaus beeinflussen die erforderlichen Gerätefunktionen, wie z. B. Kamerazugriff oder Bluetooth, die Wahl zwischen nativen, hybriden oder webbasierten Lösungen. Die Benutzererfahrung ist entscheidend, da native Anwendungen oft eine reichhaltigere und reaktionsschnellere Benutzeroberfläche bieten als webbasierte oder hybride mobile Anwendungen[1].

Leistungsüberlegungen sind besonders wichtig bei grafischen Benutzeroberflächen oder datenintensiven Aufgaben, bei denen native Anwendungen in der Regel besser abschneiden als hybride oder webbasierte mobile

Lösungen. Schließlich stellt das Update-Management ein Problem dar, insbesondere bei nativen Anwendungen, bei denen die Benutzer neue Versionen nicht immer sofort übernehmen, was die Entwicklung und den Support erschwert[1].

Weitere wichtige Auswahlfaktoren sind noch ebenfalls zu berücksichtigen. Der folgenden Grafik fasst zusammen diese ganzen Faktoren.

Considerations	Native	Hybrid	Web
Effort of supporting platforms and versions	High	Medium	Low
Device capabilities access	Full	Full	Partial
User experience	Full	Full	Medium
Performance	Very high	Very high	High
Upgrade in the client	Needed	Needed	Not needed
Ease of publication/distribution	Medium	Medium	High
Approval cycle	Mandatory	In some cases	Not required
Monetization in app store	Available	Available	Not available

Abbildung 5.4: Kriterien zur Berücksichtigungswahl von nativen, hybriden oder Web-Anwendungsansatzes [1]

## 5.2 Cross-Platform-Entwicklung

Das weltweite Marktwachstum der mobilen Welt hat die Desktop-Entwicklung längst überholt. Die technologische Innovation im Bereich der mobilen Datenverarbeitung, die zunehmenden Möglichkeiten mobiler Geräte, die drastische Verbesserung der Benutzerfreundlichkeit und des Aussehens von Handheld-Geräten sowie ihre immer niedrigeren Preise haben zu einer enormen Anzahl von mobilen Geräten auf dem Markt geführt. Der einfache Zugang zu dieser Technologie und die rasche Zunahme der verkauften mobilen Geräte haben zu einer starken Nachfrage nach mobilen Anwendungen geführt, wie aus Abbildung 1 hervorgeht. Im Einzelnen zeigt die Abbildung, dass die Anzahl heruntergeladenen Apps zwischen 2016 und 2022 weiter zugenommen hat. Besonders im Jahr 2023 stagnierten die weltweiten App-Downloads jedoch bei 257 Milliarden, was einem Anstieg von nur 1% gegenüber dem Vorjahr entspricht 5.5.

Die Entwicklung mobiler Anwendungen hat zur Verbreitung von Plattformen und Werkzeugen beigetragen, die die nativen Eigenschaften der Geräte und die Besonderheiten ihrer Bildschirme berücksichtigen. Andererseits hat die Vielfalt der Betriebssysteme für mobile Geräte dazu zur Folge, dass aufgrund der Inkompatibilität dieser Systeme mehrere Programmiersprachen eingesetzt werden müssen und eine einzige Anwendung für verschiedene Systeme und Schnittstellen entwickelt werden muss, was mit höheren Kosten und längeren Entwicklungszeiten verbunden ist. In Anbetracht dieser Situation haben verschiedene Lösungen eine Einteilung der mobilen Anwendungen in vier verschiedene Typen vorgenommen: native Anwendungen, hybride Anwendungen, spezifische Webanwendungen für eine bestimmte Plattform und generische mobile Webanwendungen, d.h. mobile Websites, die auf jeder Plattform funktionieren. Darüber hinaus haben mobile Geräte unterschiedliche Bildschirmgrößen, Auflösungen und Seitenverhältnisse, was die Darstellung von Anwendungen komplex macht. Daher ist es für einen plattformübergreifenden Ansatz unerlässlich, die Definitionen der Benutzeroberfläche (UI) klar vom restlichen Entwicklungscode zu trennen[29].

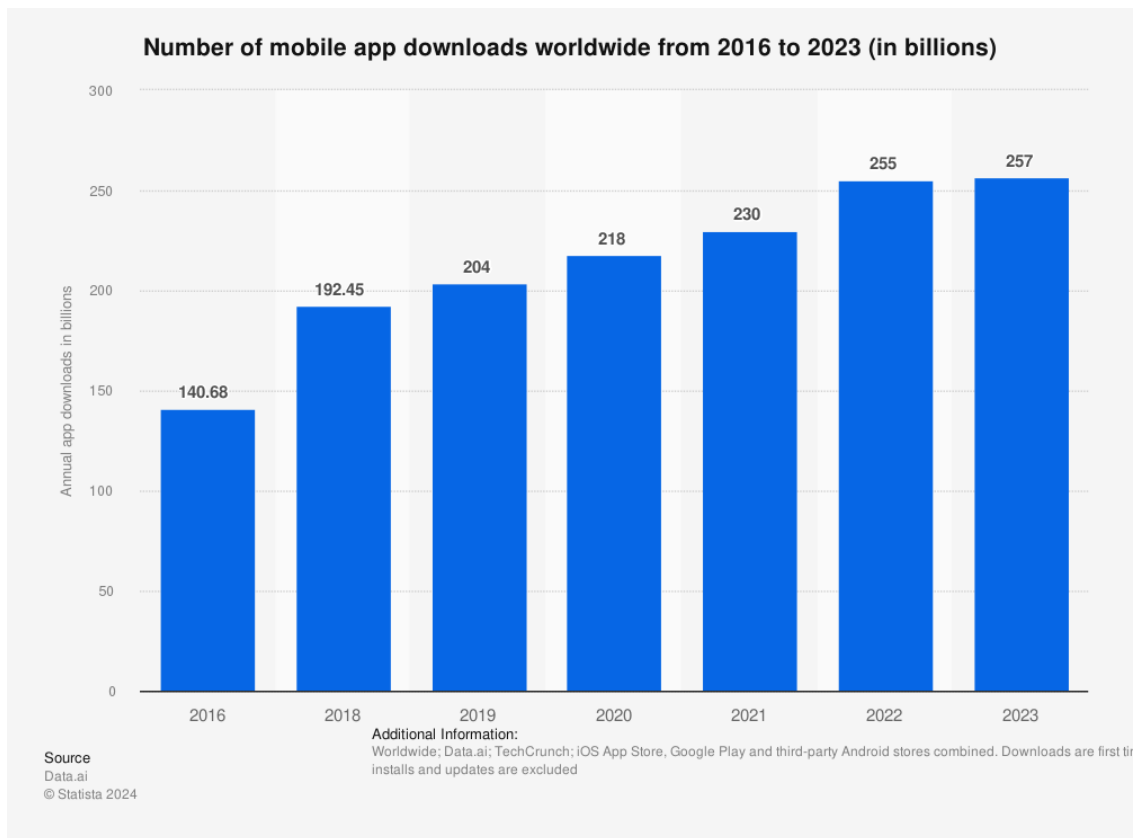


Abbildung 5.5: Geschätzte Anzahl heruntergeladener mobiler Anwendungen[30]

Die plattformübergreifende Entwicklung (Cross-Plattform Development) befasst sich mit der Entwicklung von Softwareanwendungen, die auf verschiedenen Betriebssystemen oder Plattformen ausgeführt werden können.

### 5.2.1 Aktuelle Cross-Plattform-Frameworks

Im Rahmen dieser Arbeit werden die wichtigsten und erfolgreichsten Möglichkeiten für die plattformübergreifende Entwicklung mobiler Anwendungen verglichen.

#### Apache Cordova

Apache Cordova stellt die Open-Source-Framework-Version von PhoneGap dar, einer offenen Plattform, die sich der Erstellung von nativen Anwendungen widmet, die mit verschiedenen Geräten kompatibel sind. Diese Technologie bietet Entwicklern die Möglichkeit, Anwendungen für Mobiltelefone auf der Grundlage von HTML5, CSS und JavaScript zu entwerfen. Die Benutzeroberfläche von Cordova-Anwendungen wird in Form einer WebView dargestellt, die den gesamten Bildschirm des Geräts einnimmt und in einem nativen Container ausgeführt wird. Die WebView bleibt auf allen Plattformen einheitlich, nur der native Container variiert je nach mobilem Betriebssystem. Cordova erleichtert den Zugriff auf die Funktionen von Mobilgeräten über verschiedene Plattformen hinweg durch JavaScript-Plugins. In Cordova gibt es keinen Kompilierungsprozess. Die Technologie beinhaltet eine Reihe von Standard-Plugins, die sogenannten Basis-Plugins, mit denen Entwickler die Fähigkeiten des Geräts wie Akku, Kamera, Kontakte und Speicher nutzen können [29].

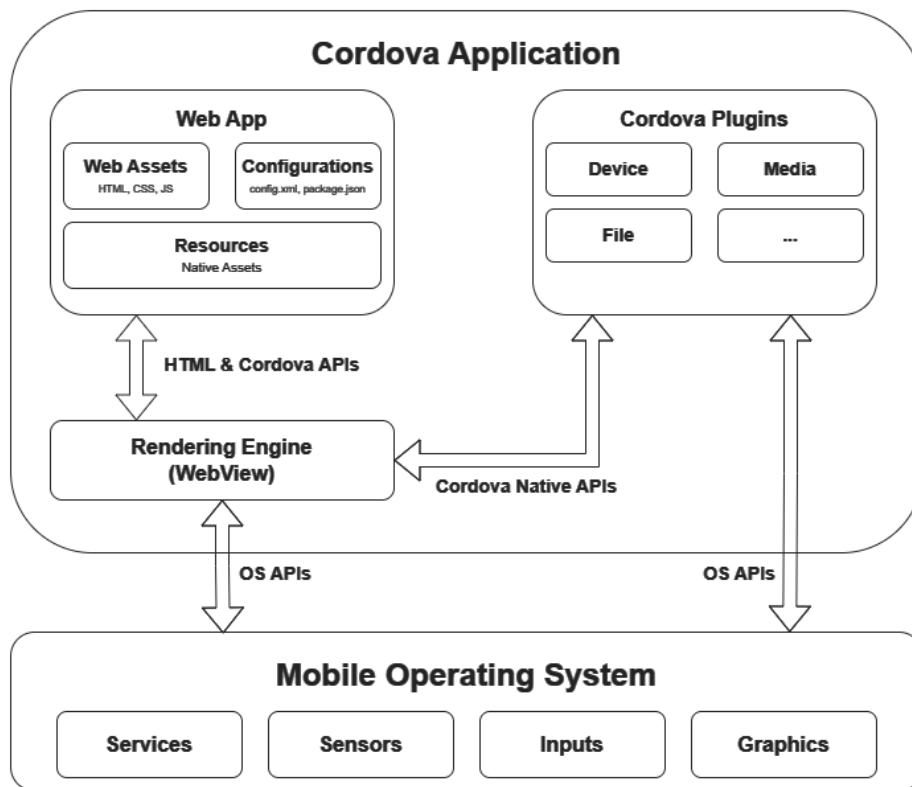


Abbildung 5.6: Cordova Architektur [31]

Die detailliertere Darstellung der Architektur von Apache Cordova wird in der obenstehenden Grafik präsentiert. Unter anderem gehören dazu die wichtigsten Komponenten[31]:

- **WebView**

Der Cordova-WebView ist in der Lage, die Anwendung mit ihrer kompletten Benutzeroberfläche auszustatten. Darüber hinaus kann er auf bestimmten Plattformen als Element innerhalb einer umfassenderen hybriden Anwendung fungieren, wobei der WebView mit nativen Anwendungskomponenten verschmolzen wird.

- **Web-App**

Hier befindet sich der Code der App. Die Anwendung selbst ist wie eine Webseite strukturiert, in der Regel eine lokale Datei namens index.html, die auf CSS, JavaScript, Bilder, Mediendateien oder andere Ressourcen verweist, die für den Betrieb wichtig sind. Die Anwendung wird in einem WebView ausgeführt, der im Wrapper der nativen Anwendung enthalten ist, die Sie über App Stores vertreiben. Innerhalb dieses Containers befindet sich eine wichtige Datei - die Datei config.xml. Diese Datei enthält Details über die Anwendung und definiert Parameter, die ihre Funktionalität beeinflussen, wie z.B. ihre Reaktionsfähigkeit auf Ausrichtungsänderungen.

- **Plugins**

Plug-ins spielen eine entscheidende Rolle im Cordova-Ökosystem, indem sie die Kommunikation zwischen Cordova und nativen Komponenten erleichtern und Verbindungen zu Standard-Geräte-APIs herstellen. Diese Interaktion ermöglicht das Auslösen von nativem Code aus JavaScript. Das Apache Cordova Projekt verwaltet eine Sammlung von Plugins, die als Core Plugins bekannt sind. Mit diesen Core-Plugins kann Ihre Anwendung auf Gerätefunktionen wie Batterie, Kamera, Kontakte usw. zugreifen.

Neben den Core-Plugins gibt es eine Vielzahl von Plugins von Drittanbietern, die zusätzliche Verbindungen zu Funktionen bieten, die möglicherweise nicht auf allen Plattformen verfügbar sind. Sie können Cordova-Plugins mithilfe von Plugin-Suchwerkzeugen oder npm finden. Sie können auch Ihre eigenen

Plugins entwickeln, wie im Plugin-Entwicklungshandbuch beschrieben. Plugins werden immer wichtiger, insbesondere um die Kommunikation zwischen Cordova und angepassten nativen Komponenten zu erleichtern.

### Xamarin

Xamarin wurde im Mai 2011 von Entwicklern gegründet, die zuvor an Mono, Mono for Android und MonoTouch gearbeitet hatten - plattformübergreifende Implementierungen der Common Language Infrastructure (CLI) und der Common Language Specifications. CLI ist eine offene Spezifikation von Microsoft, die die Ausführung von Code und eine Laufzeitumgebung beschreibt. Im Februar 2016 wurde Xamarin von Microsoft übernommen. Das Cross-To-Native Framework Xamarin wurde daraufhin über ein Plugin in Visual Studio integriert, um die plattformübergreifende Entwicklung mit C# und .NET für iOS, Android und OS X zu ermöglichen [29].

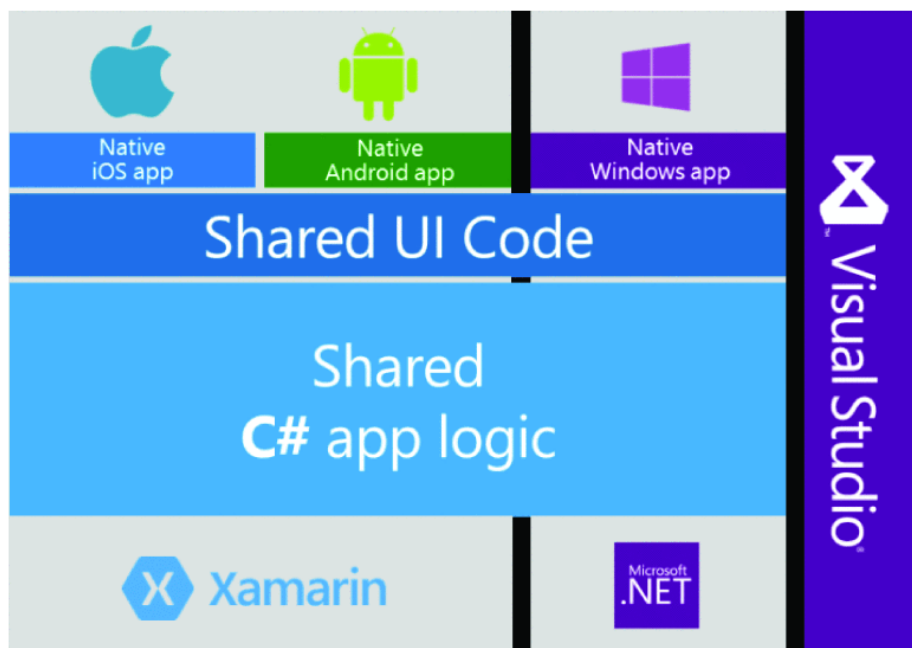


Abbildung 5.7: Cross-platform with Xamarin and .NET [29]

### Ionic

Die Ionic-Plattform wurde 2012 von Drifty Co. gegründet, um Webentwicklern die Entwicklung mobiler Anwendungen zu erleichtern. Das Framework erleichtert Entwicklern die Nutzung von Webtechnologien zur Erstellung hybrider mobiler Anwendungen. Das Ionic-Framework ist heute eine der weltweit beliebtesten Technologieplattformen für die mobile Entwicklung mit mehr als 32.990 Sternen auf GitHub, einer Plattform für das Hosting von Quellcode. Das Ionic-Framework ist ein Open-Source-Framework mit HTML, CSS und JavaScript für die Entwicklung hybrider mobiler Anwendungen. Durch die Verwendung eines hybriden Frameworks wie Ionic können Benutzer ihre Anwendungen auf verschiedenen Plattformen und Geräten mit einer einheitlichen Codebasis bereitstellen. Darüber hinaus können diese Anwendungen als Progressive Web Apps (PWAs) nahtlos in einem Standard-Webbrowser ausgeführt werden [29].

### React Native

Im Jahr 2015 wurde React Mobile von Facebook auf den Markt gebracht und ist unter der Open-Source-Lizenz des MIT verfügbar. React Native ermöglicht die Entwicklung nativ gerenderter mobiler Apps für iOS und Android mit JavaScript. Es nutzt React, eine JavaScript-Bibliothek von Facebook, die für die Gestaltung

von Benutzeroberflächen konzipiert wurde. Im Gegensatz zu herkömmlichen Webanwendungen zielt React Native nicht auf den Browser, sondern auf mobile Plattformen ab [29].

React Native weist vom Konzept her viele Gemeinsamkeiten mit Ionic auf. Die generierten Anwendungen werden in JavaScript anstatt in TypeScript geschrieben. Im Gegensatz zu Ionic verwendet React Native native Komponenten, um die Benutzerschnittstelle zu zeichnen. Diese sind wesentlich schneller als die von Ionic verwendeten Web-Elemente. Um plattformabhängige Funktionen zu integrieren, verwendet React außerdem eine "Brücke" zwischen der JavaScript-Welt und der nativen Welt. Das kostet zwar Performance, ist aber aufgrund der nativen Widgets für die Oberfläche nicht so häufig notwendig wie beispielsweise bei Ionic. Allerdings erfordert das User Interface durch die native Einbettung mehr Änderungen am jeweiligen plattformspezifischen Code [32].

## 5.3 Benutzeroberfläche und Interaktionsdesign

Die Benutzeroberfläche wurde in einem modernen und ansprechenden Design gestaltet, das nicht nur eine ästhetische, sondern vor allem eine funktional durchdachte Umgebung bietet. Ionic Angular ermöglicht eine plattformübergreifende Nutzung, klare Navigationselemente und eine einheitliche Menüstruktur machen die Bedienung intuitiv. Visuelle Elemente wie ansprechende Bilder und Grafiken tragen zur Benutzerfreundlichkeit bei. Diese Elemente ermöglichen es dem Nutzer, sich mühelos durch die verschiedenen Funktionen zu bewegen und relevante Informationen intuitiv zu finden. Der Aufbau der Benutzeroberfläche orientiert sich an bewährten Designprinzipien, um eine optimale User-Experience zu gewährleisten.

Zur klaren Orientierung wurde ein Use Case Diagramm erstellt. Dieses ermöglicht eine übersichtliche Visualisierung der verschiedenen Optionen und Interaktionsszenarien innerhalb der Anwendung und bildet die Grundlage für ein effektives User-Interface Design. Dieses Diagramm dient auch als Leitfaden für die Identifikation und Definition der einzelnen Anwendungsfälle und ermöglicht ein umfassendes Verständnis der Funktionalität und Interaktion zwischen Benutzer und Anwendung. Darüber hinaus trägt die Verwendung eines Use Case-Diagramms zu einem besseren Verständnis der Anforderungen aus Benutzersicht bei und ermöglicht so ein zielgerichtetes User-Interface-Design. Jeder identifizierte Anwendungsfall repräsentiert eine bestimmte Interaktion oder Aufgabe, die ein Benutzer innerhalb der Anwendung ausführen kann.

Dieser strukturierte Ansatz hilft nicht nur, die Benutzeranforderungen klar zu definieren, sondern fördert auch die Entwicklungskonsistenz, da jedes Element der Benutzerschnittstelle auf die unterstützten Anwendungsfälle zugeschnitten ist. Auf diese Weise wurde sichergestellt, dass das Interface nicht nur ästhetisch ansprechend ist, sondern auch funktional den Bedürfnissen der Benutzer entspricht. Das UML-Diagramm UML stellt die Akteure mit ihren verschiedenen Rollen gegenüber. Einerseits die Nutzer, die über das Register die Möglichkeit haben, ein Konto zu erstellen und sich einzuloggen, auf wichtige Dokumente zuzugreifen und diese herunterzuladen, Zugang zu den verschiedenen Terminen und Veranstaltungen sowie zu den FAQ zu erhalten. Sie können auch diversen Formulare ausfüllen, wie z.B. eine Checkliste, die Informationen für die Nutzer enthält, um sie in ihrer Stadt zu integrieren. Auf der anderen Seite sind die Administratoren verantwortlich für die Verwaltung der Inhalte, einschließlich der verschiedenen Dienste, die sie den Benutzern zur Verfügung stellen.



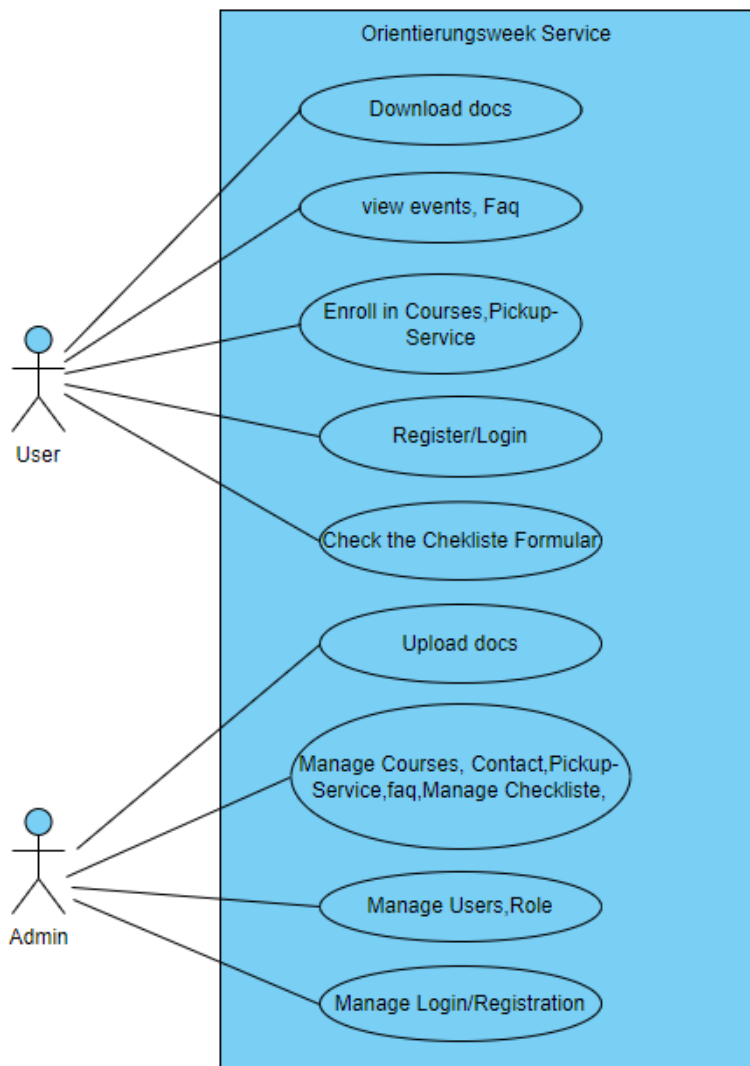


Abbildung 5.8: UML Diagramm  
Quelle: Eigene Grafik

## 5.4 Sicherheitsaspekte

Bei einem Headless-CMS wie Strapi werden API-Aufrufe an das Backend getätigt, das entweder REST oder GraphQL verwendet, um Inhalte zu übermitteln. Allerdings erhöht sich das Risiko einer Sicherheitsverletzung, wenn Anwendungen aus unterschiedlichen Domänen erlaubt werden, Anfragen an die Anwendung zu senden. Zwei Fälle listen einige Gefahren auf, die mit Ihrer Strapi-App auftreten können[33].

Zunächst ist bei der Nutzung einer REST-API:

- Während eines Distributed-Denial-of-Service-Angriffs (DoS) oder eines Distributed-Denial-of-Service-Angriffs (DDoS) sendet der Angreifer Dutzende Anfragen an den Server, um dessen Bandbreite zu überlasten. Der Host-Computer verliert dann seine Verbindung zum Internet und ist für Benutzer nicht mehr verfügbar. Die direkten Auswirkungen erstrecken sich auf die Geschäftsleistung und können zu einer Verschlechterung des Rufs des Unternehmens führen[33].
- Bei einem Cross-Site Request Forgery (CSRF)-Angriff verwendet der Angreifer die authentifizierte Anmeldung eines Benutzers, um eine böswillige Anfrage an eine Webanwendung zu senden. Das Hauptziel besteht darin, Daten zu stehlen oder Schaden anzurichten.

Und danach wenn der GraphQL API ist verwendet:

- Wenn in einer GraphQL-API der Resolver direkt für die Autorisierung verantwortlich ist, bedeutet dies, dass Autorisierungsprüfungen separat an verschiedenen Stellen durchgeführt werden müssen. Wenn diese Prüfung fehlschlägt, kann es zu einer API-Ausnutzung kommen[33].
- Da eine GraphQL-API komplex und ressourcenintensiv sein kann, könnte die effektive Implementierung einer Ratenbegrenzung übersehen werden. Dies macht sie anfällig für DoS-Angriffe. Darüber hinaus könnten komplexe Abfragen eine Schwachstelle in der Beziehung zwischen zwei Objekttypen aufdecken[33].

Einige Schlüsselbereiche von Strapi erfordern besondere Aufmerksamkeit, um bestimmte Angriffe zu verhindern.

unter anderem sind :

- Die Validierung dient als Schutzschild, um die Integrität der Benutzereingaben zu gewährleisten. Eine sorgfältige Validierung aller Eingaben ist entscheidend, um die richtigen Datentypen zu erhalten und das Eindringen von Schadcode zu verhindern. Obwohl Strapis Object-Relational Mapping (ORM) über integrierte Validierungsmechanismen verfügt, können Fehler bei der Verarbeitung von Datentypen zu Systemabstürzen führen. Außerdem ist die Fehlerberichterstattung von Strapi manchmal unzureichend, was die Diagnose von Validierungsfehlern erschwert[33].
- Sanitization bereinigt bösartigen Code, der von Benutzereingaben stammen kann. Strapi verwendet ORMs wie Bookshelf, um Daten vor dem Speichern zu entfernen. Um die Sicherheit weiter zu erhöhen, können zusätzliche Bereinigungsstufen auf den Code angewendet werden, bevor die Daten den Strapi ORM erreichen, insbesondere wenn die Benutzereingaben HTML oder Programmiersprachen enthalten.
- Rollen und Berechtigungen bestimmen, welcher Benutzer auf welche Routen in der Strapi-Anwendung zugreifen kann. Standardmäßig legt Strapi Einschränkungen für bestimmte Routen fest. Dies kann zu erheblichen Sicherheitsrisiken führen, da es den uneingeschränkten Zugriff auf bestimmte Routen ermöglicht, auch wenn dies nicht beabsichtigt ist. Es ist wichtig, bei der Vergabe von Berechtigungen vorsichtig zu sein und sicherzustellen, dass nur autorisierte Benutzer auf bestimmte Teile der Anwendung zugreifen können[33].
- Strapi Richtlinien:

Policies sind Funktionen, die für jede Anfrage eine bestimmte Logik ausführen, bevor sie den Controller erreicht. Sie werden hauptsächlich zur Absicherung der Geschäftslogik verwendet. Jeder Route in einem Strapi-Projekt kann eine Reihe von Policies zugeordnet werden. Zum Beispiel könnte eine Richtlinie mit dem Namen „is-admin“ prüfen, ob die Anfrage von einem Administrator-Benutzer gesendet wurde und den Zugriff auf kritische Routen beschränken. Es gibt sowohl globale Richtlinien, die jeder Route im Projekt zugewiesen werden können, als auch diskrete Richtlinien, die nur für eine bestimmte API oder ein bestimmtes Plugin gelten[34].

- Datenleck Unautorisierter Datenaustausch ist ein kritischer Faktor bei Sicherheitsaudits. Sensible Benutzerdaten, die in Ihrer Anwendung gespeichert sind, sollten weder absichtlich noch unabsichtlich an nicht vertrauenswürdige Quellen weitergegeben werden.

In Strapi kann ein Collection-Typ eine Eins-zu-Viele- oder Viele-zu-Viele-Beziehung zum Benutzer haben. Wenn eine GET-Anfrage an den Sammlungstyp gestellt wird, werden Benutzerdaten als Teil des [33]-Objekts zurückgegeben. Dies stellt eine Art Datenleck dar.

## 6 Implementierung

Dieser Abschnitt beleuchtet die praktischen Schritte und Entscheidungen, die zur Realisierung der Anwendung führen.

### 6.1 Umsetzung der Funktionalitäten

Der Fokus dieses Entwicklungsprozesses liegt auf der Implementierung der Funktionalitäten, die den Kern der Anwendung bilden. Wichtige Hauptfunktionalitäten stehen im Mittelpunkt der Applikationsentwicklung und sind von zentraler Bedeutung, da sie das Kernkonzept der Anwendung zusammenfassen.

#### 6.1.1 Header-Design

Zu Beginn des Projekts wurde ein Header-Design erstellt, das das Aussehen der Anwendung bestimmt. Der Header ist das erste, was die Benutzer sehen, wenn sie die Anwendung öffnen, daher wurde darauf geachtet, dass er einen guten Eindruck macht und den Benutzern hilft, sich zurechtzufinden.

Das Header-Design wurde so entwickelt, dass die Benutzer schnell auf die wichtigsten Funktionen und relevanten Informationen der App zugreifen können. Dafür wurde ein responsives Verhalten für mobile Ansichten aufweist, was bedeutet, dass es sowohl auf kleinen Smartphone-Bildschirmen als auch auf größeren Tablet-Bildschirmen gut funktioniert. Darüber hinaus bietet der Header ein konsistentes Erlebnis auf verschiedenen Bildschirmen, unabhängig davon, ob der Benutzer die iOS-App, die Android-App oder die Webversion verwendet. Diese Einheitlichkeit trägt zur Benutzerfreundlichkeit bei und vermeidet Verwirrung.

Außerdem wurde das Design des Anwendungsheaders optimiert, um sich nahtlos in das Gesamtdesign der App einzufügen. Es ergänzt die Farbpalette, die Schriftarten und die visuellen Elemente der Webanwendung, um eine kohärente und ansprechende Benutzeroberfläche zu schaffen. Dieses ganzheitliche Design trägt zu einem positiven Eindruck der Anwendung bei und fördert die Nutzerbindung.

Auf den folgenden Abbildungen sind die Header-Designs in Abhängigkeit von der Bildschirmgröße dargestellt.

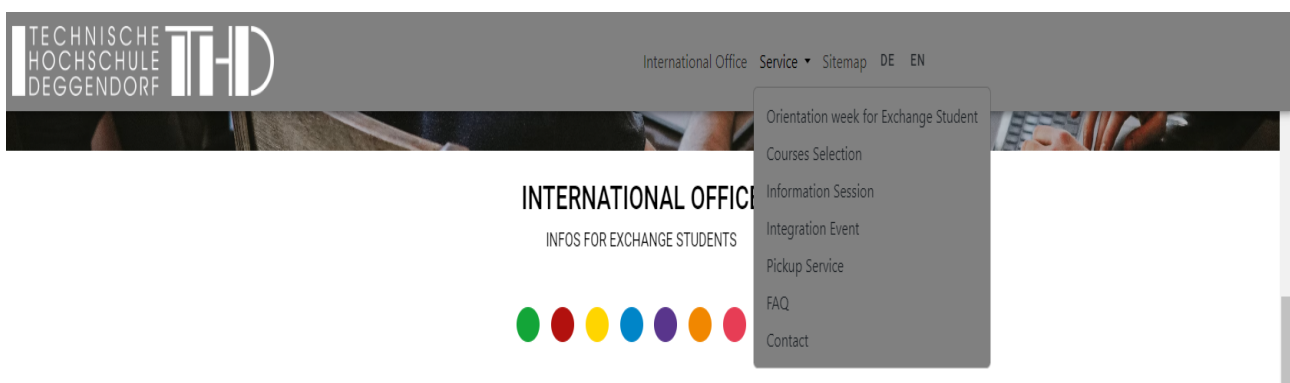


Abbildung 6.1: Header-Designs für große  
Bildschirmgröße  
Quelle: Eigene Grafik

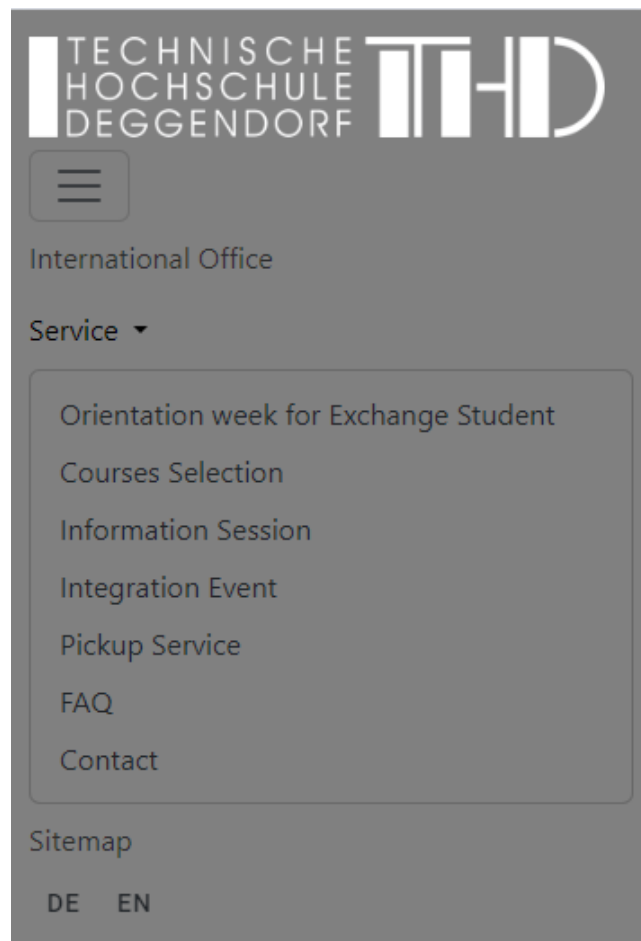


Abbildung 6.2: Header-Designs für kleine  
Bildschirmgröße  
Quelle: Eigene Grafik

Im Gegensatz zur großen Bildschirmgröße wird das Menü bei kleiner Bildschirmgröße in ein Dropdown-Menü verschoben.

### 6.1.2 Orientierungswoche Zeitplan

Die App bietet einen detaillierten Überblick über alle Veranstaltungen der Orientierungswoche. Nutzer erhalten umfassende Informationen zu verschiedenen Programmelementen, darunter Vorträge, Workshops und gesellschaftliche Veranstaltungen. Dank einer übersichtlichen Darstellung haben Nutzer die Möglichkeit, schnell und einfach auf das gesamte Veranstaltungsprogramm zuzugreifen.

Eine der Hauptfunktionen der App besteht darin, den Nutzern während der Orientierungswoche den Zeitplan jeder Veranstaltung zu präsentieren. Durch die Integration von Zeitplänen können die Teilnehmer genau bestimmen, wann und wo eine bestimmte Aktivität stattfindet. Für eine einfache Navigation und Teilnahme werden außerdem detaillierte Standortinformationen bereitgestellt. Die Integration der beiden Kernfunktionalitäten garantiert den Teilnehmern der Orientierungswoche an der THD Deggendorf ein umfassendes und benutzerfreundliches Erlebnis.

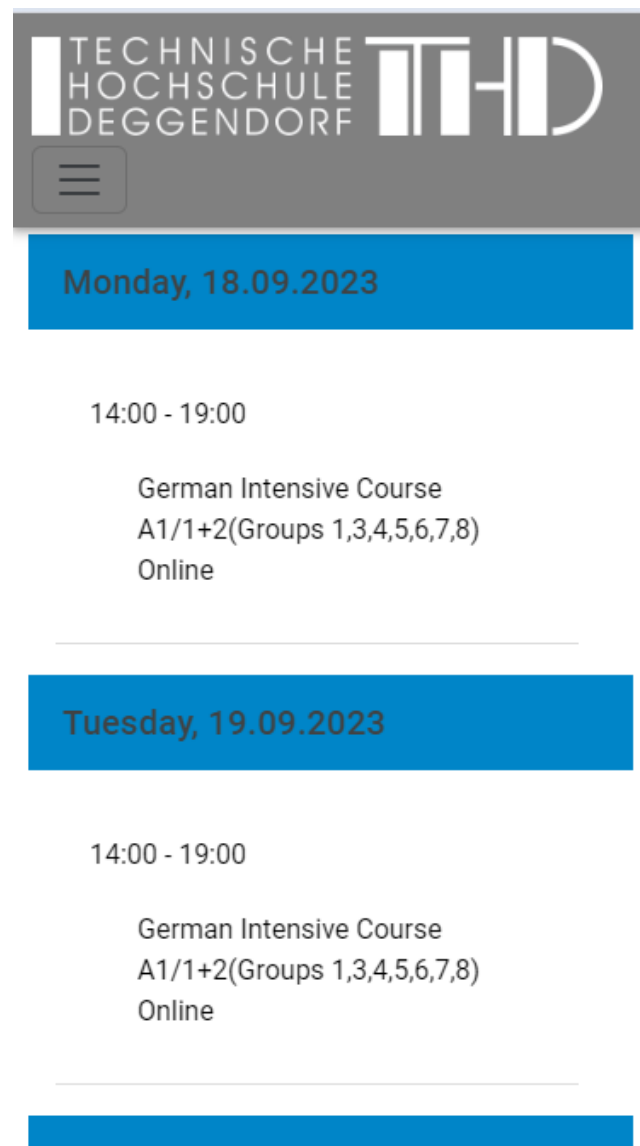


Abbildung 6.3: Orientierungswoche  
Quelle: Eigene Grafik

### 6.1.3 Dokumente Herunterladen

Ein weiteres wichtiges Merkmal der Anwendung ist der einfache Zugriff auf wichtige Informationen durch das Herunterladen von Dokumenten im PDF-Format. Dadurch können Benutzer detaillierte Dokumentationen wichtiger Ereignisse herunterladen, sei es der Kalender, informative Präsentationen oder relevante Handbücher. Die digitale Verfügbarkeit dieser Materialien erleichtert nicht nur die Vorbereitung vor der Orientierungswoche, sondern ermöglicht den Teilnehmern auch den Zugriff auf relevante Informationen während und nach den Veranstaltungen.

Die intuitive Benutzeroberfläche der App vereinfacht den Downloadvorgang. Benutzer können mit nur wenigen Klicks auf die benötigten Dokumente zugreifen, diese anzeigen und auf ihren Geräten speichern. Dies fördert nicht nur die Effizienz, sondern ermöglicht auch die papierlose und umweltfreundliche Übermittlung wichtiger Informationen.

Regelmäßige Updates stellen außerdem sicher, dass hochgeladene Dokumente immer auf dem neuesten Stand sind. Dadurch ist sichergestellt, dass die Teilnehmer jederzeit über relevante Informationen verfügen und sich auf eine reibungslose Orientierungswoche vorbereiten können. Das Herunterladen von Dokumenten ist daher eine Schlüsselfunktion, die den Benutzern angeboten werden muss und bietet ein zeitgemäßes Erlebnis.

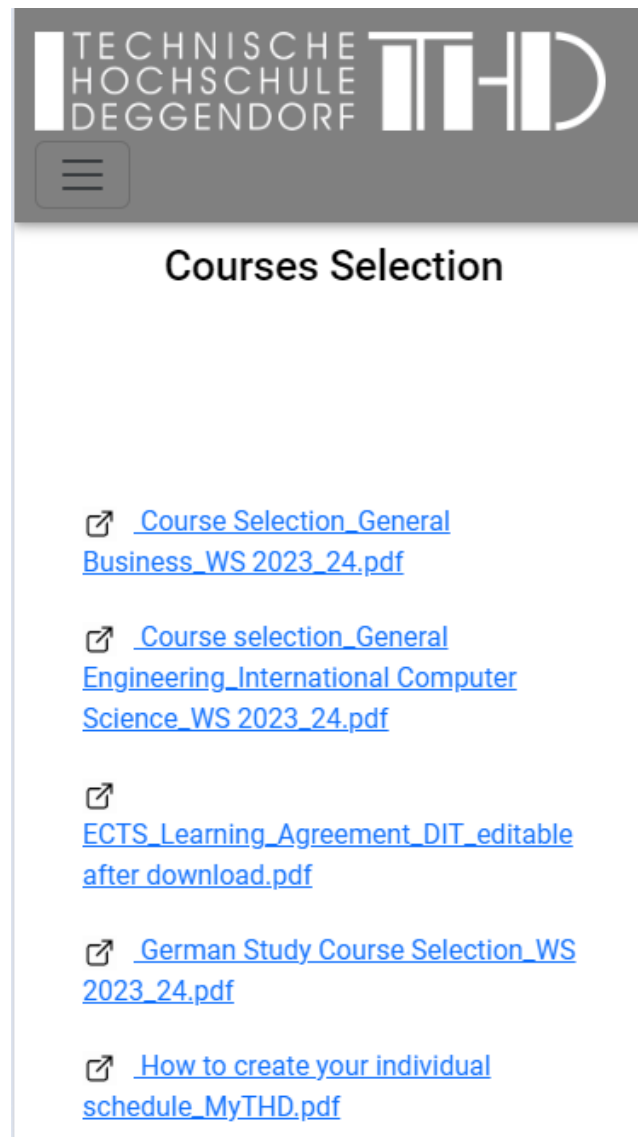


Abbildung 6.4: Dokumente Herunterladen  
Quelle: Eigene Grafik

#### 6.1.4 Pickup-Service

Eine Besonderheit der App ist der Abholservice für Austauschstudierende, die zum ersten Mal an einer Hochschule studieren. Dieser Service bietet Studierenden, die am Flughafen oder Bahnhof ankommen, einen bequemen und sicheren Transfer zu ihrer Unterkunft.

Um diesen Service zu nutzen, steht den Austauschstudierenden ein benutzerfreundliches Formular zur Verfügung. Dieses Formular enthält alle notwendigen Informationen, die das International Office benötigt, um den Abholservice effizient zu organisieren. Dazu gehören beispielsweise die Ankunftszeit, der genaue Ankunftsort und die Adresse der Unterkunft.

Ziel ist es, den Austauschstudierenden bei ihrer Ankunft die bestmögliche Unterstützung zu bieten. Sobald das Formular vollständig ausgefüllt ist, erhält das International Office die notwendigen Informationen und organisiert umgehend den Abholservice. Dies ermöglicht den Studierenden einen reibungslosen Start ins Studium und erleichtert die Integration in die neue Studienumgebung.

TECHNISCHE HOCHSCHULE DEGGENDORF THD

Full Name

Address of the Dormitory

Apartment Number  
0

Exact Date of Arrival  
tt.mm.jjjj

Estimate Time of Arrival

Phone Number  
0

Abbildung 6.5: Pickup-Service  
Quelle:Eigene Grafik

### 6.1.5 Faq

Die FAQ (Frequently Asked Questions - Häufig gestellte Fragen) sind eine zentrale Ressource für Austauschstudenten, um Antworten auf eine Vielzahl von Fragen zu erhalten, die sie während ihres Studienaufenthalts haben könnten. Dazu gehören Informationen zu Visa- und Aufenthaltsbestimmungen, zur Anmeldung zu Kursen und Prüfungen, zu Unterkunftsmöglichkeiten, zum kulturellen Leben in der Gaststadt, zu Freizeitaktivitäten, zu finanziellen Fragen, zur Gesundheitsvorsorge und vieles mehr.

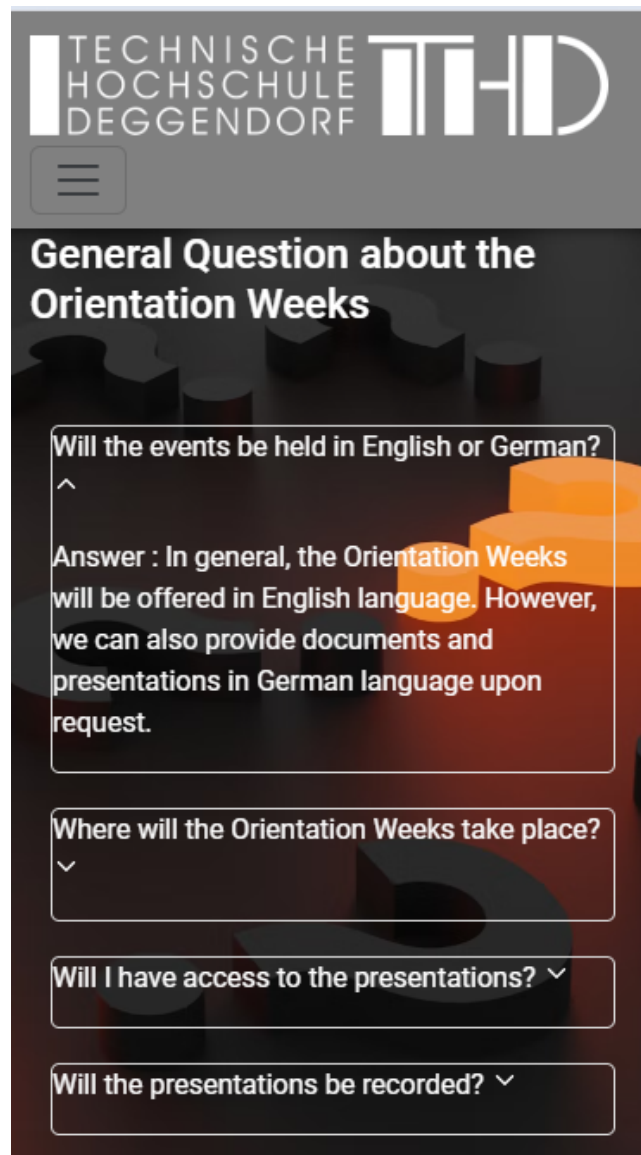


Abbildung 6.6: FAQ  
Quelle:Eigene Grafik

### 6.1.6 Checkliste

Die Checkliste ist eine Orientierungshilfe für Studierende, die in Deggendorf Behördengänge erledigen müssen. Von der Beschaffung wichtiger Dokumente wie Meldebescheinigungen und Reisepässe bis hin zur Sicherung von Bankdaten und Versicherungsnachweisen bietet die Liste eine übersichtliche Anleitung für alle notwendigen Schritte. Darüber hinaus wird auf spezielle Anforderungen bei Behördengängen, z.B. zur Ausländerbehörde oder zum Bürgeramt, sowie auf benötigte Formulare und finanzielle Mittel eingegangen. Die Checkliste bietet eine Reihe von Hilfestellungen, um sicherzustellen, dass die Studierenden über alle notwendigen Dokumente und Informationen verfügen, um die Behördengänge in Deggendorf erfolgreich zu meistern.



Checklist

DE
EN

Ausländerbehörde
☐
for the residence permit

Copy of Certificate

Ausländerbehörde
☐
of Birth (if available)

Ausländerbehörde
☐
Cash 100-110€

Copy of Registration

Bank
☐
Certificate Citizen Centre (Bürgeramt)

Bank
☐
Passport (original)

(German) Tax

Bank
☐
Identification Number

GENERATE PDF

Abbildung 6.7: Checkliste  
Quelle:Eigene Grafik

## 6.2 Backend-Implementierung mit Strapi

Bevor das Projekt startete, mussten einige Voraussetzungen erfüllt werden. Zunächst wurden der Node.js npm-Paketmanager und die Node.js-Version 18.17.1 installiert. Strapi könnte dann installiert werden. Die Installation wurde wie folgt durchgeführt:

- Ein neues Node.js-Projekt wurde initialisiert, indem der folgende Befehl ausgeführt wurde:  
`npm init -y`
- Danach erfolgt die globale Installation von Strapi mithilfe von npm:  
`npm install strapi@latest -g`
- Der nächste Schritt bestand darin, ein neues Strapi-Projekt ohne der Option „Schnellstart“ zu erstellen, um während des Einrichtungsprozesses die gewünschte Datenbank auswählen zu können:

```
strapi new internationalAppServer
```

Dadurch ist das Projekt einsatzbereit und kann durch den Befehl „npm run develop“ gestartet werden.

### 6.2.1 Collections Type

Der Prozess beginnt mit der Gestaltung von Kollektionen, die speziell auf die verschiedenen Funktionen der Anwendung zugeschnitten sind. Jede Art von Sammlung wurde mit den notwendigen Feldern ausgestattet, um die relevanten Informationen zu speichern. Gleichzeitig wurden Zugriffsrechte definiert, um sicherzustellen, dass die Daten nur von autorisierten Benutzern eingesehen oder geändert werden können. Nach der Definition der Sammlungen und Felder folgte der entscheidende Schritt der Generierung des Servercodes. Dieser Schritt automatisierte den Prozess der Datenspeicherung in der Datenbank und ermöglichte eine nahtlose Verbindung zwischen Backend und Frontend. Mit Hilfe von API-Endpunkten können nun Daten effizient zwischen den beiden Bereichen ausgetauscht werden. So kann das Frontend über definierte Endpunkte auf die im Backend gespeicherten Daten zugreifen. Die klare Trennung von Backend-Logik und Benutzeroberfläche gewährleistet eine robuste und skalierbare Architektur der Anwendung.

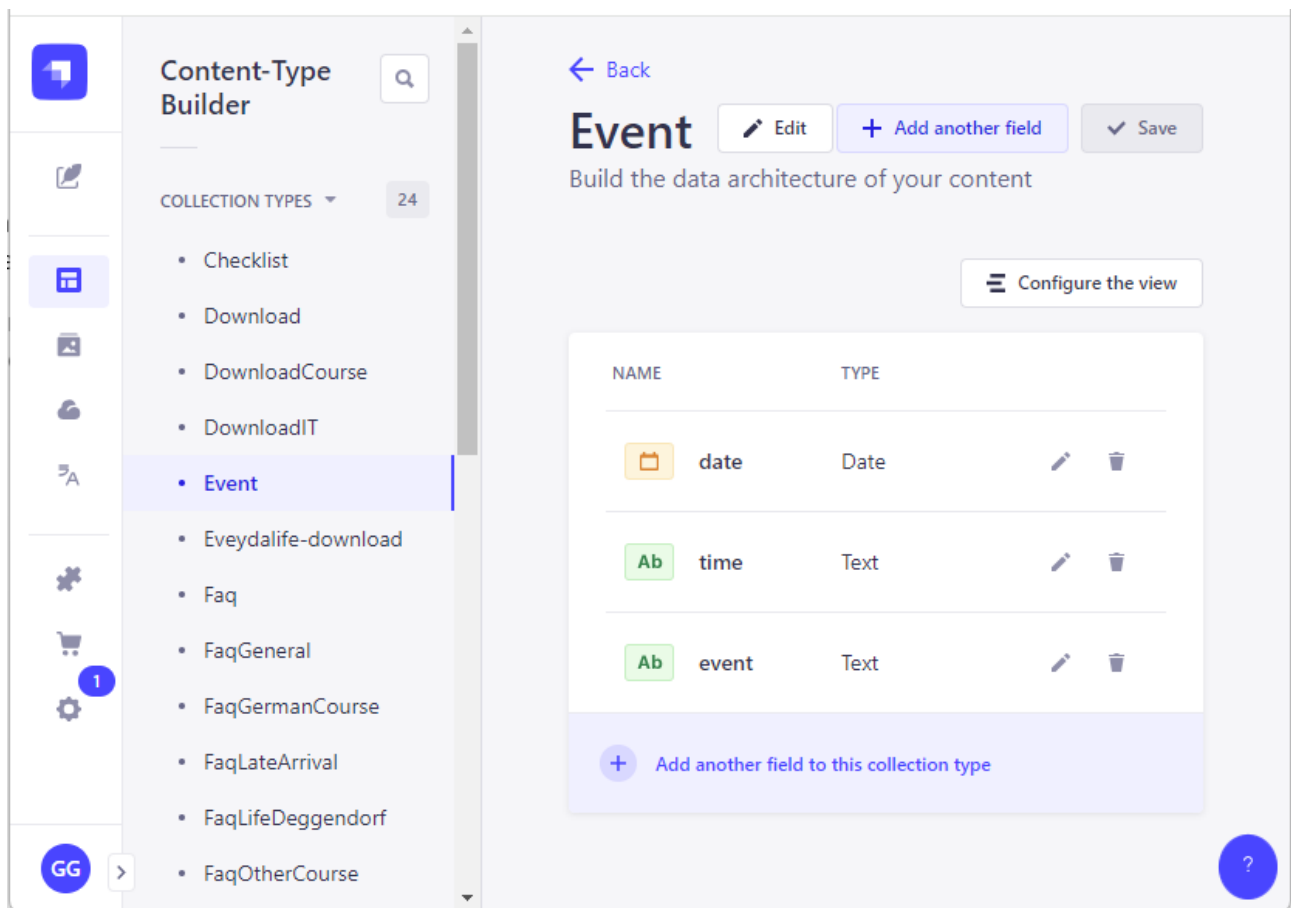


Abbildung 6.8: Collections Type Übersicht  
Quelle:Eigene Grafik

### 6.2.2 Media Library

Die Medienbibliothek der Anwendung dient als zentraler Speicherort für alle Mediendateien. Dokumente und Bilder können einfach in die Medienbibliothek hochgeladen werden, so dass wertvolle Inhalte organisiert und leicht zugänglich bleiben. Mediendateien können nicht nur hochgeladen und gespeichert, sondern auch direkt in der Medienbibliothek bearbeitet werden.

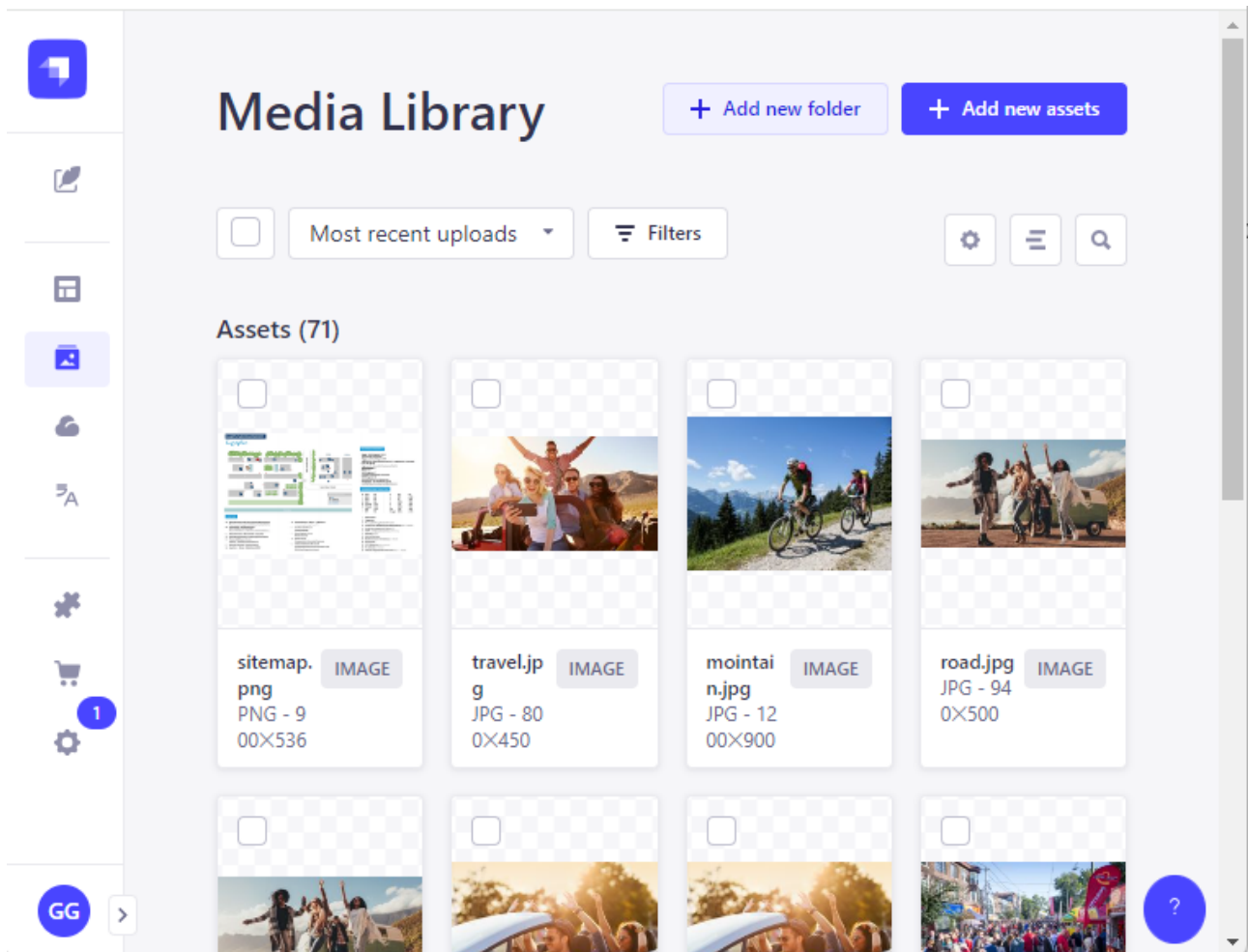


Abbildung 6.9: Übersicht der Medienbibliothek  
Quelle: Eigene Grafik

## 6.3 Frontend-Implementierung mit Ionic Angular

Um Ionic zu installieren, müssen zunächst einige Voraussetzungen erfüllt werden, unter anderem die Installation von Node.js und dessen Paketmanager. Nach dem Download des Paketmanagers erfolgt direkt die Installation von Ionic.

- Im Terminal der ausgewählten IDE, Visual Studio Code, wurde Ionic global installiert:

```
npm install -g @ionic/cli
```

- Im nächsten Schritt wird ein neues Ionic Angular-Projekt erstellt:

```
ionic start internationalAppClient blank --type=angular
```

Damit ist das Projekt betriebsbereit und kann mit dem Befehl `ionic serve` oder `ng serve` gestartet werden.

### 6.3.1 Orientierungswoche im Blick

Nachdem der Collection Type 'Event' im Strapi Content Manager erstellt wurde, um Informationen über das Orientierungswochenprogramm zu speichern, wurde in der Ionic Angular-Schnittstelle ein 'event.service.ts'-Dienst erstellt. Der EventService in Angular ruft Ereignisse von einer API ab und berücksichtigt dabei den Standort. Er verwendet HttpClient, um eine Anfrage an die API zu senden. Die Hauptmethoden sind `getAllEvents` und `getEvent`. Mit `getAllEvents` können alle Ereignisse abgerufen werden, während mit `getEvent` ein bestimmtes Ereignis anhand seiner ID abgerufen werden kann. Der Dienst unterstützt die Lokalisierung von Ereignissen, indem er die API-Antwort in die angegebene Sprache übersetzt. Dadurch können lokalisierte

Ereignisdaten nahtlos in eine Angular-Anwendung integriert werden. Im folgenden Codeabschnitt wird dies detailliert erläutert.

```

1 > import { Injectable } from '@angular/core'; ...
7
8 > @Injectable({ ...
10 })
11 export class EventService {
12     private apiUrl = `${environment.API}/api/events`;
13
14     constructor(private http: HttpClient) {}
15
16     getAllEvents(language: string): Observable<any> {
17         const url = `${this.apiUrl}?_locale=${language}&populate=*`;
18         return this.http.get<any>(url).pipe(
19             map((response) => this.transformResponse(response, language))
20         );
21     }
22
23     getEvent(eventId: number, language: string): Observable<any> {
24         const url = `${this.apiUrl}/${eventId}?_locale=${language}&populate=*`;
25         return this.http.get<any>(url).pipe(
26             map((response) => this.transformResponse(response, language))
27         );
28     }
29
30     // Method to transform the API response
31     private transformResponse(response: any, language: string): any {
32         if (response && response.data) {
33
34             response.data = response.data.map((event: any) => {
35                 if (event.attributes.localizations) {
36                     const localization = event.attributes.localizations.data.find(
37                         (loc: any) => loc.attributes.locale === language
38                     );
39                     if (localization) {
40                         event.attributes = localization.attributes;
41                     }
42                 }
43             });
44         }
45     }
46 }

```

Abbildung 6.10: Event-Service  
Quelle:Eigene Grafik

Als nächstes wird eine Komponente namens EventComponent entwickelt, die für die Anzeige der Ereignisse verantwortlich ist. Sie interagiert mit EventService, um die ortsbezogenen Ereignisdaten über TranslateService von der API abzurufen. Zu den Hauptfunktionen gehören das Laden von Ereignissen für die aktuelle Sprache, das Formatieren von Daten, das Gruppieren von Ereignissen nach Tag und Monat und das automatische Aktualisieren, wenn sich die Sprache ändert. Insgesamt bietet EventComponent eine benutzerfreundliche Lösung zur Anzeige von Ereignissen, die Sprachunterstützung und Lokalisierung ermöglicht. Der folgende Codeausschnitt gibt einen detaillierten Einblick in diese Funktionalität.

```

1 > import { Component, OnInit } from '@angular/core'; ...
10
11 > interface Event { ...
22 }
23 > interface GroupedEvents { ...
26 }
27 > interface ApiResponse { ...
37 }
38
39 > @Component({ ...
44 })
45 export class EventComponent implements OnInit {
46   events: Event[] = [];
47   groupedEvents: GroupedEvents[] = [];
48
49   constructor(
50     private eventService: EventService,
51     private datePipe: DatePipe,
52     private translate: TranslateService
53   ) {}
54
55   ngOnInit() {
56     this.loadEvents();
57     // Subscribe to language change events
58     this.translate.onLangChange.subscribe(() => {
59       this.loadEvents();
60     });
61   }
62   loadEvents() {
63     // Fetch events for the current language
64     const currentLanguage = this.translate.currentLang || 'en';
65     this.eventService.getAllEvents(currentLanguage).subscribe(
66       (response: ApiResponse) => {
67         console.log('English Events:', response.data);
68       },
69       (error: any) => {
70         console.error('Error fetching English events:', error);
71       }
72     );
73     // Fetch events for German
74     this.eventService.getAllEvents('de').subscribe(
75       (response: ApiResponse) => {
76         console.log('German Events:', response.data);
77       },
78       (error: any) => {
79         console.error('Error fetching German events:', error);

```

Abbildung 6.11: EventComponent

Quelle:Eigene Grafik

### 6.3.2 Überblick zum Herunterladen von Dokumenten

Nach der Einrichtung des Collection-Typen „downloadCourse“ im Strapi, um Dokumente zum Download zu speichern, wurde im Ionic Angular Frontend der Service „download-course.service.ts“ erstellt, der Methoden zum Abrufen aller verfügbaren Downloads oder zum Abrufen eines bestimmten Downloads anhand seiner ID bereitstellt. Der Service verwendet den HttpClient für Anfragen und konfiguriert die Basis-API-URL über Umgebungsvariablen. Auf diese Weise ermöglicht er eine effiziente Integration der Kursdownloads in die Angular-Anwendung.

```

1  import { Injectable } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3  import { Observable } from 'rxjs';
4
5  import { environment } from 'src/environments/environment';
6
7
8  @Injectable({
9    providedIn: 'root'
10 })
11 export class DownloadCourseService {
12   private apiUrl = `${environment.API}/api/download-courses?populate=*`;
13
14   constructor(private http: HttpClient) { }
15
16   getAllDownloads(): Observable<any> {
17     return this.http.get<any>(this.apiUrl);
18   }
19
20   getDownload(downloadId: number): Observable<any> {
21     return this.http.get<any>(`${this.apiUrl}/${downloadId}`);
22   }
23 }

```

Abbildung 6.12: Course Service  
Quelle:Eigene Grafik

Anschließend wird eine Komponente namens „DownloadCourseComponent“ erstellt, die für die Anzeige der herunterladbaren Kurse verantwortlich ist und mit dem DownloadCourseService interagiert, um Kursdaten von der API abzurufen. Bei der Initialisierung lädt die Komponente alle verfügbaren Kursdownloads und bietet eine Methode zum Herunterladen der Kursmaterialien. Insgesamt ermöglicht die Komponente eine benutzerfreundliche Darstellung und Interaktion mit herunterladbaren Kursen in der Anwendung. Eine detaillierte Darstellung dieser Funktionalität wird im folgenden Codeabschnitt gegeben.

```

1 > import { Component, OnInit } from '@angular/core'; ...
7
8
9 > interface Download { ...
24 }
25
26 > @Component({ ...
31 })
32 export class DownloadCourseComponent implements OnInit {
33     downloads: Download[] = [];
34     environment = environment;
35
36     constructor(private downloadCourseService: DownloadCourseService ) { }
37
38     ngOnInit() {
39         this.loadDownloads();
40     }
41
42     loadDownloads() {
43         this.downloadCourseService.getAllDownloads().subscribe(
44             (response) => {
45                 if (response && response.data) {
46                     this.downloads = response.data;
47                 } else {
48                     console.error('Invalid API response format:', response);
49                     this.downloads = [];
50                 }
51             },
52             (error) => {
53                 console.error('Error fetching downloads:', error);
54                 this.downloads = [];
55             }
56         );
57     }
58 }

```

Abbildung 6.13: Download Course  
Quelle:Eigene Grafik

### 6.3.3 Einblick in den Pickup-Service

Der Service „PickupServiceService“ von Angular ermöglicht die Kommunikation mit der API zur Erstellung von Abholservices. Für die Implementierung des Pickup-Service wurde eine Methode erstellt. Im Wesentlichen wurde die Methode createPickupService() implementiert. Dieser Dienst ermöglicht das Senden von HTTP-POST-Anfragen, um neue Abholservices mit spezifischen Details wie Name, Adresse, Ankunftsdatum, Zeit, Telefonnummer und E-Mail zu erstellen. Der Dienst verwendet den HttpClient und Umgebungsvariablen, um die API-Endpunkte zu konfigurieren. Insgesamt bietet der Dienst eine effiziente Möglichkeit, Abholservices in der Angular-Anwendung zu verwalten und zu erstellen.

```

1  import { Injectable } from '@angular/core';
2  import { HttpClient } from "@angular/common/http";
3  import { environment } from 'src/environments/environment';
4
5  > export interface PickupService { ...
16 }
17
18 @Injectable({
19   providedIn: 'root'
20 })
21 export class PickupServiceService {
22   pickupServiceCreated$: any;
23
24   constructor(private http: HttpClient) {}
25
26   API = `${environment.API}/api/pickup-services`;
27
28   createPickupService(fullName: string, dormitoryAddress: string,
29     apartmentNumber:number,arrivalDateTime:Date,time:Date,phoneNumber:number,email:string) {
30     return this.http.post<PickupService[]>(this.API, {
31       data: {
32         fullName,
33
34         dormitoryAddress,
35         apartmentNumber,
36         arrivalDateTime,
37         time,
38         phoneNumber,
39         email
40     }

```

Abbildung 6.14: Pickup-Service  
Quelle:Eigene Grafik

Daraufhin wurde die PickupServicePage erstellt, um die Abholdienste in der Anwendung zu verwalten. Die Seite kommuniziert mit dem PickupServiceService, um eine HTTP-POST-Anfrage an die API zu senden. Im Erfolgsfall zeigt sie eine Erfolgsmeldung an und setzt das Formular zurück. Im Fehlerfall wird die entsprechende Fehlermeldung ausgegeben. Insgesamt bietet die Seite eine benutzerfreundliche Schnittstelle für die Interaktion mit der API zur Erstellung von Abholdiensten.



```

1 > import { HttpResponse } from '@angular/common/http'; ...
13
14 > @Component({ ...
18 })
19 export class PickupServicePage implements OnInit {
20     successMessage: string = "";
21     pickupServices: PickupService = { ...
30     };
31
32 >     constructor( ...
37 ) {}
38     ngOnInit() { }
39
40     createPickupService() {
41         this.pickupServices.time = formatDate(new Date(), 'HH:mm:ss.SSS', 'en-US');
42
43         this.pickupServiceService
44             .createPickupService(
45                 this.pickupServices.fullName,
46                 this.pickupServices.dormitoryAddress,
47                 this.pickupServices.apartmentNumber,
48                 this.pickupServices.arrivalDateTime,
49                 this.pickupServices.time as any,
50                 this.pickupServices.phoneNumber,
51                 this.pickupServices.email
52             )
53             .subscribe(
54                 (res: any) => {
55                     console.log('Success:', res);
56                     this.successMessage = 'Pickup-Service created successfully!';
57                     this.location.go(this.location.path());
58                     this.pickupServices = {
59                         fullName: "",
60                         dormitoryAddress: "",
61                         apartmentNumber: 0,
62                         arrivalDateTime: new Date(),
63                         time: "",
64                         phoneNumber: 0,
65                         email: "",
66                     };
67                 },
68                 (error) => {
69                     console.error('Error:', error);
70                     if (error instanceof HttpResponse) {
71                         console.error('Status:', error.status);
72                         console.error('Status Text:', error.statusText);
73                         console.error('Response:', error.error);

```

Abbildung 6.15: Pickup-Service Page  
Quelle: Eigene Grafik

### 6.3.4 Darstellung der Faq

Wie in früheren Fällen wurde nach der Implementierung einer Collections-Type FAQ in Strapi, die Inhalte zu verschiedenen Fragen und Antworten enthielt, ein Dienst namens „FaqService“ erstellt, um diese Inhalte über den API-Endpunkt abzurufen. Dieser Dienst verwendet HttpClient, um Anfragen an die API zu senden und die Antworten zu verarbeiten. Die getAllFaqs-Methode wird verwendet, um alle FAQs für eine bestimmte Sprache abzurufen und die API-Antwort so umzuwandeln, dass nur die relevanten Daten extrahiert werden. Die Schnittstellen „FAQ“, „Answer“ und „Children“ definieren die Struktur der zurückgegebenen Daten.

```

15  getAllFaqs(language: string): Observable<Faq[]> {
16      const url = `${this.apiUrl}?locale=${language}&populate=*`;
17      return this.http.get<any>(url).pipe(
18          map((response: any) => {
19              return response.data.map((item: any) => {
20                  return {
21                      id: item.id,
22                      attributes: item.attributes,
23                      showAnswer: false
24                  };
25              });
26          });
27      );
28  }
29
30  // Method to transform the API response
31  private transformResponse(response: any, language: string): any {
32      if (response && response.data) {
33
34          response.data = response.data.map((item: any) => {
35              if (item.attributes.localizations) {
36                  const localization = item.attributes.localizations.data.find(
37                      (loc: any) => loc.attributes.locale === language
38                  );
39                  if (localization) {
40                      item.attributes = localization.attributes;
41                  }
42              }
43              return item;
44          });
45      }
46      return response;

```

Abbildung 6.16: Faq Service  
Quelle:Eigene Grafik

Auf der generierten FAQ-Seite, die den FaqService verwendet, um die FAQ-Daten abzurufen und anzuzeigen, wird die Methode loadFaqs verwendet, die die Methode getAllFaqs des FaqService aufruft, um die FAQs für die angegebene Sprache abzurufen. Wenn die Abfrage erfolgreich ist, werden die Daten in der Variablen this.faqs gespeichert. Im Fehlerfall wird eine Fehlermeldung an die Konsole ausgegeben und die Variable this.faqs auf ein leeres Array gesetzt, um sicherzustellen, dass keine ungültigen Daten angezeigt werden.

Zusätzlich wird die Methode toggleAnswer(faq: Faq) verwendet, um die Antwort auf eine FAQ anzuzeigen oder auszublenden. Wenn die Antwort bereits angezeigt wird, wird sie ausgeblendet und umgekehrt. Auf diese Weise können Benutzer Antworten auf Fragen nach Bedarf ein- und ausblenden. Der folgende Abschnitt gibt einen detaillierten Überblick über diese Funktionalität.

```

65   loadFaq(language: string) {
66       this.faqGeneralService.getAllFaq(language).subscribe(
67           (response: Faq[]) => {
68               this.faq = response;
69           },
70           (error: any) => {
71               console.error(`Error fetching ${language} FAQs:`, error);
72               this.faq = [];
73           }
74       );
75   }
76
77   toggleAnswer(faq: Faq) {
78       faq.showAnswer = !faq.showAnswer;
79   }
80 }

```

Abbildung 6.17: Faq-Page  
Quelle: Eigene Grafik

### 6.3.5 Implementierung der Checkliste

Der Content-Type, der für die Checkliste im Content-Manager Strapi definiert wurde, wird in Ionic verwendet, um über den API-Endpoint einen Service zu erstellen. Dieser Service ruft die Daten des Collection-Types 'Checkliste' ab. Anschließend wird eine Angular-Komponente namens 'ChecklistPage' erstellt, die eine Liste von Checkliste-Elementen lädt. So können die Austauschstudierenden die Elemente überprüfen. Die geladenen Elemente werden vom 'ChecklistService' abgerufen und angezeigt. Die Studierenden können Elemente überprüfen. Die Komponente aktualisiert daraufhin die Liste der überprüften Elemente. Außerdem bietet sie eine Funktion zur Erzeugung eines PDF-Dokuments, das die überprüften Elemente enthält. Das PDF enthält ein Logo, den Titel 'Checkliste' und die überprüften Elemente. Schließlich wird das generierte PDF-Dokument gespeichert und die Seite aktualisiert.

```

26 | loadChecklistItems() {
27 |   this.checklistService.fetchChecklistItems().subscribe((data: ChecklistItem[]) => {
28 |     this.checklistItems = data;
29 |     console.log('Checklist Items:', this.checklistItems);
30 |   });
31 | }
32 |
33 | onCheckboxChange(item: ChecklistItem) {
34 |   const index = this.checkedItems.findIndex(checkedItem => checkedItem.id === item.id);
35 |   if (index !== -1) {
36 |     this.checkedItems.splice(index, 1);
37 |   } else {
38 |     this.checkedItems.push(item);
39 |   }
40 |   console.log('Checked Items:', this.checkedItems);
41 | }
42 |
43 | generatePDF() {
44 |   const doc = new jsPDF();
45 |   let y = 10;
46 |
47 |
48 |   const logo = new Image();
49 |   logo.src = 'assets/icon/thd_logo1.jpg';
50 |   const logoWidth = 80;
51 |   const logoHeight = 40;
52 |   const pageWidth = doc.internal.pageSize.width;
53 |   const headerMargin = 10;
54 |   const headerY = headerMargin;
--

```

Abbildung 6.18: Checklist-Service  
Quelle:Eigene Grafik

## 6.4 Internationalisierung

Das Internationalisierungs-Plugin (i18n) von Strapi ist ein leistungsstarkes Tool, mit dem Benutzer auf einfache Weise lokalisierte Inhalte in verschiedenen Sprachen, sogenannte Lokalisierungen, erstellen, verwalten und verteilen können. Mit dem i18n-Plugin können Administratoren nahtlos mehrere lokalisierte Versionen erstellen und ihre Inhalte im Admin-Bereich generieren. Dies erleichtert die Anpassung von Inhalten an verschiedene Zielgruppen. Für Entwickler bietet das i18n-Plugin eine beispiellose Flexibilität und ermöglicht die Erstellung lokalisierter Projekte, die Inhalte dynamisch abrufen und präsentieren, die auf die Länder oder Sprachpräferenzen des Publikums zugeschnitten sind[35].

Obwohl das i18n-Plugin nicht automatisch Inhalte übersetzt oder die Admin-Oberfläche an bestimmte Sprachen anpasst, geht seine Funktionalität weit über die einfache Übersetzung hinaus. Es bildet die Grundlage für die Erstellung mehrsprachiger Websites oder Anwendungen und erleichtert die Übermittlung personalisierter Botschaften an jedes Publikum, unabhängig von dessen Sprache. Durch die Lokalisierung von Inhalten und die entsprechende Anpassung von Nachrichten und Struktur sorgt das i18n-Plugin für eine ansprechende und integrative Benutzererfahrung für alle Benutzer[36].

## 7 Evaluation und Verbesserung

In diesem Kapitel wird zunächst untersucht, inwieweit die in Kapitel 4 aufgestellten Anforderungen erfüllt werden. Zu diesem Zweck wurden Tests und Bewertungen durch eine kleine Gruppe von Studenten durchgeführt, um sicherzustellen, dass die Anwendung den festgelegten Anforderungen und Qualitätsstandards entspricht.

### 7.1 Usability-Tests

Der Test basiert sich auf folgende Kriterien:

#### 7.1.1 Benutzerfreundlichkeit

- Funktionsfähige Links auf allen Seiten

Auf die Frage, ob alle Linken aktiv seien, antwortete die Mehrheit der Studierenden mit „Ja“.

- Die Gewährleistung der korrekten Verlinkung aller Elemente sowie die deutliche Beschreibung des Ziels jedes Links

Die Studenten waren sich einig, dass das Ziel erreicht wurde.

- Klar strukturierte Benutzeroberfläche

Die Studierenden empfanden das Anwendungsprogramm als völlig verständlich.

#### 7.1.2 Leistung

- Optimierung der Anwendung für schnelles Laden

Es wurde bemerkt, dass die Seiten ziemlich reaktionsfähig sind.

- Gewährleistung der klaren Sichtbarkeit aller Anzeigeelemente An der Größe einiger Anzeigeelemente wurden kleine Anpassungen vorgenommen.

#### 7.1.3 Visuel Erscheinungsbild und Design

- Klare Verwendung von Farbe oder Größe zur Unterscheidung von Elementen

Einige fragten, warum das Design der Website der Hochschule ähnelte. Das Argument war das dass das Ziel darin bestand, im Auftrag des Internationalen Büros eine mobile Anwendung zu erstellen und Daher sollte sich das Design nicht vollständig von dem der Universitätswebsite unterscheiden.

#### 7.1.4 Nützlichkeit der Funktion

- Hervorhebung der wichtigsten Funktionen für eine schnelle Auffindbarkeit

Die Testnutzer fanden die Verwendung des Seitenleisten-Menüs auf der Seite zur Orientierungswoche wichtig, da es den Zugang zu den verschiedenen Teilen der Seite erleichtert.

- Korrekte Ausführung aller Funktionen

Für die Studierenden erfüllt die Funktionen ihre Ziele.

### 7.1.5 Nicht Erfüllte Anforderungen

Aufgrund der Komplexität der Integration von Benachrichtigungen in die App und spezifischer Anforderungen an die Erinnerung an Kurstermine konnte diese Funktionalität nicht realisiert werden.

## 7.2 Verbesserungsvorschläge und Anpassung

- Eine Verbesserung wird für eine Funktion der deutsche Kurswahl vorgeschlagen, die nicht erwähnt wurde, weil sie als nicht wesentlich erachtet wurde, nämlich das deutsche Kurswahlformular. Anstelle der Eingabe eines Kurscodes durch die Studierenden sollte das Formular nämlich eine Liste der Kurse und ihrer jeweiligen Codes enthalten, um so eine falsche Dateneingabe zu vermeiden.
- Es wurde auch die Frage nach der Mehrsprachigkeit gestellt. Angesichts dessen, dass die Anwendung für Austauschstudenten gedacht ist, hätte die App mehr als zwei Sprachen anbieten sollen.

# 8 Schlussfolgerungen

## 8.1 Zusammenfassung

Im Rahmen dieser Arbeit zur Konzeption und Implementierung einer App für das International Office der Technischen Hochschule Deggendorf wurden verschiedene Aspekte behandelt und umgesetzt, um eine effiziente Lösung für die Anforderungen des International Office zu schaffen. Durch eine gründliche Analyse der Bedürfnisse und Anforderungen sowie eine intensive Auseinandersetzung mit den zugrundeliegenden Prinzipien wurde eine App entwickelt, die nutzerzentriert, skalierbar, sicher und barrierefrei ist.

Die Diskussion im theoretischen Hintergrundkapitel hat gezeigt, dass die Berücksichtigung dieser Prinzipien entscheidend ist, um eine qualitativ hochwertige und bedarfsgerechte Anwendung zu entwickeln. Die Umsetzung dieser Prinzipien in der praktischen Implementierung wurde erfolgreich durchgeführt, wobei besonderes Augenmerk auf eine benutzerfreundliche Oberfläche, eine robuste Architektur und die Einhaltung von Sicherheitsstandards gelegt wurde.

Die Ergebnisse dieser Arbeit zeigen, dass die entwickelte Anwendung einen signifikanten Mehrwert für das International Office bietet. Sie erleichtert die Kommunikation und den Informationsaustausch, verbessert die Effizienz der Prozesse und trägt dazu bei, den Bedürfnissen der Studierenden und der Mitarbeiter des International Office besser gerecht zu werden. Durch die verschiedenen in der Anwendung realisierten Funktionen kann die Anwendung einen nachhaltigen Beitrag zur Erfüllung der Aufgaben des International Office der Universität leisten und damit die Effizienz und Effektivität des International Service steigern und so einen positiven Beitrag zur Internationalisierung der Universität und ihrer Wettbewerbsfähigkeit auf dem globalen Bildungsmarkt leisten.

Darüber hinaus ist die Verwendung einer KI für den wissenschaftlichen Teil dieser Arbeit anzumerken. Der Einsatz von KI zur Neuformulierung des Textes führte zu schnellen und präzisen Ergebnissen. Dank ihrer maschinellen Lernfähigkeiten kann die KI den Originaltext analysieren und ihn flüssig und natürlich umschreiben, wobei die ursprüngliche Bedeutung und Absicht erhalten bleibt. Dadurch wurde nicht nur Zeit gespart, sondern auch die Qualität der Inhalte insgesamt verbessert. Außerdem profitierte der Einsatz von KI bei dieser Arbeit von ihrer Fähigkeit, große Textmengen effizient zu verarbeiten, was insbesondere bei der Überarbeitung mehrerer Dokumente oder Artikel von entscheidender Bedeutung war. Zusammenfassend lässt sich sagen, dass sich die Integration von KI in den Textredaktionsprozess als effiziente und vorteilhafte Lösung erwiesen hat.

## 8.2 Ausblick

Im Hinblick auf zukünftige Entwicklungen und Erweiterungen der für das International Office der Technischen Hochschule Deggendorf entwickelten Anwendung gibt es verschiedene Möglichkeiten, die App weiter zu verbessern und an sich ändernde Anforderungen anzupassen. Eine mögliche Richtung für zukünftige Entwicklungen ist die Integration zusätzlicher Funktionen und Module, um den Funktionsumfang der Anwendung zu erweitern.

Eine Möglichkeit besteht darin, das Applikationssystem durch die Implementierung von Feedbackmechanismen und die Analyse von Nutzungsdaten kontinuierlich zu verbessern. Dies würde es ermöglichen, das Anwendungssystem kontinuierlich an die Bedürfnisse der Nutzer anzupassen und das Nutzungserlebnis weiter zu optimieren.

Darüber hinaus könnten technologische Weiterentwicklungen wie die Einführung von KI-basierten Funktionen oder die Integration von Augmented Reality neue Möglichkeiten eröffnen, die Plattform noch interaktiver

und innovativer zu gestalten.

Insgesamt bietet der Ausblick auf zukünftige Entwicklungen eine Vielzahl von Möglichkeiten, die für das International Office der Technischen Hochschule Deggendorf entwickelte App kontinuierlich zu verbessern und an sich verändernde Anforderungen und technologische Trends anzupassen. Durch eine strategische und iterative Weiterentwicklung kann die Anwendung langfristig einen wichtigen Beitrag zur Internationalisierung der Hochschule und zur Förderung des interkulturellen Austauschs leisten.



# Abbildungsverzeichnis

2.1	Node.js Architektur[4]	4
2.2	Angular-Modul Struktur[7]	6
2.3	Ionic-Nutzungsstatistiken[11]	8
2.4	Ionic Architektur[12]	9
2.5	Traditional CMS versus headless CMS[14]	10
2.6	Strapi Content Management System[15]	11
2.7	HTML Architektur [19]	12
2.8	Visual Studio Code Quelle:Eigene Grafik	14
5.1	Mobile Anwendungsentwicklung[27]	20
5.2	Hybride App[29]	21
5.3	Native App[29]	22
5.4	Kriterien zur Berücksichtigungswahl von nativen, hybriden oder Web-Anwendungsansatzes [1]	23
5.5	Geschätzte Anzahl heruntergeladener mobiler Anwendungen[30]	24
5.6	Cordova Architektur [31]	25
5.7	Cross-platform with Xamarin and .NET [29]	26
5.8	UML Diagramm Quelle:Eigene Grafik	28
6.1	Header-Designs für große Bildschirmgröße Quelle:Eigene Grafik	30
6.2	Header-Designs für kleine Bildschirmgröße Quelle:Eigene Grafik	31
6.3	Orientierungswoche Quelle:Eigene Grafik	32
6.4	Dokumente Herunterladen Quelle:Eigene Grafik	33
6.5	Pickup-Service Quelle:Eigene Grafik	34
6.6	FAQ Quelle:Eigene Grafik	35
6.7	Checkliste Quelle:Eigene Grafik	36
6.8	Collections Type Übersicht Quelle:Eigene Grafik	37
6.9	Übersicht der Medienbibliothek Quelle:Eigene Grafik	38
6.10	Event-Service Quelle:Eigene Grafik	39
6.11	EventComponent Quelle:Eigene Grafik	40
6.12	Course Service Quelle:Eigene Grafik	41
6.13	Download Course Quelle:Eigene Grafik	42
6.14	Pickup-Service Quelle:Eigene Grafik	43
6.15	Pickup-Service Page Quelle:Eigene Grafik	44
6.16	Faq Service Quelle:Eigene Grafik	45
6.17	Faq-Page Quelle:Eigene Grafik	46
6.18	Checklist-Service Quelle:Eigene Grafik	47

# Literatur

- [1] Nicolás Serrano, Josune Hernantes und Gorka Gallardo. “Mobile Web Apps”. In: *IEEE Software* 30.5 (2013), S. 22–27. DOI: 10.1109/MS.2013.111.
- [2] Venkata Keerti Kotaru. “Angular for Material Design”. In: *Leverage Angular Material and TypeScript to Build a Rich User Interface for Web Apps*. <https://doi.org/10.1007/978-1-4842-5434-9>. Berkeley, CA: Apress, 2020, S. XIX, 364.
- [3] Node.js.org. “Node.js — Introduction to Node.js”. [Online]. 2024, [Zuletzt aufgerufen: 19.01.2024], Verfügbar unter <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>.
- [4] Biztech.com. “Top Node.js Development Trends 2023 You Cannot Miss”. [Online]. 2024, [Zuletzt aufgerufen: 05.03.2024], Verfügbar unter <https://www.biztechcs.com/blog/nodejs-development-trends/>.
- [5] David Herron. “Node.js web development”. In: *Server-side web development made easy with Node 14 using practical examples, fifth edition*. Packt Publishing, 2020, S. 10–11. URL: <https://books.google.de/books?id=430DwAAQBAJ>.
- [6] G. Geetha u. a. “Interpretation and Analysis of Angular Framework”. In: *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*. 2022, S. 1–6. DOI: 10.1109/ICPECTS56089.2022.10047474.
- [7] Mark Root. “Angular Application Structure”. [Online]. 2018, [Zuletzt aufgerufen: 15.02.2024], Verfügbar unter <https://www.intertech.com/angular-module-tutorial-application-structure-using-modules/>.
- [8] Pranay Rana. “Angular Application Structure”. [Online]. 2024, [Zuletzt aufgerufen: 15.02.2024], Verfügbar unter <https://pranayamr.blogspot.com/2018/05/angular-application-structure.html>.
- [9] Steffi. “Was ist Angular? Informationen, Vor- und Nachteile.”. [Online]. 2024, [Zuletzt aktualisiert : 02.02.2024, Zuletzt aufgerufen: 06.03.2024]. Verfügbar unter <https://platri.de/en/tech-wiki-en/what-is-angular-definition-advantages-disadvantages/>.
- [10] Chris Griffith. “What is Apache Cordova Framework and What is the Difference from PhoneGap?”. [Online]. 2019, [Zuletzt aufgerufen: 07.03.2024], Verfügbar unter <https://ionic.io/resources/articles/what-is-apache-cordova>.

- [11] trends.builtwith.com. “*Ionic Usage Statistics*”. [Online]. 2024, [Zuletzt aufgerufen: 14.02.2024], Verfügbar unter <https://trends.builtwith.com/framework/Ionic>.
- [12] Juilee Waranashiwar und Manda Ukey. “Ionic Framework with Angular for Hybrid App Development”. In: *International Journal of New Technology and Research* (2018).
- [13] N. A. Alhazmy u. a. “Building a Comprehensive Content Management System with NPM, Vue.js, Node.js, Postgresql, and Strapi”. In: *eng* (2023). <https://ojs.iuli.ac.id/index.php/eng/article/view/78>, S. 1–5.
- [14] K. Elshafie und M. Haider. *Designing Web APIs with Strapi: Get started with the Strapi headless CMS by building a complete learning management system API*. Packt Publishing, 2022, S. 4. URL: <https://books.google.de/books?id=5NpaEAAAQBAJ>.
- [15] GetApp.de. “*Strapi*”. [Online]. 2024, [Zuletzt aufgerufen: 14.02.2024], Verfügbar unter <https://www.getapp.de/software/125811/strapi>.
- [16] docs.strapi.io. “*REST API — Strapi Documentation*”. [Online]. 2024, [Zuletzt aufgerufen: 27.02.2024], Verfügbar unter <https://docs.strapi.io/dev-docs/api/rest>.
- [17] docs.strapi.io. “*GraphQL API — Strapi Documentation*”. [Online]. 2024, [Zuletzt aufgerufen: 27.02.2024], Verfügbar unter <https://docs.strapi.io/dev-docs/api/graphql>.
- [18] Felix Schürmeyer. “*HTML Basis: Erklärung der HTML5 Grundstruktur*”. [Online]. 2022, [Zuletzt aufgerufen: 19.02.2024], Verfügbar unter <https://hellocoding.de/blog/coding-language/html/grundstruktur-basis-html>.
- [19] Harry. “*Tutorialgarage.com – HTML5 und CSS3: Aufbau und Strukturierung von HTML-Dokumenten*”. [Online]. 2024, [Zuletzt aufgerufen: 20.01.2024], Verfügbar unter <https://www.tutorialgarage.com/html5-css3/aufbau-von-html-dokumenten>.
- [20] Gavin Bierman, Martin Abadi und Mads Torgersen. “Understanding TypeScript”. In: *ECOOP 2014 – Object-Oriented Programming*. Hrsg. von Richard Jones. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, S. 257–281.
- [21] GeeksforGeeks.org. “*CSS Preprocessor SASS*”. [Online]. 2019, [Zuletzt aufgerufen: 19.01.2024], Verfügbar unter <https://www.geeksforgeeks.org/css-preprocessor-sass/>.
- [22] Binildas Christudas. “MySQL”. In: *Practical Microservices Architectural Patterns: Event-Based Java Microservices with Spring Boot and Spring Cloud*. Berkeley, CA: Apress, 2019, S. 877–884. URL: <https://doi.org/10.1007/978-1-4842-4501-927>.
- [23] Alessandro Del Sole. “Introducing Visual Studio Code”. In: *Visual Studio Code Distilled: Evolved Code Editing for Windows, macOS, and Linux*. Berkeley, CA: Apress, 2021, S. 1–15. URL: <https://doi.org/10.1007/978-1-4842-6901-51>.
- [24] v7.material.angular.io. “*Angular Material*”. [Online]. 2020, [Zuletzt aufgerufen: 11.02.2024], Verfügbar unter <https://v7.material.angular.io/>.

- [25] Hochmayr Manuel. “*Das Ionic Framework in der Praxis — Objectbay*”. [Online]. 2024 , [Zuletzt aufgerufen: 29.01.2024], Verfügbar unter <https://www.objectbay.com/blog/ionic-hybride-app-entwicklung>.
- [26] Andrea Sánchez Blanco. “*Development of hybrid mobile apps : Using Ionic Framework*”. [Online]. 2016 , Verfügbar unter <https://www.theseus.fi/handle/10024/114145?show=full>.
- [27] Neha Koram und Rakhi Garg. “Review on Mobile App Development: Tools and Techniques”. In: *2023 IEEE World Conference on Applied Intelligence and Computing (AIC)*. 2023, S. 260–266. DOI: 10.1109/AIC57670.2023.10263908.
- [28] Chris W. Griffith. *Mobile app development with Ionic*. Beijing [China]; Sebastopol, CA: O’Reilly Media, Inc, 2017, S. 3–5.
- [29] Carlos Manso Pinto und Carlos Coutinho. “From Native to Cross-platform Hybrid Development”. In: *2018 International Conference on Intelligent Systems (IS)*, S. 669–676. DOI: 10.1109/IS.2018.8710545.
- [30] Statista.com. “*Annual number of mobile app downloads worldwide 2023 — Statista*”. [Online]. 2024 , [Zuletzt aufgerufen: 07.03.2024], Verfügbar unter <https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/>.
- [31] Cordova.apache.org. “*Architectural overview of Cordova platform - Apache Cordova*”. [Online]. 2024, [Zuletzt aktualisiert : 26.01.2024, Zuletzt aufgerufen: 20.03.2024] , Verfügbar unter <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>.
- [32] Leonard Thiele. “*Cross-Plattform-Entwicklung einer mobilen Anwendung mit Flutter – Konzeption und-prototypische Umsetzung*”. bachelorThesis. 2019. URL: <http://hdl.handle.net/20.500.12738/8585>.
- [33] Anumadu Moses. “*Strapi Security Checklist*”. [Online]. 2021 , [Zuletzt aufgerufen: 18.02.2024] , Verfügbar unter <https://strapi.io/blog/strapi-security-checklist>.
- [34] docs.strapi.io. “*Policies — Strapi Documentation*”. [Online]. 2024 , [Zuletzt aufgerufen: 18.02.2024], Verfügbar unter <https://docs.strapi.io/dev-docs/backend-customization/policies>.
- [35] docs.strapi.io. “*Internationalization (i18n) — Strapi Documentation*”. [Online]. 2024 , [Zuletzt aufgerufen: 11.02.2024], Verfügbar unter <https://docs.strapi.io/dev-docs/plugins/i18n>.
- [36] Strapi.io. “*Internationalization - Strapi Features*”. [Online]. 2024 , [Zuletzt aufgerufen: 11.02.2024], Verfügbar unter <https://strapi.io/features/internationalization>.