

Ostbayerische Technische Hochschule Regensburg
Fakultät für Informatik und Mathematik

Masterarbeit

Zur Erlangung des akademischen Grades
Master of Science (M. Sc.)

StudyMap: Konzeption und Evaluierung eines innovativen Studiengangfinders am Beispiel der OTH-Regensburg

Vorgelegt von: Andreas Huber <andreas.huber@st.oth-regensburg.de>

Matrikelnummer: 3370380

Studiengang: Master Informatik (Schwerp. Software Engineering)

Erstgutachter: Prof. Dr. Markus Heckner

Zweitgutachter: Prof. Dr. Daniel Jobst

Abgabefrist: 20. März 2024

Erklärung zur Masterarbeit

1. Mir ist bekannt, dass dieses Exemplar der Abschlussarbeit als Prüfungsleistung in das Eigentum der Ostbayerischen Technischen Hochschule Regensburg übergeht.
2. Ich erkläre hiermit, dass ich diese Abschlussarbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Regensburg, den 23. Februar 2024

Andreas Huber

Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	VII
Abkürzungsverzeichnis	VIII
1 Einleitung	1
1.1 Einleitung und Motivation	1
1.2 Problemstellung/Zielsetzung	1
2 Theoretischer Hintergrund und vorhandene Studiengangsfinder-Konzepte	3
2.1 Bisheriges Vorgehen bei der Studienorientierung	3
2.2 Konzepte von Studiengangsfidern	4
2.3 Theorie und Anwendung von interaktiven ähnlichen Systemen	5
3 Methodik	6
3.1 Beschreibung der Algorithmen	6
3.1.1 Force-Directed-Graphs	6
3.1.2 K-Means Clustering Algorithmus	8
3.1.3 Multidimensionale Skalierung (MDS)	9
3.2 Beschreibung der Datenquelle und -beschaffung	12
3.2.1 Herkunft der Daten	12
3.2.2 Festlegung der Inhaltskategorien	13

4 Konzept eines innovativen Studiengangsfinders	16
4.1 Vorstellung der OTH-Regensburg als Fallbeispiel	16
4.1.1 Über die OTH-Regensburg	16
4.1.2 Herausforderung und Zielsetzung	16
4.2 Konzept für eine automatisch generierte Infografik	17
4.2.1 Visualisierung der Studiengänge	17
4.2.2 Berücksichtigung von Schwerpunkten	18
4.2.3 Interaktivität	18
4.3 Datenpflege und -sicherung	20
4.3.1 Verwaltungssoftware vs. Anleitung	20
4.3.2 Staging- und Produktivumgebung	20
4.3.3 Datensicherung	21
5 User-centered Design Research	22
5.1 Mockup-Studie: Bewertung der Benutzerfreundlichkeit und Funktionalität	22
5.1.1 Methodik und Teilnehmende	22
5.1.2 Ablauf des Tests	22
5.1.3 Ergebnisse der Mockup-Studie	23
5.2 Prototypen-Studie: Evaluation durch Studieninteressierte	24
5.2.1 Methodik und Teilnehmende	24
5.2.2 Ablauf des Tests	26
5.2.3 Ergebnisse der Prototypen-Studie	28
5.3 Zusammenfassung der Studien	39

6 Implementierung und Deployment	40
6.1 Technologieauswahl und Implementierungsdetails	40
6.1.1 PixiJS - Interaktive Grafik	40
6.1.2 Python - Berechnung der Positionen	47
6.1.3 Node.js - REST-API	51
6.1.4 Angular - Administrationsoberfläche	57
6.2 Softwarearchitektur	62
6.2.1 Frontend-Komponenten	64
6.2.2 Backend-Komponenten	66
6.3 Software Deployment	68
6.3.1 Planung	68
6.3.2 Testen	71
6.3.3 Bereitstellung	77
7 Diskussion und Ausblick	79
7.1 Zusammenfassung der Ergebnisse und wichtigsten Erkenntnisse	79
7.2 Vergleich mit anderen Studienorientierungs-Tools	79
7.3 Potenzielle Erweiterungen und zukünftige Anwendungen	79
Quellenverzeichnis	80
Anhang	84
A Google Search Trends 2021: „OTH Regensburg Studiengänge“	85
B Studienverlaufsplan: Informatik	86

C	Studienverlaufsplan: International Computer Science	87
---	---	----

Abbildungsverzeichnis

1	Schrittweises Durchführen des Layout-Algorithmus von Fruchterman/Reingold	7
2	K-Means Clustering Algorithmus	8
3	Erstes Mockup einer möglichen Implementierung	17
4	Mockup: Tooltip über Bubble	18
5	Mockup: Popup mit Details eines Studiengangs	19
6	StudyMap Prototyp	25
7	Prototypen-Studie: Aufgaben (Umfrage Teil 2)	26
8	Ergebnisse: Teil 1	28
9	Welcher Studiengang ist ähnlich zu Architektur? (Aufgabe 1)	29
10	Prototyp: Architektur -> Bauingenieurwesen (Aufgabe 1)	30
11	Prototyp: Vergleichsfeature	30
12	Prototyp: Vergleichsfeature öffnen	31
13	Wordcloud der Antworten (Aufgabe 2)	32
14	Wie schwer fandest du Aufgabe 1? (Auswertung Aufgabe 1)	32
15	Wie schwer fandest du Aufgabe 2? (Auswertung Aufgabe 2)	33
16	Wordcloud zu „Was findest du an dem Konzept gut?“ (Umfrage Teil 3)	34
17	Wordcloud zu „Was findest du an dem Konzept schlecht?“ (Umfrage Teil 3)	35
18	Wordcloud zu „Hast du neue Ideen für den Prototypen?“ (Umfrage Teil 3)	37
19	StudyMap: Tooltip über Bubble	42
20	StudyMap: Supergruppe mit Label	43
21	StudyMap: Supergruppe ohne Label - Hitarea	43

22	StudyMap: Tooltip links neben Bubble	46
23	Admin UI: Überblick	59
24	Admin UI: Neue Daten einpflegen	60
25	Admin UI: Neue Daten einpflegen - Ergebnisdialog	60
26	Admin UI: Studiengangsdetails verwalten (Ausschnitt)	61
27	Softwarearchitektur von StudyMap	63
28	Basic-Auth-Middleware von StudyMap	65

Tabellenverzeichnis

1	Exemplarische Eingabetabelle für den MDS-Algorithmus	9
2	Aufteilung der Werte bei auf Studiengang genormte Werte	13
3	Probanden der Mockup-Studie	22
4	Auswertung der Fragen zu den Aufgaben	33
5	StudyMap-API: REST-Endpunkte	51

Abkürzungsverzeichnis

1 Einleitung

1.1 Einleitung und Motivation

Die Wahl des richtigen Studiengangs ist ein wichtiger Schritt im Leben eines jeden angehenden Studierenden. Sie legt den Grundstein für die akademische und berufliche Entwicklung und beeinflusst den individuellen Bildungsweg maßgeblich. Angesichts der Vielfalt an verfügbaren Studiengängen stehen Studieninteressierte vor einer komplexen Entscheidung, die durch eine breite Palette von Inhalten, Schwerpunkten und Karrierewegen geprägt ist. Diese Fülle kann zu Unsicherheit und Verwirrung führen, wodurch nicht selten falsche Studienentscheidungen getroffen werden. (Beckmann et al., 2021)

Die vorliegende Masterarbeit greift dieses weit verbreitete Problem auf und stellt einen innovativen Ansatz zur Studienorientierung vor. Das Ziel ist es, eine benutzerfreundliche, automatisch generierte Infografik-basierte Plattform zu entwickeln, die Studieninteressierte dabei unterstützt, fundierte Entscheidungen über ihren zukünftigen Bildungsweg zu treffen. Das Tool kombiniert moderne Datenanalyse, interaktive Benutzeroberflächen und nutzerzentriertes Design, um den Orientierungsprozess zu optimieren.

1.2 Problemstellung/Zielsetzung

Die derzeitige Studienorientierung wird häufig durch eine Informationsflut und eine begrenzte Transparenz der verfügbaren Studiengänge behindert. Häufig kennen Studieninteressierte einen Studiengang oder eine ungefähre Richtung, in die sie gehen möchten. Um ähnliche Studiengänge oder alle Studiengänge in der gewünschten Richtung zu finden, fehlt oft der Überblick. (Beckmann et al., 2021) Diese Unsicherheit kann zu falschen Studienentscheidungen führen, die wiederum Studienabbrüche durch Motivationsmangel zur Folge haben (Heublein et al., 2010). Das Hauptziel dieser Masterarbeit ist es, ein innovatives Tool zu entwickeln, das diese Herausforderungen adressiert und die Studienorientierung für Studieninteressierte optimiert.

Die spezifischen Ziele dieser Arbeit sind:

1. Entwicklung eines Konzepts für einen Studiengangsfinder auf Basis einer interaktiven Infografik-basierten Benutzeroberfläche.
2. Implementierung und technische Umsetzung des entwickelten Systems am Beispiel der Ostbayerischen Technischen Hochschule Regensburg (OTH-Regensburg).

3. Evaluation und Validierung deines Prototypen durch Tests mit potenziellen Nutzern und Analyse der Ergebnisse.
4. Diskussion der Stärken und Schwächen des entwickelten Systems sowie der möglichen Auswirkungen auf die Studienberatung. 
5. Die Ergebnisse dieser Arbeit sollen dazu beitragen, die Studienorientierung für Studieninteressierte zu erleichtern und ihnen bessere Informationen und Orientierungshilfen zur Verfügung zu stellen.

Die folgenden Abschnitte dieser Arbeit beschäftigen sich mit dem theoretischen Hintergrund, der angewandten Methodik, der Konzeption und Umsetzung des Studiengangsfinders an der OTH-Regensburg, der Evaluierung der Plattform sowie einer Diskussion und einem Ausblick auf zukünftige Entwicklungen und Anwendungen. 

2 Theoretischer Hintergrund und vorhandene Studiengangsfinder-Konzepte

2.1 Bisheriges Vorgehen bei der Studienorientierung

Die Studienorientierung ist ein entscheidender Schritt im Bildungsweg eines jeden Studieninteressierten und die Entscheidung für ein zukünftiges Studienfach und eine Hochschule wird traditionell von verschiedenen Faktoren beeinflusst. Eine umfassende Studie von CHE und EINSTIEG aus dem Jahr 2007 gibt Einblicke in die Präferenzen und Entscheidungsmuster von Studieninteressierten.

Die Studie zeigt, dass für einen Großteil der Befragten (87,2 %) die Wahl der Hochschule und des Hochschulortes weniger relevant ist als das gewählte Studienfach (Hachmeister et al., 2023). Dies deutet darauf hin, dass das persönliche Interesse an einem bestimmten Studienfach die Entscheidung stärker beeinflusst als die geografische Lage oder der Ruf der Hochschule.

Ein weiteres wichtiges Ergebnis der Studie ist, dass 64,6 % der Befragten ihr Studienfach nach ihren Neigungen und Begabungen wählen (Hachmeister et al., 2023). Dies verdeutlicht, dass persönliche Neigungen und individuelle Fähigkeiten die Studienentscheidung maßgeblich beeinflussen.

Die Erkenntnis, dass sich Studieninteressierte bei ihrer Studienfachwahl von unterschiedlichen Motivationsfaktoren leiten lassen, führt zu der Vorstellung, dass es verschiedene Entscheidungstypen gibt. Intrinsische Altruisten, heimatgebundene Hedonisten, serviceorientierte Unabhängige und leistungsstarke Karriereorientierte sind einige der identifizierten Gruppen, die jeweils unterschiedliche Prioritäten bei der Fach-, Hochschul- und Ortswahl setzen. Diese unterschiedlichen Entscheidungsmuster unterstreichen die Vielfalt individueller Studienmotivationen. (Hachmeister et al., 2023)

In der Praxis bedeutet dies, dass Hochschulen wie die OTH-Regensburg darauf achten müssen, ihre Studiengänge ansprechend und überzeugend zu präsentieren. Gerade dann, wenn die Hochschule als mögliche Option in Betracht gezogen wird, spielt die Möglichkeit, das gewünschte Fach entsprechend der individuellen Interessen darzustellen, eine entscheidende Rolle.

Ein innovativer Studiengangsfinder, der Fächer inhaltlich kategorisiert und ähnliche Studiengänge präsentiert, könnte hier eine wichtige Rolle spielen. Mit einer solchen Lösung könnten Studieninteressierte effektiv über Alternativen informiert werden, insbesondere wenn der ursprünglich angestrebte Studiengang nicht verfügbar ist. Dies trägt dazu bei, dass die Hoch-

schule potenzielle Studierende auch dann überzeugen kann, wenn ihre erste Wahl nicht direkt verfügbar ist, aber dennoch ähnliche, attraktive Alternativen bietet.

2.2 Konzepte von Studiengangsfindern

Die Historie der Studiengangsfinder zeigt eine dominierende Tendenz hin zu umfragebasierten Konzepten. In diesem Zusammenhang werden Studierende durch Umfragen zu ihren Interessen, Fähigkeiten und Präferenzen befragt, um auf dieser Grundlage Studiengangsempfehlungen zu generieren.



Umfragebasierte Studiengangsfinder vertrauen auf die subjektiven Bewertungen der Nutzer und versuchen, durch direkte Befragungen der Studierenden ihre Präferenzen zu ermitteln. Die daraus resultierenden Empfehlungen basieren auf den angegebenen Interessen und Vorlieben. Jedoch sind diese Empfehlungen stark von der Qualität der gestellten Fragen und der Interpretation der Antworten abhängig. Um die bewusste Lenkung des Algorithmus zu verhindern, enthalten viele Umfrage-Tools ein Minimum von 50 Fragen. Die Umfrage enthält außerdem neben einfachen Fragen wie „Interessierst du dich für Informatik?“, sehr allgemein gehaltene (oft persönliche) Fragen. Algorithmen werten dann die Antworten auf Aussagen, wie beispielsweise „Wenn ich zur Party gehe, suche ich Kontakt mit nur wenigen, die ich kenne.“ aus und versuchen durch Zuordnung beispielsweise mithilfe des Myers-Briggs-Typenindikator (Company, 2023), passende Studiengänge zu finden. (Uniturm.de, 2023)

Beispiele für umfragebasierte Tools:

- HeyStudium von ZEIT ONLINE (<https://studiengaenge.zeit.de/>)
- Orientierungstest von Uniturm.de (<https://www.uniturm.de/studienwahltest-studiumsfinder/start>)
- OrientierungsTest vom Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg (was-studiere-ich.de)
- Studienwahltest von studieren.org (studieren.org)

Trotz ihrer weiten Verbreitung weisen umfragebasierte Ansätze gewisse Limitationen auf. Die Ergebnisse können stark von der Selbsteinschätzung der Studierenden beeinflusst sein, wodurch das Risiko von Verzerrungen oder unvollständigen Informationen besteht.

2.3 Theorie und Anwendung von interaktiven ähnlichen Systemen

Der Bereich interaktiver Systeme bildet einen bedeutenden Bestandteil der modernen Informatik und User-Experience-Forschung. Dieser Abschnitt widmet sich den theoretischen Grundlagen sowie den praktischen Anwendungen solcher Systeme und hebt ihre Relevanz für die Studienorientierung hervor. (Preim & Dachselt, 2010)

Theoretische Grundlagen: Interaktive Systeme basieren auf Konzepten aus den Bereichen Informationsvisualisierung, Algorithmik und Benutzerinteraktion. Die Theorie hinter diesen Systemen beschäftigt sich mit Methoden zur effektiven Repräsentation und Darstellung von Ähnlichkeiten zwischen verschiedenen Datenpunkten, in diesem Fall Studiengängen. Dabei werden Algorithmen wie die multidimensionale Skalierung (MDS) verwendet, um hochdimensionale Daten in eine übersichtliche und verständliche Form zu bringen. Darüber hinaus spielen Konzepte der Nutzerzentrierung eine zentrale Rolle, um sicherzustellen, dass die Systeme den Bedürfnissen und Erwartungen der Nutzer entsprechen.



Praktische Anwendungen: Der Einsatz interaktiver Systeme dieser Art erstreckt sich über verschiedene Bereiche, von Produktempfehlungen in E-Commerce-Plattformen bis hin zur Visualisierung von Beziehungen in sozialen Netzwerken. Im speziellen Kontext der Studienorientierung ermöglichen solche Systeme eine ansprechende und nutzerfreundliche Darstellung von Studiengängen auf Basis inhaltlicher Ähnlichkeiten. Die Visualisierung dient Studieninteressierten als unterstützendes Werkzeug, um alternative Studienoptionen leichter zu erkunden und fundierte Entscheidungen bei der Studienwahl zu treffen.

Der Einsatz derartiger interaktiver Systeme im Bereich der Studienorientierung bietet somit einen innovativen Ansatz, um Studiengänge nutzergerecht zu präsentieren. Die Kombination aus theoretischer Fundierung und praxisorientierter Anwendung schafft die Grundlage für die Entwicklung eines effizienten und benutzerfreundlichen Studiengangsfilters, wie er im weiteren Verlauf dieser Arbeit detailliert vorgestellt und diskutiert wird.



3 Methodik

3.1 Beschreibung der Algorithmen

Der Studiengangsfinder soll Studieninteressierten einen schnellen Überblick über alle in Frage kommenden Studiengänge ermöglichen. Dabei soll dem Nutzer eine interaktive Grafik präsentiert werden, mit der er anhand von Studieninhalten (z.B. „Gesundheit und Soziales“) sofort alle relevanten Studiengänge findet. Dazu ist es notwendig die Studiengänge nach ihren Inhalten zu gruppieren (Clustering) und schließlich visuell ästhetisch aufzubereiten.

Bei der Festlegung des Clustering-Algorithmus für den Studiengangfinder wurden verschiedene Optionen in Betracht gezogen, darunter K-Means Clustering, Force-Directed Layouts und Multidimensionale Skalierung (MDS). Nach einer gründlichen Abwägung der Vor- und Nachteile fiel die Wahl auf MDS. Die Gründe für diese Entscheidung und eine Erläuterung der jeweiligen Algorithmen werden in den folgenden Kapiteln gegeben.

3.1.1 Force-Directed-Graphs

Force-Directed Graph Drawing ist eine Methode zur Visualisierung von Graphen, bei der die Positionen der Knoten und Kanten aufgrund von Kräften bestimmt werden. Das Vorgehen hierbei ist vereinfacht inspiriert von Modellen der Teilchenphysik und wird häufig mit dem Verhalten von Federn verglichen. Ziel des Algorithmus ist es durch Kanten verbundene Knoten nah beinander zu platzieren und somit eine ästhetisch ansprechende Visualisierung eines Graphen zu berechnen. In Abbildung 1 erkennt man ausgehend von einer zufälligen Positionierung (Zustand 0), eine schrittweise Optimierung der Darstellung. Wie stark oder schwach sich die Knoten jeweils „anziehen“ bzw. „abstoßen“ wird durch die Gleichmäßigkeit der Verteilung auf der sogenannten Zeichenfläche bestimmt. (**force-directed-layouts**)

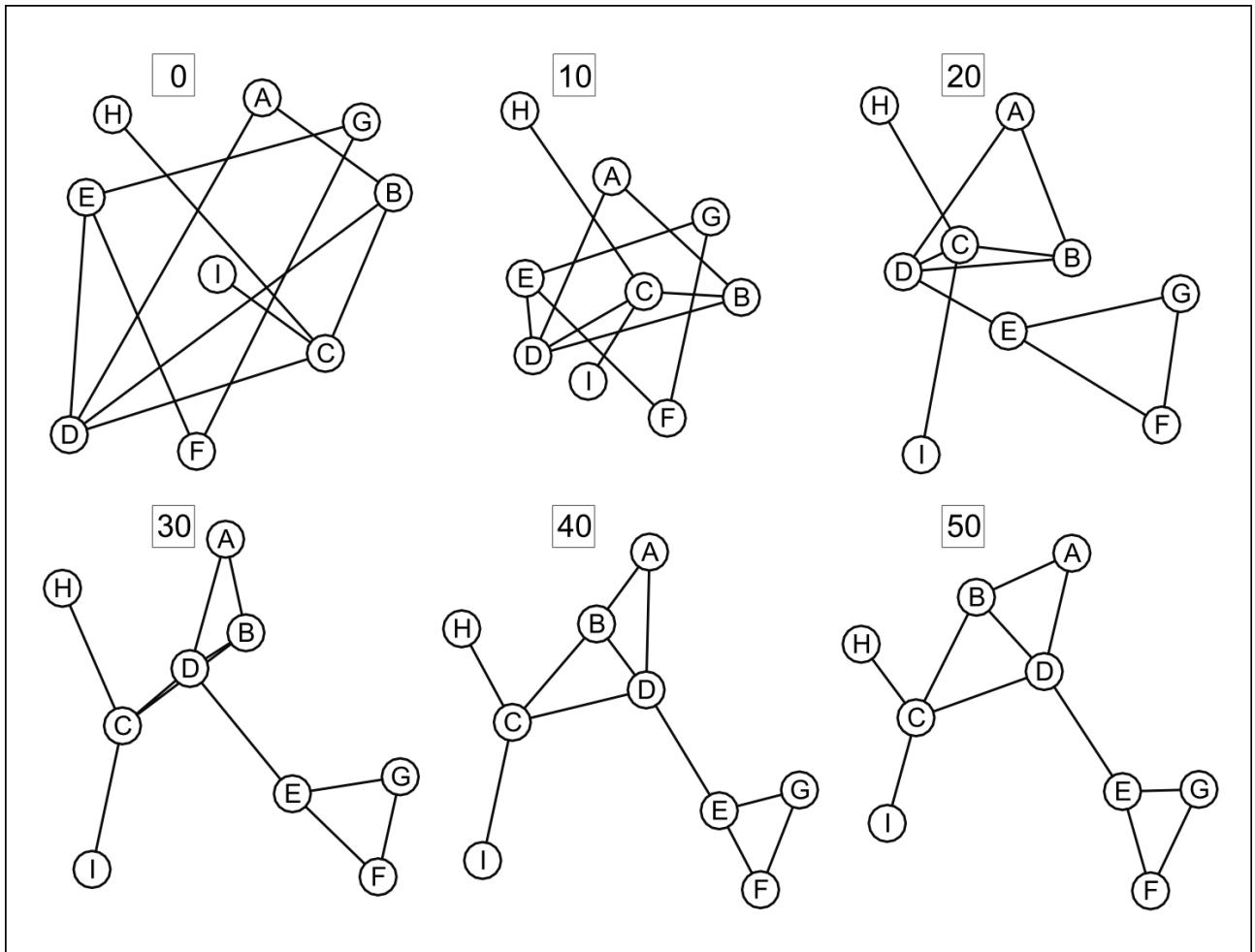


Abbildung 1: Schrittweises Durchführen des Layout-Algorithmus von Fruchterman/Reingold
Quelle: Fruchterman, Thomas M. J./Reingold, Edward M.: Graph Drawing by Force-Directed Placement

Force-Directed Layouts sind besonders effektiv für die übersichtliche Darstellung von Netzwerken, in denen die Verbindungen zwischen den Elementen im Vordergrund stehen. (**force-directed-layout**) Im Gegensatz dazu erfordert der Studiengangsfinder eine kontinuierliche Positionierung der Studiengänge basierend auf inhaltlichen Ähnlichkeiten, was nicht unbedingt der Stärke von Force-Directed Layouts entspricht. Force-Directed Graph Drawing benötigt als Voraussetzung bereits einen Graphen mit Knoten und Kanten, welche im Fall des Studiengangsfinders die Ähnlichkeiten zwischen den einzelnen Studiengängen entsprechen würde. Gerade die Berechnung der Ähnlichkeit zwischen den einzelnen Studiengängen ist jedoch wesentlicher Bestandteil dieser Arbeit, weshalb der Algorithmus nicht näher untersucht wurde.

3.1.2 K-Means Clustering Algorithmus

K-Means ist ein Clustering-Algorithmus, der Datenpunkte in k vordefinierte Gruppen oder Cluster einteilt. Die Wahl von K stellt die Anzahl der Cluster dar, und der Algorithmus versucht, die Datenpunkte so zu gruppieren, dass die Varianz innerhalb der Cluster minimiert wird (siehe Abbildung 2). (**kmeans**)

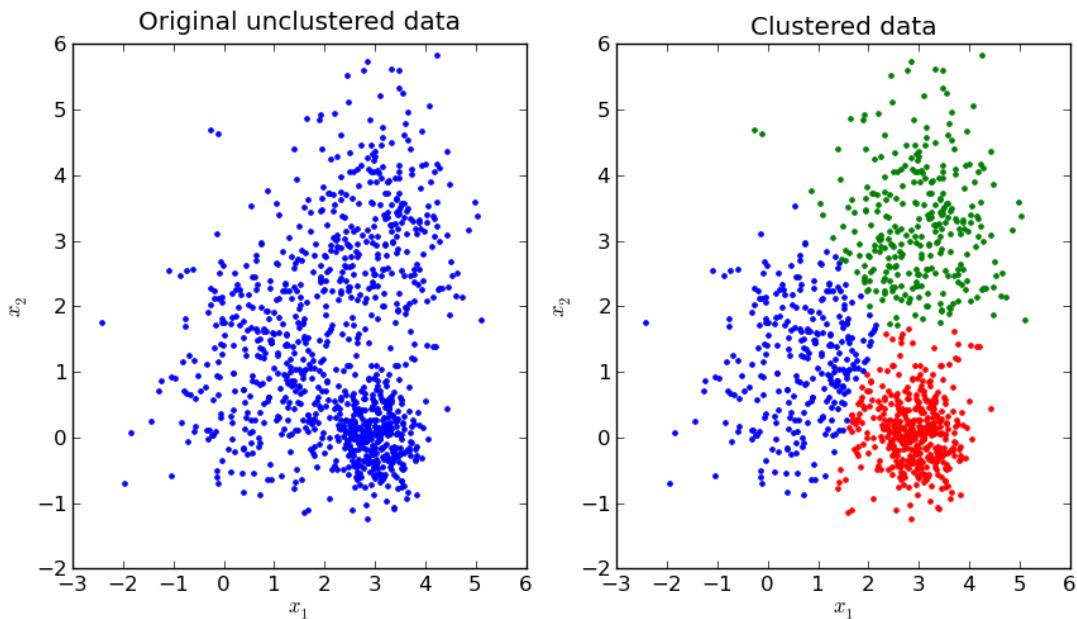


Abbildung 2: K-Means Clustering Algorithmus
Quelle: <https://mubaris.com/posts/kmeans-clustering/>

Die Entscheidung, den K-Means-Algorithmus nicht zu verwenden, basiert darauf, dass dieser hauptsächlich darauf abzielt, Datenpunkte zu gruppieren und weniger auf deren präzise Positionierung in einem zweidimensionalen Raum. Der Algorithmus fordert außerdem eine vordefinierte Clusteranzahl k . Das stellt im Falle des Studiengangsfilters jedoch keine Einschränkung dar, da jeder Cluster eine Inhaltskategorie, wie zum Beispiel „Informatik“ repräsentieren würde. Trotzdem erweist sich der K-Means Clustering Algorithmus als unpassend, da er sehr empfindlich gegenüber Ausreißern ist, da er versucht, Clusterzentren zu finden, die die Gesamtvarianz minimieren. Wenn es Ausreißer in den Studienschwerpunkten gibt, könnten sie das Ergebnis beeinflussen. (**kmeans-spikes**)

Insgesamt erfüllt der MDS-Algorithmus die spezifischen Anforderungen des Studiengangsfilters, indem er eine übersichtliche und interpretierbare Visualisierung der Studiengänge basierend auf inhaltlichen Ähnlichkeiten ermöglicht.

3.1.3 Multidimensionale Skalierung (MDS)

MDS ermöglicht die Reduktion n -dimensionaler Daten auf m -Dimensionen, wodurch eine anschauliche Darstellung in Form von Koordinatenpaaren ermöglicht wird ([intro-to-multidimensional-scaling](#)). Dieser Aspekt ist entscheidend, um Studiengänge in einem zweidimensionalen Diagramm zu positionieren, wobei ähnliche Studiengänge aufgrund ihrer inhaltlichen Ähnlichkeiten nahe beieinander liegen. Die Anwendung des MDS-Algorithmus auf eine genormte Tabelle, in der im Falle von StudyMap die Studiengänge nach ihren Anteilen an verschiedenen Inhaltenkategorien gewichtet sind, ermöglicht eine effektive Positionierung im Diagramm (siehe Tabelle 1).

Kürzel	Architektur	Gesundheit	Technik	Informatik	Wirtschaft	Internat.
AT	0,55	0,06	0,09	0,04	0	0
B	0,75	0	0	0,1	0	0,01
ID	0,1	0,05	0,15	0,05	0,05	0
HK	0,01	0,7	0,01	0,01	0,1	0,05
PA	0	0	0,6	0,2	0,12	0,02
IE	0	0	0,34	0,32	0,22	0,04
LP	0	0,89	0,01	0	0,04	0,02
SA	0	0,98	0	0	0	0
IN	0	0	0,05	0,9	0	0,2
IW	0	0	0,05	0,75	0,15	0,2

Tabelle 1: Exemplarische Eingabetabelle für den MDS-Algorithmus

Tabelle 1 stellt eine stark vereinfachte Eingabetabelle für den MDS-Algorithmus dar. Die erste Spalte enthält die Kürzel der verschiedenen Studiengänge, wobei beispielsweise *AT* für den Bachelor-Studiengang Architektur steht. Die folgenden Spalten enthalten genormte Werte zwischen 0 und 1, die die prozentualen Anteile verschiedener Inhaltenkategorien in den jeweiligen Studiengängen repräsentieren. Diese genormten Werte werden im n -dimensionalen Raum positioniert. Durch Anwendung des MDS-Algorithmus werden im Verlauf die n -Dimensionen auf zwei Dimensionen reduziert, um schließlich eine ästhetisch ansprechende Visualisierung in Form eines Diagramms zu generieren.

Der klassische MDS-Algorithmus besteht aus den folgenden vier Schritten ([mds-for-dimensionality-reduction](#))

1. Berechnung der Distanzmatrix D_{ij}

Die Berechnung der Distanzmatrix D_{ij} erfolgt mithilfe des euklidischen Abstands zwischen den Studiengängen im n -dimensionalen Raum. Der euklidische Abstand zwischen zwei Punkten P_i und P_j wird nach der Formel

$$d(P_i, P_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \dots + (z_i - z_j)^2}$$

berechnet, wobei x_i, y_i, \dots, z_i die Koordinaten von Punkt P_i im n-dimensionalen Raum sind (**mds-ceopedia**). Für die Studiengangsdaten bedeutet das, dass die n-Dimesionen die verschiedenen Inhaltskategorien repräsentieren.



Die euklidische Distanzmatrix D_{ij} enthält dann die euklidischen Abstände zwischen jedem Paar von Studiengängen. In der Matrix **si** die Elemente D_{ij} die Distanzen zwischen den Studiengängen i und j . Je näher die Studiengänge in der Distanzmatrix beieinander liegen, desto näher werden sie im finalen Diagramm platziert und umgekehrt.

Konkret in Python implementiert, sieht die Berechnung der Distanzmatrix wie folgt aus:

```

1 def calculate_distance_matrix(X):
2     euclidean = lambda x,y:ma.sqrt(np.sum((np.array(x)-np.array(y))**2))
3     D = []
4     for x in X:
5         tmp = []
6         for y in X:
7             tmp.append(euclidean(x, y))
8         D.append(tmp)
9     return D

```

2. Anwendung der Centering Matrix C zur Normalisierung der Distanzen



Die Formel $C = I - \frac{1}{n}\vec{e} * \vec{e}^T$ berechnet die sogenannte Centering Matrix. I ist die Einheitsmatrix, $\frac{1}{n}$ ist der Kehrwert der Anzahl der Datenpunkte und \vec{e} ist ein Vektor gefüllt mit Einsen. Das Symbol \vec{e}^T bezeichnet wie üblich die Transposition des Vektors. (**an-introduction-to-mds**)

Die Centering Matrix C wird verwendet, um die Distanzmatrix D zu zentrieren. Das Zentrieren ist wichtig, um die Distanzen zwischen den Punkten in der n-dimensionalen Raummatrix zu normieren und somit den Schwerpunkt der Daten im Raum zu korrigieren. (**an-introduction-to-mds**)



Schließlich wird diese eingesetzt um die Zentriermatrix B aus der Distanzmatrix zu berechnen:

$$B = -\frac{1}{2} * C * D_{ij} * C$$

3. Spektralzerlegung



Die Matrix B wird nun spektral zerlegt, um die Eigenwerte λ_i und die zugehörigen Eigenvektoren v_i zu erhalten.

$$B = W * \Lambda * W^{-1}$$

Hierbei ist W die Matrix der Eigenvektoren und Λ ist eine Diagonalmatrix mit den Eigenwerten auf der Hauptdiagonale. Die Eigenvektoren werden anschließend sortiert und die größten positiven Eigenwerte $\lambda_1 \dots \lambda_m$ mit dazugehörigen Eigenvektoren $v_1 \dots v_m$ aus B extrahiert. (**an-introduction-to-mds**)

4. Projektion der Datenpunkte



Um nun die Datenpunkte von einem höherdimensionalen Raum (basierend auf den Beziehungen in der Distanzmatrix D) auf einen niedrigdimensionalen Raum, der durch die Eigenvektoren und Eigenwerte repräsentiert wird, abzubilden, benötigt man folgende Projektion (**intro-to-multidimensional-scaling**):

$$X = V_m \Lambda_m^{1/2}$$

Die Variable m steht für die Anzahl der gewünschten Dimensionen. Im Falle von StudyMap entspricht $m = 2$, um das Ergebnis in einem 2D-Diagramm zu visualisieren. V_m steht für die Eigenvektoren und Λ wie bereits im vorherigen Absatz beschrieben für die Diagonalmatrix mit den Eigenwerten. (**an-introduction-to-mds**)

Abschließend enthält X die Matrix mit den auf m -Dimensionen reduzierten Koordinaten, welche dann z.B. in einem Diagramm visualisiert werden können. An dieser Stelle wird für die Berechnung der Studiengänge (mit $m = 2$) ermöglicht, dass innerhalb einer 2D-Darstellung, ähnliche Studiengänge aufgrund ihrer inhaltlichen Ähnlichkeiten nahe beinander positioniert werden. Diese räumliche Anordnung erleichtert die intuitive Analyse von Beziehungen zwischen den einzelnen Bachelor- und Masterstudiengängen.

3.2 Beschreibung der Datenquelle und -beschaffung

3.2.1 Herkunft der Daten

Die Grundlage für den Studiengangsfinder wurde in enger Zusammenarbeit mit Frau Rösel, der Vizepräsidentin der Hochschule Regensburg, gelegt. Wir initiierten den Prozess durch die Definition klarer Anforderungen an die benötigten Informationen. Dazu gehört unter anderem die Definition der Inhaltskategorien der Studiengänge. Der erste Entwurf entsprach folgender Aufteilung:

1. Architektur und Bau
2. Design und Medien
3. Gesundheit und Soziales
4. Technik
5. Informatik und Mathematik
6. Marketing und Kommunikation
7. Erneuerbare Energien, Nachhaltigkeit und Umwelttechnik
8. Wirtschaft und Management
9. Internationales

Diese Anforderungen wurden dann innerhalb der Hochschule kommuniziert und von einer Teilmenge der Studiendekane für vereinzelte Studiengänge ausgefüllt. Das heißt konkret, dass für 22 Studiengänge jeweils für alle dieser Inhaltskategorien ein Wert festgelegt wurde.

Frau Rösel bleibt bis zur Übergabe der Arbeit weiterhin für die Beschaffung der Daten verantwortlich. Eine systematische Bewertung aller Studiengänge anhand des Modulplans und der jeweiligen ECTS pr+nhaltskategorie wird von Frau Rösel in Zusammenarbeit mit den Studiengangsverantwortlichen entwickelt.

Das European Credit Transfer System (ECTS) ist ein System zur Normierung von Studienleistungen innerhalb des Europäischen Hochschulraums. Dadurch können Unterschiede zwischen nationalen Hochschulsystemen ausgeglichen werden. (Commission, n. d.) Jedes Fach an

der OTH Regensburg hat eine bestimmte Anzahl von Credits (ECTS), die sich nach der Intensität des Faches richtet.

 Die Bewertung des Studiengangsfinders erfolgt auf Basis der Inhaltstypen und wird nun neutral nach der ECTS-Anzahl der jeweiligen Kategorie berechnet. So wird eine objektive Einstufung der Studiengänge für den verwendeten MDS-Algorithmus gewährleistet. Durch objektive Berechnungen ist es in Zukunft denkbar, die Aufgabe der Datenaktualisierung an eine dritte Person zu delegieren.

3.2.2 Festlegung der Inhaltstypen

Im nächsten Schritt werden  die Werte auf die Zeile normiert. Das heißt, die Summe aller Inhaltstypen-Werte pro Studiengang beträgt eins.

Im Umkehrschluss bedeutet das , dass die Studiengänge aufgrund von Überschneidungen in den Kategorien nicht korrekt abgebildet werden können. Beispiel:

Studiengang	...	Informatik und Mathematik	Internationales	...
Informatik	...	0,8	0,2	...
International Computer Science	...	0,4	0,6	...

Tabelle 2: Aufteilung der Werte bei auf Studiengang genormte Werte

Tabelle 2 zeigt die Problematik der Überschneidungen. Der Studiengang International Computer Science ist von den Inhalten nahezu identisch zum Studiengang Informatik. Der Hauptunterschied ist, dass die Fächer in Englisch angeboten werden. Dies hat zur Folge, dass der Studiengang eigentlich sowohl in der Kategorie Informatik und Mathematik, als auch in der Kategorie Internationales einen hohen Wert benötigt. Da die Zeilensumme eins beträgt, ist dies nicht realistisch abbildbar. Aus diesem Grund wurde entschieden die Werte pro Zelle, d.h. pro Kategorie und Studiengang auf eins zu normieren. Mit dieser Aufteilung, kann ein Studiengang sowohl in Informatik und Mathematik beispielweise einen Wert von 0,8 haben, als auch in der Kategorie Internationales.

Die anfängliche Kategorisierung der Studiengänge in neun allgemeine Inhaltstypen stellte sich als unzureichend heraus, da nach der Anwendung des MDS-Algorithmus viele Studiengänge überlappend dargestellt wurden. Diese Herausforderung führte zu einer entscheidenden Überarbeitung des Kategorisierungssystems, um präzisere und differenzierte Bewertungen zu ermöglichen. Die Lösung bestand in der Einführung von sogenannten Supergruppen, die eine tiefere Bewertung der Studiengänge ermöglichten.

 Ursprünglich waren Studiengänge wie *Architektur* und *Bauingenieurwesen* in einer einzigen Kategorie *Architektur und Bau* zusammengefasst. Die Neuerung bestand darin, diese in separate Kategorien wie *Architektur* und *Bau* zu unterteilen. Ein zusätzliches Feld in der Eingabedatei legte fest, welche dieser Kategorien später in der Benutzeroberfläche zu einer Supergruppe zusammengeführt werden sollten. Dieser Ansatz ermöglichte eine präzisere Bewertung von Studiengängen, insbesondere bei Studiengängen wie *Architektur* und *Bauingenieurwesen*. Hier konnte der Studiengang *Architektur* als mehr *Architektur* und weniger *Bau* bewertet werden, während es bei *Bauingenieurwesen* genau umgekehrt war. Vor dieser Anpassung konnte nur ein Wert für *Architektur und Bau* festgelegt werden.

Um die Benutzeroberfläche übersichtlich zu halten, wurden Supergruppen eingeführt. Diese ermöglichen eine aggregierte Darstellung mehrerer Kategorien, ohne die Benutzeroberfläche unnötig ~~zu~~ komplex zu gestalten. Frau Rösel brachte entscheidende Impulse in diesen Prozess ein. Durch ihre Rückmeldung wurden darüber hinaus neue Kategorien wie *Sprachkompetenzen*, *Digitalität* und *Future Skills* eingeführt. Diese dienen dazu, eine noch präzisere Differenzierung und Bewertung der Studiengänge zu ermöglichen. Folgende Liste zeigt die neuen Kategorien - jeder Listeneintrag enthält dabei eine Supergruppe mit einem oder mehreren Kategorien:

1. Architektur, Bau (Architektur und Bau)
2. Design, Medien (Design und Medien)
3. Gesundheit, Soziales (Gesundheit und Soziales)
4. Maschinenbau, Elektrotechnik (Technik)
5. Informatik (Informatik)
6. Naturwissenschaften, Mathematik (Naturwissenschaften und Mathematik)
7. Marketing, Kommunikation (Marketing und Kommunikation)
8. Erneuerbare Energien, Umwelttechnik (Erneuerbare Energien und Umwelttechnik)
9. Nachhaltigkeit (Nachhaltigkeit)
10. Wirtschaft, Management (Wirtschaft und Management)
11. Sprachkompetenzen (Sprachkompetenzen)
12. Digitalität (Digitalität)
13. Future Skills (Future Skills)

Die neuen Kategorien, ergänzt durch die Einführung der Supergruppen, schaffen eine optimierte Grundlage für den MDS-Algorithmus. Dieser kann nun Studiengänge präziser positionieren, da die Unterscheidungen und Bewertungen auf einer feineren Ebene vorgenommen werden. Diese Anpassungen tragen entscheidend dazu bei, das Ziel einer verfeinerten Studienorientierung und -wahl zu erreichen.

4 Konzept eines innovativen Studiengangsfinders

4.1 Vorstellung der OTH-Regensburg als Fallbeispiel

Die Ostbayerische Technische Hochschule Regensburg (OTH-Regensburg) dient in dieser Masterarbeit als praxisorientiertes Fallbeispiel für die Konzeption und Umsetzung eines innovativen Studiengangsfinders.



4.1.1 Über die OTH-Regensburg

Die OTH-Regensburg, gegründet im Jahr 1971, ist eine der größten Hochschulen für angewandte Wissenschaften in Deutschland. Mit ihrem breiten Spektrum an praxisorientierten Studiengängen und ihrer engen Verbindung zur Industrie bietet die OTH-Regensburg eine ideale Umgebung für die Entwicklung und Implementierung eines innovativen Studiengangsfinders. (Regensburg, 2023)



Die Hochschule zeichnet sich durch eine moderne Infrastruktur, hochqualifizierte Dozenten und eine enge Zusammenarbeit mit regionalen Unternehmen aus. Mit mehr als 50 Bachelor- und Masterstudiengängen in den Bereichen Ingenieurwissenschaften, Naturwissenschaften, Wirtschaft und Sozialwissenschaften bietet die OTH-Regensburg eine breite Palette an Studienmöglichkeiten. (Regensburg, 2023)

4.1.2 Herausforderung und Zielsetzung

Wie viele Bildungseinrichtungen steht auch die OTH-Regensburg vor der Herausforderung, ihre Vielzahl an Studiengängen für Studieninteressierte transparenter und zugänglicher zu machen. Der Studiengangsfinder, der im Rahmen dieser Masterarbeit entwickelt wird, soll dazu beitragen, potenziellen Studierenden eine klarere Orientierung über die verfügbaren Studienmöglichkeiten zu bieten und ihre Entscheidungsfindung zu unterstützen.

Durch die Anwendung des Studiengangsfinders StudyMap an der OTH-Regensburg wird nicht nur ein innovatives Werkzeug für Studieninteressierte geschaffen, sondern auch die Effektivität und Transparenz der Studienorientierung an der Hochschule selbst verbessert. Die Integration und algorithmische Verarbeitung von Daten zu Studiengängen und Schwerpunkten ermöglicht eine präzisere Darstellung der Studienvielfalt und fördert gleichzeitig die strategische Ausrichtung der Hochschule.

4.2 Konzept für eine automatisch generierte Infografik

Die Grundlage für das Konzept der automatisch generierten Infografik bildet die methodische Vorarbeit, die in den vorhergehenden Kapiteln detailliert erläutert wurde. Im Zuge der Methodik, wie in Kapitel 3 beschrieben, wurden Studiengänge anhand verschiedener Kategorien bewertet und mithilfe des MDS-Algorithmus in zweidimensionale Koordinaten transformiert. Das daraus resultierende Zahlenmaterial bildet die Grundlage für die Visualisierung in der Infografik. In den folgenden Unterkapiteln wird ein erstes Designkonzept für die Software vorgestellt.

4.2.1 Visualisierung der Studiengänge

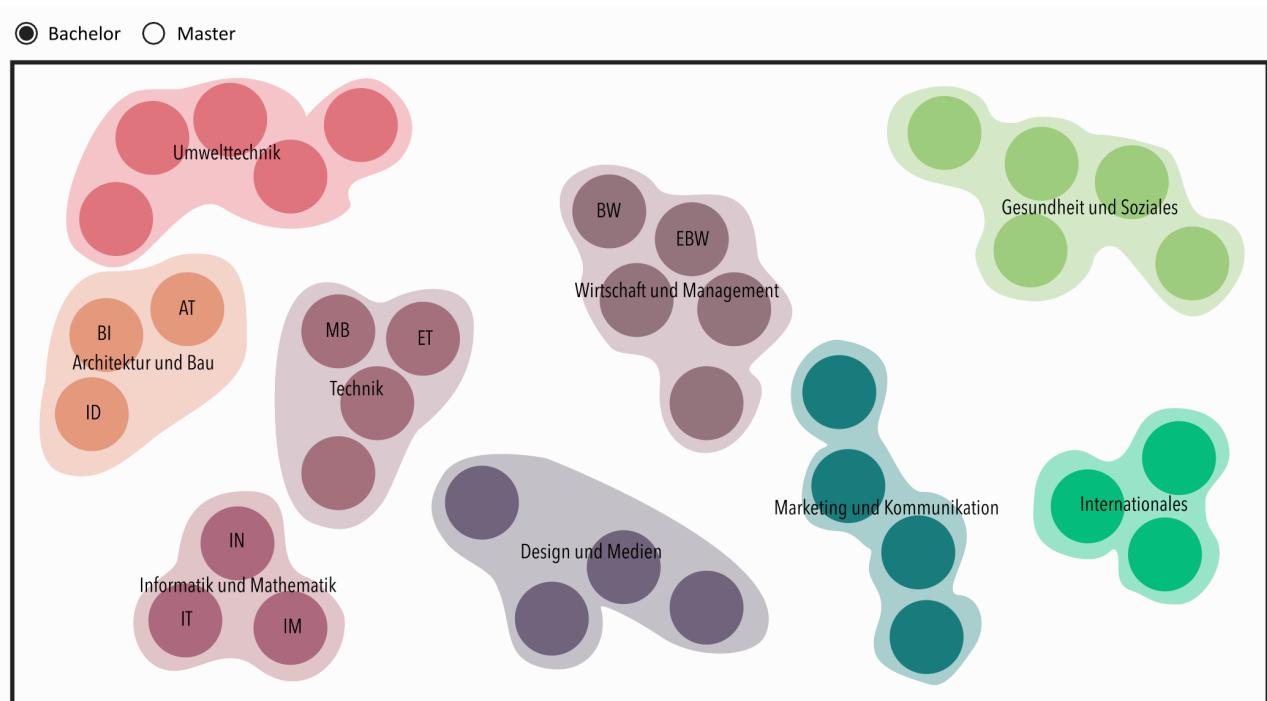


Abbildung 3: Erstes Mockup einer möglichen Implementierung
Quelle: Eigene Darstellung

Die automatisch generierte Infografik präsentiert die Vielfalt der Studiengänge auf eine möglichst anschauliche Weise. Dabei werden inhaltlich ähnliche Studiengänge in Form von sogenannten „Bubbles“ in unmittelbarer Nähe zueinander positioniert (siehe Abbildung 3). Die Bubbles enthalten jeweils die Kürzel der Studiengänge. Exemplarisch enthalten in Abbildung 3 nur einige wenige Bubbles ein Kürzel. Durch die Verwendung von PixiJS kann die resultierende Grafik dynamisch in allen Browsern, die WebGL und Canvas unterstützen dargestellt werden.

4.2.2 Berücksichtigung von Schwerpunkten

Ein wesentlicher Aspekt des Infografik-Konzepts ist die Berücksichtigung von Schwerpunkten. Die Supergruppen, die in der Datenverarbeitung definiert wurden, werden hier als leichter Schleier um die darin beinhalteten Studiengänge präsentiert (siehe Abbildung 3). Außerdem werden alle Supergruppen mit einer zentrierten Beschriftung versehen. Dies ermöglicht es den Nutzern, gezielt nach Studiengängen innerhalb bestimmter Kategorien zu suchen und ihre Präferenzen entsprechend auszurichten.

4.2.3 Interaktivität

Die Infografik wird durch interaktive Elemente angereichert, um den Nutzern eine individuelle Erkundung der Daten zu ermöglichen. Beispielsweise können durch Mausklicks zusätzliche Informationen zu einem Studiengang abgerufen werden.

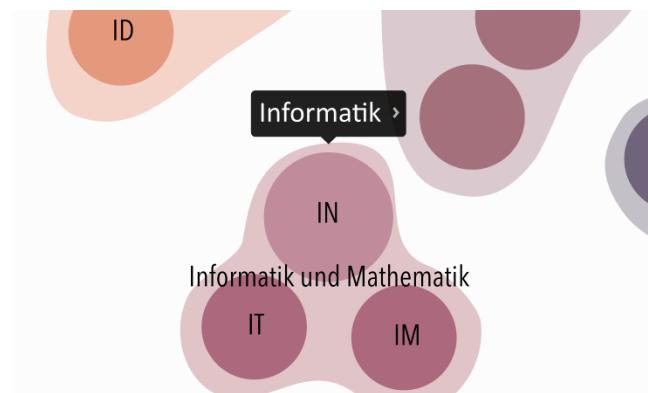


Abbildung 4: Mockup: Tooltip über Bubble
Quelle: Eigene Darstellung

Sobald ein Benutzer auf einem Desktop-Gerät (z.B. Laptop) mit der Maus über eine Bubble fährt, erscheint ein Tooltip mit dem vollständigen Namen des Studiengangs (siehe Abbildung 4). Ein Tooltip ist ein grafisches Element, das kurze Textinformationen liefert. Aufgrund der fehlenden Maus auf mobilen Endgeräten, erscheint der Tooltip darauf erst beim Klick auf eine Bubble.

Möchte der Studieninteressierte mehr über den Studiengang erfahren, bietet das Konzept zusätzlich die Möglichkeit, ein Popup mit Details zu öffnen (siehe Abbildung 5). Bei Desktop-Geräten genügt ein einfacher Klick auf die Sprechblase oder den sich öffnenden Tooltip, bei mobilen Endgeräten ist ein zweiter Klick auf die jeweiligen Elemente erforderlich.



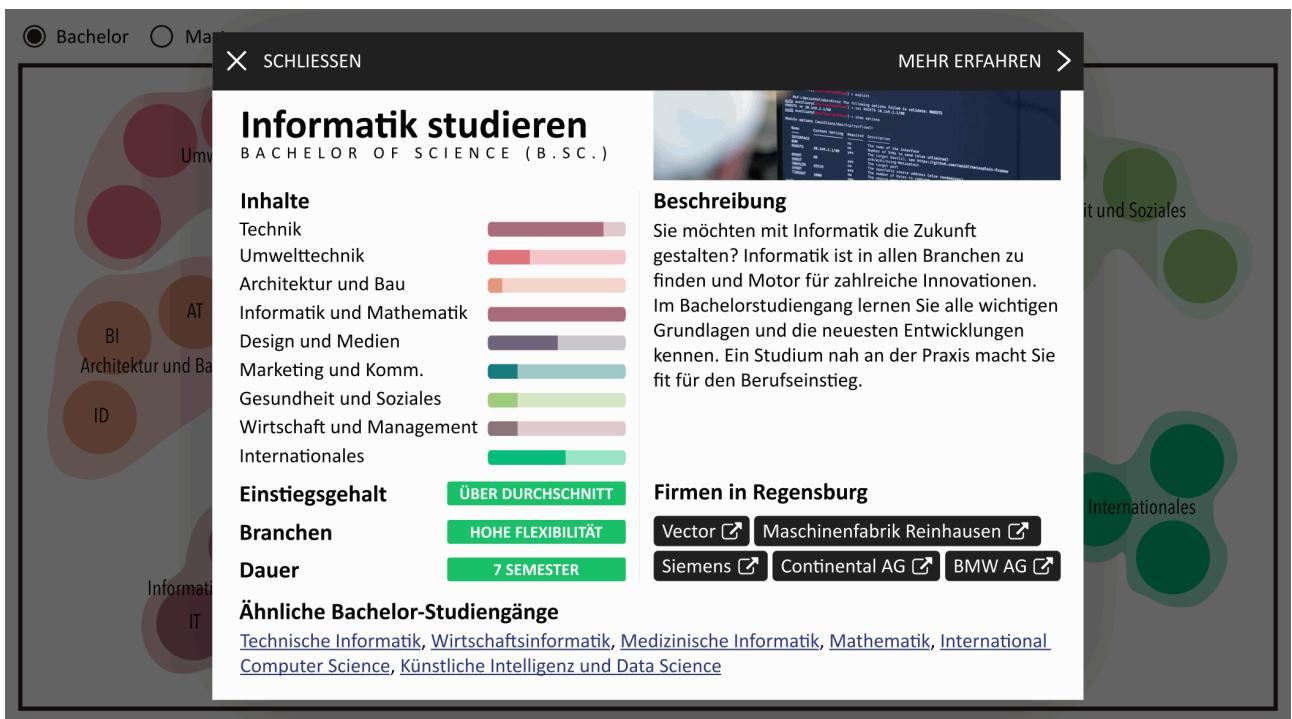


Abbildung 5: Mockup: Popup mit Details eines Studiengangs
Quelle: Eigene Darstellung

Im Folgenden wird das Konzept des Popup-Dialogs beschrieben, das in Abbildung 5 visualisiert ist.

Innerhalb des Popups findet der Nutzer eine prägnante Kurzbeschreibung des Studiengangs. Die Inhalte werden durch Fortschrittsbalken visualisiert, die den Anteil der verschiedenen Kategorien darstellen. So kann beispielsweise die Kategorie „Informatik“ zu 90% ausgefüllt sein, um den Schwerpunkt des Studiengangs zu verdeutlichen.

Zusätzlich liefert das Popup Informationen über Einstiegsgehälter, relevante Branchen, Studiendauer und ähnliche Studiengänge. Letztere werden nicht statisch dargestellt, sondern vom Algorithmus dynamisch berechnet, um stets aktuelle und präzise Informationen zu liefern.

Außerdem enthält der Dialog eine Liste von Unternehmen in Regensburg, die gezielt nach Absolventen dieses Studiengangs suchen. Dieser Aspekt gibt dem Nutzer einen Einblick in potentielle Arbeitgeber und eröffnet berufliche Perspektiven direkt in der Region.

Durch die Integration dieses Fensters wird die Benutzerfreundlichkeit erheblich verbessert, da der Nutzer einen tieferen Einblick in die einzelnen Studiengänge erhält, ohne die Seite wechseln zu müssen.



4.3 Datenpflege und -sicherung

Um StudyMap stets mit aktuellen Daten betreiben zu können, müssen diese gepflegt werden. Wie in Unterabschnitt 3.2.1 erläutert, stammen die Daten bisher von der Vizepräsidentin der OTH-Regensburg. Bis zur Übergabe des Projekts wird das auch so bleiben.

4.3.1 Verwaltungssoftware vs. Anleitung

Aufgrund des technischen Aufwands beim Einpflegen der Daten per SSH- und SCP-Verbindung sowie dem manuellen Aufruf des Python-Skripts zur Generierung der Positionsdaten wird hier eine Unterstützung benötigt. Während SSH (Secure Shell) dazu da ist eine verschlüsselte Verbindung zur Kommandozeile des Servers aufzubauen, ermöglicht SCP (Secure Copy) den sicheren Austausch von Dateien mit dem Server.

Bei einer manuellen Aktualisierung der Daten über SCP müssen die neuen Dateien auf den Server kopiert werden. Anschließend ist das Python-Skript mit den richtigen Parametern aufzurufen, um den Prozess zu starten. Um die im nächsten Abschnitt beschriebene Produktiv- und Stagingumgebung einzuhalten, sind außerdem verschiedene Ordnerstrukturen zu beachten.

Der technische Hintergrund der Person, die das Projekt in Zukunft betreuen wird, ist ungewiss. Daher sollte ein Verwaltungstool zur Vereinfachung des Pflegeprozesses entwickelt werden. Die Alternative einer Anleitung in Form eines PDF-Dokuments könnte aufgrund der Komplexität der Vorgänge schnell zu kompliziert und umständlich werden. Des Weiteren erhöht sich bei jedem manuell zu erledigenden Schritt die Wahrscheinlichkeit für Fehler.



4.3.2 Staging- und Produktivumgebung

Im diesem Abschnitt wird der Bedarf einer Trennung zwischen Staging- und Produktivumgebung erläutert. Unter einer Staging-Umgebung versteht man in der Informatik ein Testsystem, welches unter annähernd realen Bedingungen erlaubt die Anwendung zu testen - meist ist diese nur über einen speziellen Zugang für authentifizierte Personen erreichbar. Die Produktivumgebung hingegen bezeichnet die für die Öffentlichkeit freigegebene Version der Software.



Für einen Menschen ist der Prozess der Positionsberechnung nicht deterministisch nachvollziehbar, daher ist eine Testumgebung erforderlich. Im ungünstigsten Fall überlappen sich die Bubbles im Canvas bei vorheriger ähnlicher Bewertung der Studiengänge. Für diesen Fall muss eine Staging-Umgebung zur Verfügung stehen, auf der ein neuer Datensatz getestet werden kann. Sobald dieser optimiert und von den verantwortlichen Personen freigegeben wird, sollte

der Staging-Datensatz per Knopfdruck in die Produktivumgebung überführt werden können.

Dieser zweistufige Prozess erhöht die Fehlersicherheit und ermöglicht es, neue Berechnungen zu testen, ohne sie direkt der Öffentlichkeit zur Verfügung zu stellen.

4.3.3 Datensicherung

Neben der im vorigen Abschnitt erläuterten Trennung von Staging- und Produktivumgebung ist ein weiterer wesentlicher Bestandteil des Konzepts die Datensicherung. Sollte trotz Stagingumgebung ein Fehler auftreten, müssen die alten Daten wiederhergestellt werden können.

Für diesen Fall sieht das Konzept vor, dass bei jeder Produktivschaltung der Staging-Version ein Backup erstellt wird. Dieses Backup sollte im Verwaltungstool per Knopfdruck in die Staging-Umgebung zurückgespielt werden können. Von dort aus kann es begutachtet und schließlich wieder produktiv geschaltet werden.

Durch diese doppelte Sicherung sollte ein Großteil der versehentlich eingelesenen ungültigen Daten vermieden werden. Zusätzlich muss es immer möglich sein, den aktuell verwendeten Datensatz herunterzuladen. Wenn eine neu zuständige Person den aktuellen Datensatz bearbeiten möchte, kann sie so den aktuellen Stand herunterladen und bearbeiten, ohne die vorher zuständige Person kontaktieren zu müssen. Darüber hinaus steht im Idealfall immer ein sogenannter Template-Datensatz zum Download zur Verfügung, der im Falle einer korrupten Datensicherung und einer kaputten Staging- oder Produktivversion einen validen Datenstand als Basis liefert.

5 User-centered Design Research

5.1 Mockup-Studie: Bewertung der Benutzerfreundlichkeit und Funktionalität

Bei der Entwicklung eines nutzerzentrierten Designs steht die direkte Einbindung der Zielgruppe im Mittelpunkt. Dieser Ansatz, der als User-Centered Design (UCD) bezeichnet wird, stellt die Integration der Bedürfnisse, Erwartungen und Anforderungen der Benutzer in den Entwicklungsprozess in den Mittelpunkt. Ziel ist es, Produkte und Anwendungen zu gestalten, die nicht nur funktional und ästhetisch ansprechend sind, sondern auch die bestmögliche Benutzerfreundlichkeit bieten. 

5.1.1 Methodik und Teilnehmende

Im Rahmen des UCD-Prozesses wurde das zuvor in Kapitel 4.2 vorgestellte Mockup-Konzept für den Studiengangsfinder entwickelt, das als Prototyp diente. Um sicherzustellen, dass das entworfene Konzept den Bedürfnissen der potenziellen Nutzer entspricht, wurde eine gezielte User-Centered-Design-Studie durchgeführt. Zu diesem Zweck wurden sechs Studierende und Absolventen verschiedener Fachrichtungen eingeladen, den Mockup-Entwurf zu evaluieren (siehe Tabelle 3).

#	Studiengang	Geschlecht	Datum
1	Betriebswirtschaft B.A.	Weiblich	16.09.2023
2	Psychologie M.Sc.	Weiblich	20.09.2023
3	Informatik B.Sc.	Männlich	25.09.2023
4	Regenerative Energietechnik und Energieeffizienz B.Eng.	Männlich	07.10.2023
5	Bioprozessinformatik B.Sc.	Männlich	08.10.2023
6	Betriebswirtschaft B.A.	Weiblich	12.10.2023

Tabelle 3: Probanden der Mockup-Studie

5.1.2 Ablauf des Tests

Die Studie wurde in Form von Einzelinterviews durchgeführt. Die Teilnehmer wurden ermutigt, ihre persönlichen Perspektiven, Erfahrungen und Vorschläge einzubringen. Außerdem wurden sie gebeten, ihre Gedanken zum Entwurf laut zu äußern. Anschließend wurden ihnen nacheinander alle Grafiken des Entwurfs gezeigt (1. Abbildung 3, 2. Abbildung 4, 3. Abbildung 5).

Offene Fragen zu Designelementen, Verständlichkeit von Informationen und allgemeinen Nutzererfahrungen sollten Aufschluss über mögliche Stärken und Schwächen des Design-Prototyps geben. Ziel war es, frühzeitig wertvolles Feedback zu erhalten, um Optimierungen und Anpassungen vornehmen zu können, bevor die Umsetzung in die nächste Phase geht. Die lauten Gedanken der Teilnehmerinnen und Teilnehmer wurden von Herrn Huber in einem Freitextfeld dokumentiert.

5.1.3 Ergebnisse der Mockup-Studie

Die Rückmeldungen der Studienteilnehmenden lassen sich in folgende konsistente Muster zusammenfassen.



5.1.3.1 Bild 1: Visualisierung der Studiengänge

Die Übersichtskarte wurde von den Teilnehmenden als unübersichtlich und geclustert wahrgenommen. Besonders die Farbähnlichkeit der Studiengänge wurde als störend empfunden.

Eine konkrete Verbesserungsmöglichkeit besteht darin, die Überschriften der Nebel zu vergrößern und fett darzustellen, um die Lesbarkeit zu verbessern. Darüber hinaus wurde vorgeschlagen, alle Abkürzungen der Studiengänge standardmäßig auszublenden und nur den Nebel zu beschriften, um die Übersichtlichkeit zu erhöhen. Sobald der Benutzer mit der Maus in die Nähe des Nebels kommt, sollte die Überschrift des Nebels ausgeblendet und die Abkürzungen aller darin enthaltenen Studiengänge eingeblendet werden. Um die Zufälligkeit und Ähnlichkeit der Farben zu vermeiden, könnten die Studiengänge in den Fakultätsfarben und der Nebel im Hintergrund neutral (z.B. grau) eingefärbt werden.

5.1.3.2 Bild 2: Interaktivität

Das Hover-Event über eine Bubble mit einem Tooltip wurde durchweg als passend und verständlich betrachtet. Dieses Element wurde von den Teilnehmenden als intuitiv und funktional empfunden.

5.1.3.3 Bild 3: Popup mit Details eines Studiengangs

Über die Anordnung der Informationen im Popup gingen die Meinungen auseinander. Während einige eine absteigende Sortierung nach Relevanz bevorzugten, sprachen sich andere für

eine alphabetische Sortierung aus. Positiv bewertet wurde die Funktion, ähnliche Studiengänge nach Ähnlichkeit zu sortieren. Die Darstellung der Branchen wurde als weniger aussagekräftig empfunden und teilweise vorgeschlagen, diese Information zu entfernen. Die Möglichkeit, sich Unternehmen in Regensburg anzeigen zu lassen, wurde positiv bewertet und es wurde ange regt, diese Funktion ausklappbar zu gestalten, um die Übersichtlichkeit zu erhöhen. Es wurde außerdem empfohlen, die Informationen zu den Einstiegsgehältern zu erläutern, möglicherweise durch die Integration von Fragezeichen symbolen und Tooltips.

5.1.3.4 Generelle Anmerkungen

Zu den allgemeinen Verbesserungsvorschlägen gehören ein besserer Farbkontrast, mehr Erläuterungen zu verschiedenen Informationen und eine einheitliche und intuitive Anordnung der Designelemente, um die Übersichtlichkeit insgesamt zu erhöhen.

5.2 Prototypen-Studie: Evaluation durch Studieninteressierte

Nach der Entwicklung des Prototyps auf Basis des Mockups und der Erkenntnisse aus der Mockup-Studie wurden Studieninteressierte eingeladen, den Prototyp von StudyMap zu testen. Nach der selbstständigen Interaktion mit dem Prototyp wurden die Teilnehmenden gebeten, an einer Umfrage teilzunehmen, um ihre Erfahrungen, Meinungen und Verbesserungsvorschläge zu dokumentieren.

Der folgende Abschnitt beleuchtet die Methodik, den genauen Ablauf des Tests und stellt die Ergebnisse dieser Evaluationsstudie vor, die wertvolle Einblicke in die Effektivität und Benutzerfreundlichkeit des Prototyps liefert.

5.2.1 Methodik und Teilnehmende

Der nächste Schritt im UCD-Prozess ist das Testen des Prototyps mit potenziellen Nutzern, d.h. Studieninteressierten. Zu diesem Zweck wurde  auf der Grundlage des Mockups und der im vorherigen Kapitel beschriebenen Mockup-Studie ein Prototyp entwickelt (siehe Abbildung 6). Ziel dieser Studie ist es, frühe Designentscheidungen zu hinterfragen und das Tool auf Usability zu testen. Der Vorteil einer solch frühen Evaluierung ist, dass größere Änderungen am Endprodukt wesentlich aufwendiger sind als in einem frühen Entwicklungsstadium.

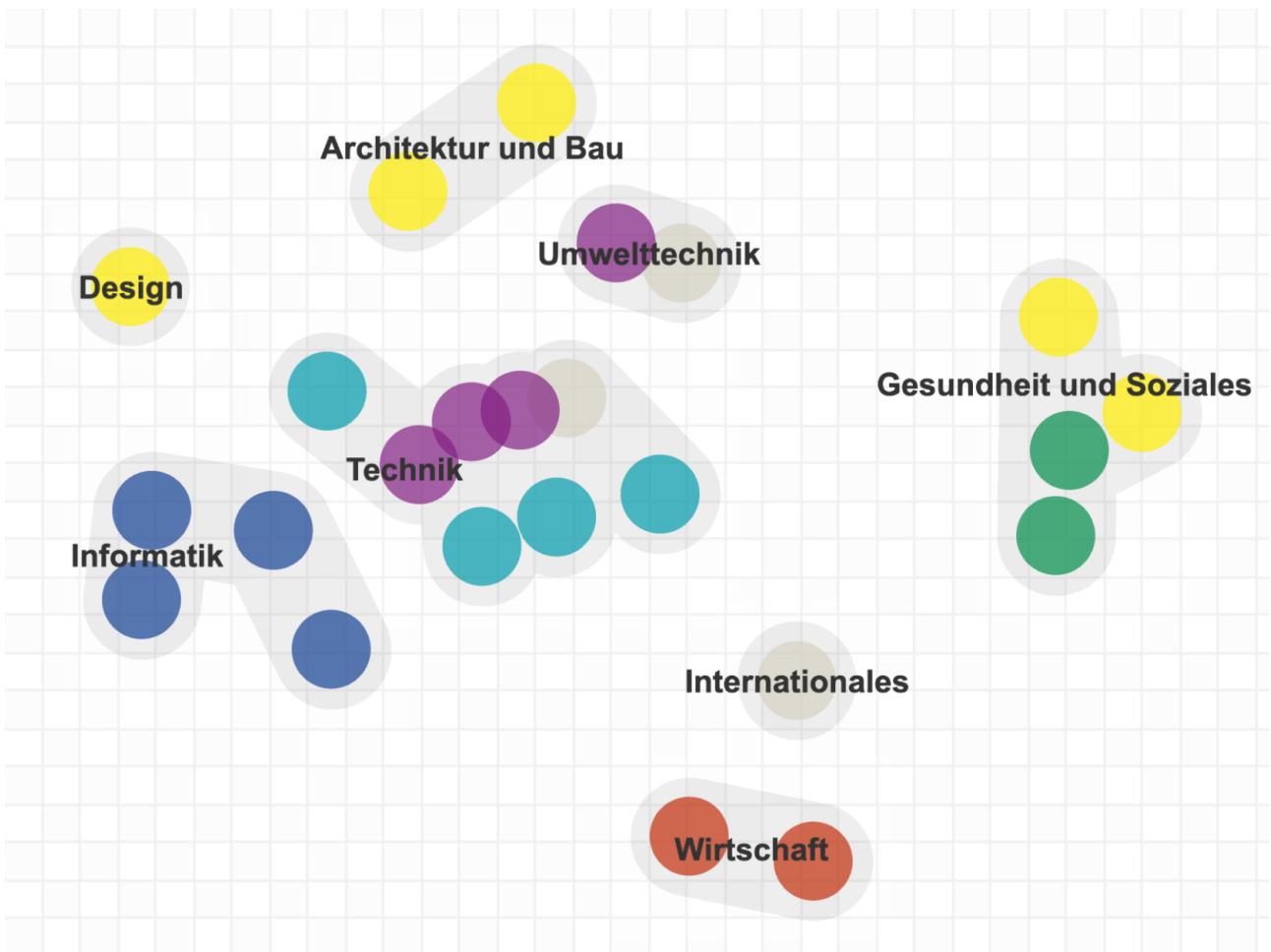


Abbildung 6: StudyMap Prototyp
Quelle: Eigene Darstellung

An der Studie nahmen 40 Schülerinnen und Schüler der 11. Klasse des Johannes-Nepomuk-Gymnasiums in Rohr i.NB teil. Alle diese Schüler werden voraussichtlich im Jahr 2026 die allgemeine Hochschulreife erlangen und entsprechen damit genau der Zielgruppe eines Studien-gangfinders. Durch die örtliche Nähe des Gymnasiums zur OTH-Regensburg ist die Eignung des Teilnehmerkreises zusätzlich gegeben.

Die Befragung fand in zwei Gruppen (Klasse 11A und Klasse 11B) im Johannes-Neopomuk-Gymnasium vor Ort im Computerraum der Schule statt. Dort konnten die Schülerinnen und Schüler an den Windows-Rechnern der Schule sowohl den Prototyp testen als auch die Umfrage in Einzelarbeit beantworten.

5.2.2 Ablauf des Tests

Die Studie begann mit einer kurzen Einführung, in der den Teilnehmenden der Zweck der Studie und die Handhabung des Prototyps erläutert wurden. Anschließend wurden die Teilnehmenden gebeten, den Prototyp selbst zu verwenden und zwei Aufgaben innerhalb einer Online-Umfrage zu lösen, um die Benutzerfreundlichkeit, die Navigationsstruktur und die Funktionalitäten des Studiengangsfinders zu testen (siehe Abbildung 7 - **Umfrage Teil 2**). In dieser Phase wurden sowohl quantitative als auch qualitative Daten erhoben, um ein umfassendes Bild der Benutzererfahrung zu erhalten.

The figure shows a vertical stack of four rectangular survey questions, each with a thin gray border:

- Aufgabe 1: Welcher Studiengang ist ähnlich zu Architektur? ***
Meine Antwort
[Text input field]
- Aufgabe 2: Wie unterscheiden sich Informatik und medizinische Informatik? ***
Meine Antwort
[Text input field]
- Wie schwer fandest du Aufgabe 1? ***
1 2 3 4 5
Sehr schwer Sehr leicht
- Wie schwer fandest du Aufgabe 2?**
1 2 3 4 5
Sehr schwer Sehr leicht

Abbildung 7: Prototypen-Studie: Aufgaben (Umfrage Teil 2)
Quelle: Eigene Darstellung

Nach der Interaktion mit dem Prototyp wurden die Teilnehmenden gebeten, zusätzlich zu den Aufgaben weitere Fragen in der anonymen Online-Umfrage zu beantworten.

Die ersten Fragen der Umfrage waren Multiple-Choice-Fragen mit den Antwortmöglichkeiten „Ja“, „Nein“ und „Keine Angabe“ (**Umfrage Teil 1**).

1. Willst du nach dem Abi studieren?
2. Wenn ja, weißt du schon, was für ein Studium?
3. Würde dir StudyMap bei der Wahl des Studiengangs helfen?

4. Die Antwort hier ist „Ja“ (Aufmerksamkeitsfrage)

Danach folgten die beiden oben beschriebenen Aufgaben (siehe Abbildung 7). Zuletzt wurden die folgenden Fragen mit Freitextfeldern zur Beantwortung gestellt (**Umfrage Teil 3**):

1. Was findest du an dem Konzept gut?
2. Was findest du am Konzept schlecht?
3. Hast du neue Ideen für den Prototyp?
4. Wenn du bei „Würde StudyMap dir bei der Studienwahl helfen?“ NEIN angekreuzt hast:
Warum nicht?

Die Fragen wurden bewusst mit Freitextfeldern zur Beantwortung versehen, um den Teilnehmenden die Möglichkeit zu geben, mit allen möglichen Gedankengängen zu antworten. Dies kann dazu führen, dass Ideen entstehen, die bisher noch nicht bedacht wurden und somit im Endprodukt umgesetzt werden können. Die Befragung dauerte ca. 15 Minuten. Durch die Kombination der Interaktion mit dem Prototyp und der schriftlichen Befragung konnten vielfältige Einblicke in die Nutzerperspektive gewonnen werden.

5.2.3 Ergebnisse der Prototypen-Studie

5.2.3.1 Umfrage Teil 1: Fragen



Abbildung 8: Ergebnisse: Teil 1
Quelle: Google Forms + selbst ergänzte absolute Werte

Abbildung 8 zeigt, dass die erste Frage („Möchtest du nach dem Abitur studieren?“) von 92,5 % der Befragten bejaht wurde. Während zwei Personen (5 %) die Frage verneinten, machte nur eine Person (2,5 %) keine Angabe. Daraus lässt sich schließen, dass es sich bei der gewählten Teilnehmergruppe tatsächlich um die richtige Zielgruppe von StudyMap handelt.

Auf die Frage „Wenn ja, weißt du schon, was du studieren willst?“ antworteten 30 % (12) der Befragten mit „Ja“. 67,5 % (27) haben mit „Nein“ geantwortet und 2,5 % haben wiederum die Angabe verweigert. Die Person, die bei Frage 1 angegeben hat, nicht zu antworten, hat bei dieser Frage mit „Nein“ geantwortet. Die Person, die bei Frage 2 mit „Keine Angabe“ stimmte, hatte zuvor bei Frage 1 mit „Ja“ gestimmt. Die beiden Schüler, die bei Frage 1 mit „Nein“ für ein Studium geantwortet haben, haben auch Frage 2 mit „Nein“ votiert.

Zieht man die beiden Schülerinnen und Schüler, die sich bei Frage 1 gegen ein Studium entschieden haben, von den „Nein“-Antworten bei Frage 2 ab, so verbleiben 25 von 37 Studieninteressierten, die noch nicht wissen, was sie studieren werden. Daraus ergibt sich ein Anteil von 67,57 % der Schüler der 11. Klasse, welche voraussichtlich nach dem Abitur eine Studienorientierung benötigen. Dieser Anteil an Schülern könnte mithilfe des innovativen Studiengangsfilters zu einem passenden Studiengang geführt werden.

Die dritte Frage („Würde dir StudyMap bei der Studienwahl helfen?“) wurde von 37 von 40 Teilnehmern (92,5 %) bejaht. Zwei Teilnehmer antworteten mit „Nein“ und eine Person enthielt sich der Stimme. Im Falle einer Verneinung hatten die Befragten die Möglichkeit, dies im Anschluss in einem Freitextfeld zu begründen. Die beiden Personen, die mit „Nein“ geantwortet haben, gaben an, dass StudyMap ihnen nicht bei der Studienwahl helfen würde, da sie bereits wüssten, was sie studieren wollen. Die Antworten auf diese Frage verdeutlichen noch einmal den Bedarf des Studiengangsfinders als Orientierungshilfe für Studieninteressierte.

Die Frage Nr. 4 der Umfrage Teil 1 wurde von allen Teilnehmern korrekt mit „Ja“ beantwortet. Sie diente dazu, willkürlich ausgefüllte und damit nicht auswertbare Teilnahmen aus der Befragung auszusortieren.

5.2.3.2 Umfrage Teil 2: Aufgaben

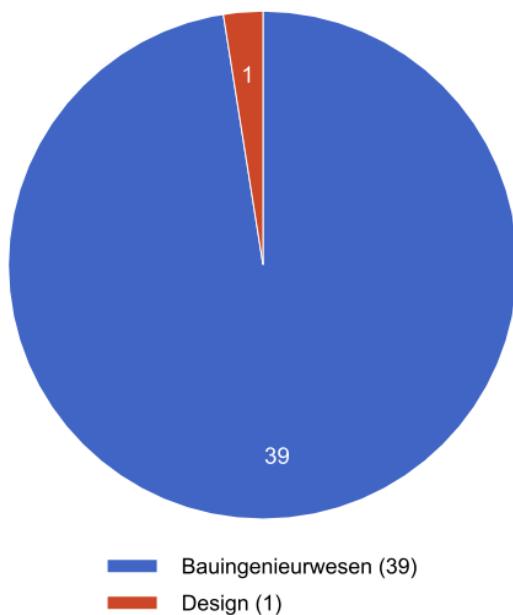


Abbildung 9: Welcher Studiengang ist ähnlich zu Architektur? (Aufgabe 1)
Quelle: Eigene Darstellung

In Aufgabe 1 sollten die Studienteilnehmer herausfinden, welcher Studiengang dem Studiengang „Architektur“ ähnlich ist. Als Hilfsmittel stand ihnen der Live-Prototyp zur Verfügung. Wie in den vorherigen Kapiteln erläutert, werden ähnliche Studiengänge durch den MDS-Algorithmus nahe beieinander platziert. Die Lösung war daher der Studiengang „Bauingenieurwesen“ (siehe Abbildung 10). 39 von 40 Studierenden (97,5 %) lösten diese Aufgabe richtig. Die einzige falsche abgegebene Antwort entsprach dem Studiengang „Design“. „Design“ ist im Prototyp

eine Inhaltskategorie und kein Studiengang, ist aber tatsächlich dem Studiengang „Architektur“ benachbart. Die hohe Übereinstimmung mit der Lösung weist auf die Intuitivität des Konzepts hin und stärkt damit die Basis für die weitere Entwicklung.

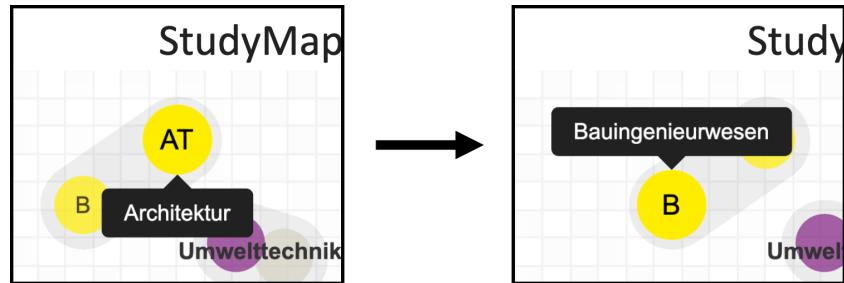


Abbildung 10: Prototyp: Architektur -> Bauingenieurwesen (Aufgabe 1)
Quelle: Eigene Darstellung

Das Screenshot zeigt einen Vergleichsprototypen. Oben links steht 'X SCHLIESSEN' und oben rechts 'MEHR ERFAHREN >'. Unter einem Suchfeld mit 'Medizinische Informatik' steht 'Studieninhalte vergleichen'. Unten rechts befindet sich ein 'X'. Im Zentrum ist der Titel 'Informatik BACHELOR OF SCIENCE (B.S.C.)' zu sehen, neben ihm zwei Symbole: ein Buch und eine Kappe. Links unter 'Inhalte' sind folgende Kategorien aufgelistet: Architektur, Bau, Design, Gesundheit, Informatik, Internationales, Kommunikation, Management, Marketing, Mathematik, Medien, Soziales, Technik, Umwelttechnik, Wirtschaft. Diese sind durch horizontale Balken dargestellt, wobei die Länge der Balken variiert. Rechts unter 'Beschreibung' steht: 'Sie möchten mit Informatik die Zukunft gestalten? Informatik ist in allen Branchen zu finden und Motor für zahlreiche Innovationen. Im Bachelorstudiengang lernen Sie alle wichtigen Grundlagen und die neuesten Entwicklungen kennen. Ein Studium nah an der Praxis macht Sie fit für den Berufseinstieg.' Unter 'Beschreibung' steht 'Firmen in Regensburg' mit einer Liste von Firmen: Vector, Siemens, BMW AG, Maschinenfabrik Reinhausen, Continental AG. Am unteren rechten Rand befindet sich ein grüner Chat-Symbol.

Abbildung 11: Prototyp: Vergleichsfeature
Quelle: Eigene Darstellung

In der zweiten Aufgabe sollten die Studieninteressierten herausfinden, inwiefern sich die allgemeine Informatik vom Studiengang Medizinische Informatik unterscheidet. Testweise wur-

de ein Feature entwickelt und zum Prototyp hinzugefügt, das es ermöglicht, Studiengänge zu vergleichen, ohne das Popup mit den Studiengangdetails schließen und erneut öffnen zu müssen.

Abbildung 11 zeigt den Vergleich der Studiengänge Informatik und Medizinische Informatik. Die farbigen Inhaltsbalken zeigen immer die Inhalte des ursprünglich ausgewählten Studiengangs an. Wenn ein Studiengang zum Vergleich hinzugefügt wird, werden dessen Inhalte mit grauen Balken über die anderen Inhalten gelegt. Es ist wichtig, dass die Inhalte klar und verständlich dargestellt werden, ohne dabei subjektive Bewertungen einzubeziehen. Der Vergleich kann über das X-Symbol im grauen Balken gelöscht und eingeklappt werden.

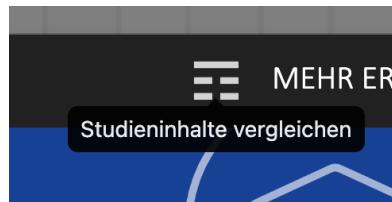


Abbildung 12: Prototyp: Vergleichsfeature öffnen
Quelle: Eigene Darstellung

Die Schülerinnen und Schüler können also mit dieser Funktion die Aufgabe bearbeiten. Wie in Abbildung 12 gezeigt, kann das Feature über den mit einem Tooltip beschrifteten Knopf im Header des Popups ausgeklappt werden. Um zur Lösung zu gelangen, gibt es zwei Wege:

1. Informatik-Bubble anklicken und mit Medizinische Informatik vergleichen
2. „Medizinische Informatik“-Bubble anklicken und mit Informatik vergleichen

Beides ergibt dasselbe Inhaltsdiagramm, aus dem sich die Unterschiede in den Studiengängen ablesen lassen.

Die Auswertung der Antworten auf Aufgabe 2 zeigt, dass die Mehrheit der Probanden das System und das Konzept hinter StudyMap verstanden hat. In Abbildung 13 sind einige der genannten Schlagwörter in Form einer *Wordcloud* dargestellt.

Eine Wordcloud ist eine grafische Darstellung von Textdaten. Dabei werden häufig vorkommende Wörter größer und prominenter dargestellt als seltener vorkommende Wörter. Sie bietet eine schnelle visuelle Zusammenfassung der wichtigsten Begriffe in einem Text oder einer Gruppe von Texten. Außerdem dient sie dazu, einen Überblick über die Antworten zu geben, ohne alle ausformulierten Antworten zu enthalten.

Die Begriffe *Gesundheit*, *Informatik*, *Mathematik*, *Internationales* und *Medien* sind sehr prominent (siehe Abbildung 13). Bei diesen Begriffen handelt es sich um inhaltliche Kategorien, die im Prototyp verwendet werden und sich in den beiden Studiengängen unterscheiden. Daher kann die anfänglich genannte These, dass das Konzept verstanden wurde, bestätigt werden. Die Antworten der Teilnehmenden sind im Anhang ausformuliert vorzufinden.

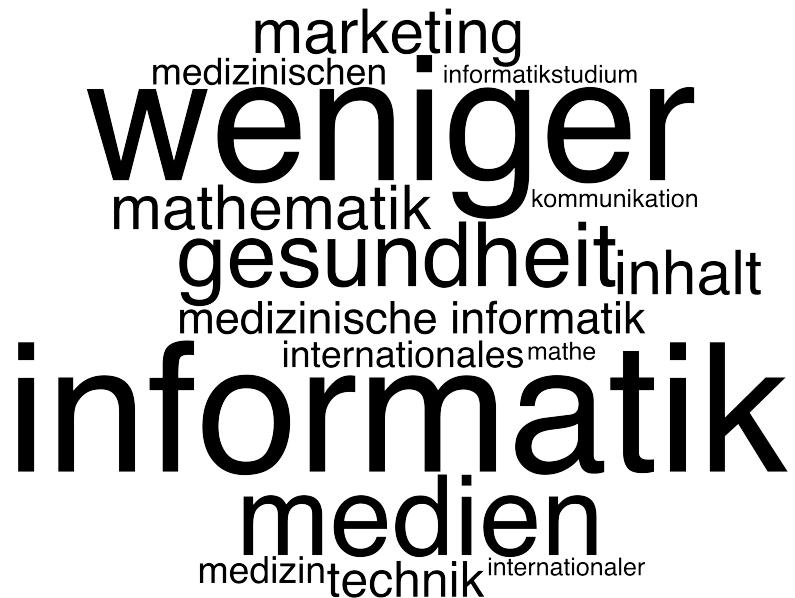


Abbildung 13: Wordcloud der Antworten (Aufgabe 2)
Quelle: Eigene Darstellung

Anschließend an die beiden Aufgaben wurden die Teilnehmer nach ihrer Schwierigkeitsbeurteilung befragt. Die Schüler und Schülerinnen bewerteten jeweils die Aufgaben von 1 bis 5 (1: sehr schwer, 5: sehr leicht). Anhand der folgenden Abbildung 14 ist erkennbar, dass die meisten Teilnehmer Aufgabe 1 als einfach empfanden.

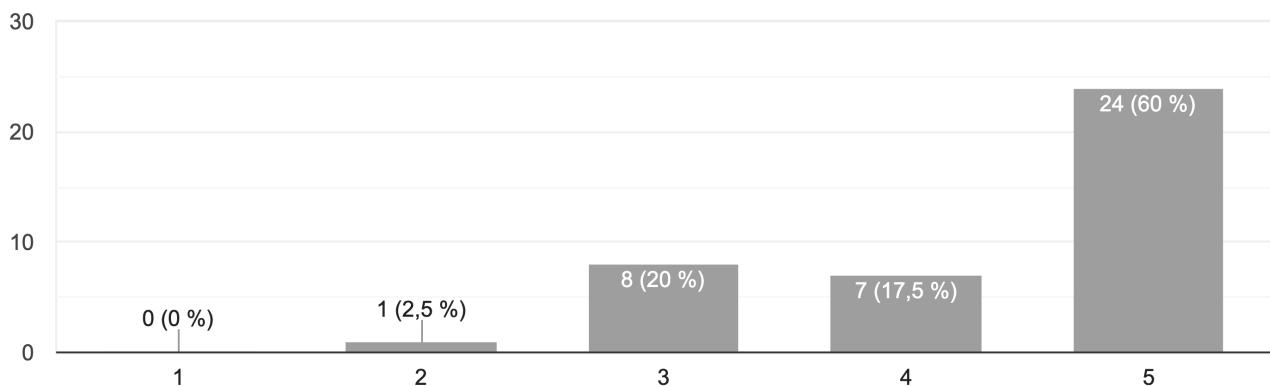


Abbildung 14: Wie schwer fandest du Aufgabe 1? (Auswertung Aufgabe 1)
Quelle: Google Forms

77,5 % (31/40) der Befragten gaben an, dass die Aufgabe leicht zu bewältigen war (Schwierigkeitsstufe 4 oder 5). Aufgabe 2 wurde erneut als deutlich schwieriger bewertet. Obwohl 70 % angaben, dass die Aufgabe leicht zu bewältigen war, lag das Gewicht deutlich mehr auf Schwierigkeit 4 als auf Schwierigkeit 5 (siehe Abbildung 15). Ferner ist der Anstieg bei der Schwierigkeitsstufe 2 (schwer) mit 15 % signifikant. Bei dieser Einstufung hatte die vorherige Aufgabe im Vergleich nur einen Anteil von 2,5 % mit einer Stimme. Im Verlauf wird diese Erkenntnis durch eine Tabelle 4 mit den stochastischen Werten belegt.

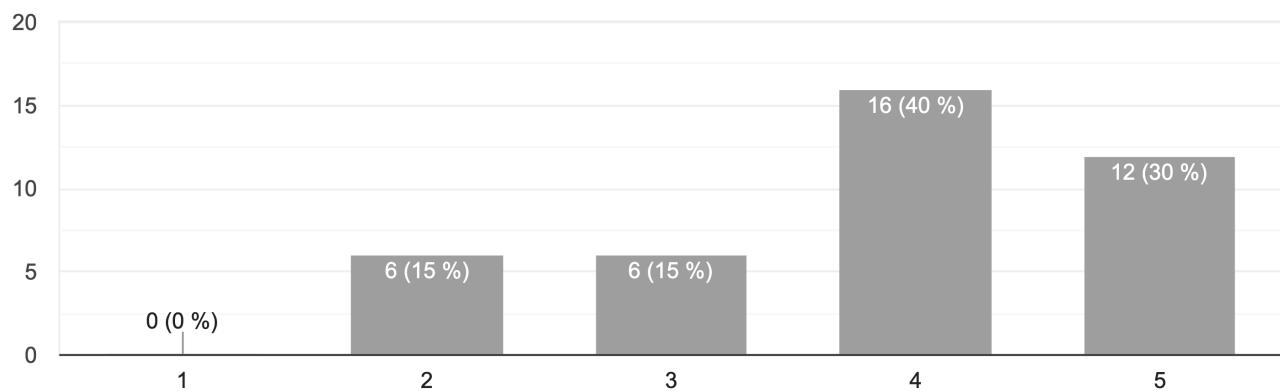


Abbildung 15: Wie schwer fandest du Aufgabe 2? (Auswertung Aufgabe 2)
Quelle: Google Forms

Zusammenfassend zeigt die stochastische Analyse in Tabelle 4 die bereits aus den Diagrammen ersichtlichen Tendenzen. Sowohl der Durchschnitt als auch der Median sind bei Aufgabe 2 niedriger, was darauf hindeutet, dass diese als schwieriger empfunden wurde. Minimum und Maximum sind in beiden Aufgaben gleich. Es bedeutet, dass es bei beiden Aufgaben Befragte gab, die die Aufgaben als sehr leicht und als eher schwer empfanden. Keiner fand die Aufgaben jedoch unlösbar. Die Standardabweichung ist bei Aufgabe 2 höher als bei Aufgabe 1. Dies könnte bedeuten, dass Aufgabe 1 so einfach war, dass es für viele selbstverständlich war, die richtige Lösung zu finden und deshalb der Großteil sehr einfach ankreuzen konnte. Bei Aufgabe 2 lag der Mittelwert eher in der Mitte der Skala, weshalb die Streuung höher ist und somit die Meinungen über die Schwierigkeit stärker differierten.

Fragen \ Ergebnisse	Durchschnitt	SD	Median	Min.	Max.
Wie schwer fandest du Aufgabe 1?	4.350	0.882	5.000	2	5
Wie schwer fandest du Aufgabe 2?	3.850	1.014	4.000	2	5

Tabelle 4: Auswertung der Fragen zu den Aufgaben

5.2.3.3 Umfrage Teil 3: Aufgaben

Der dritte Teil der Studie beinhaltet drei Fragen, die von den Schülerinnen und Schülern in Freitextfeldern beantwortet werden konnten. Aus diesem Grund werden auch diese Felder mithilfe von Wordclouds ausgewertet. So kann  schnell einen Überblick über die wichtigsten Punkte der Antworten erhalten.



Abbildung 16: Wordcloud zu „Was findest du an dem Konzept gut?“ (Umfrage Teil 3)
Quelle: Eigene Darstellung

Abbildung 16 zeigt die Wordcloud für die erste Frage des letzten Umfrageteils „Was findest du an dem Konzept gut?“. Wie bei der vorherigen Wortwolke sind häufig verwendete Begriffe prominenter (größer), während kleinere Begriffe nur selten verwendet wurden. Wie zuvor sind auch hier die vollständigen Antworten im Anhang zu  finden. Ignoriert man die Begriffe *Studiengang* und *Studiengänge*, sind die Begriffe *vergleichen* und *Vergleich* mit insgesamt 20 Vorkommnissen sehr präsent. Es lässt sich schlussfolgern, dass die Implementierung der Vergleichsfunktion in den Prototypen erfolgreich ist und weiterverfolgt und optimiert werden sollte.

Weitere Schlagwörter, die ins Auge fallen, sind *schnell*, *übersichtlich*, *Überblick*, *direkt*, *leicht*, *praktisch* und *ähnliche*. Diese Begriffe beziehen sich möglicherweise auf das allgemeine Konzept des Studiengangsfinders und unterstützen erneut die Basis des Konzepts, dass eine solche Art von Software bei der Studiengangsfundung für Studieninteressierte sehr nützlich sein kann.



Abbildung 17: Wordcloud zu „Was findest du an dem Konzept schlecht?“ (Umfrage Teil 3)
Quelle: Eigene Darstellung

Die zweite Frage lautete: „Was findest du am Konzept schlecht?“ Für diese Frage wurde ebenfalls eine Wordcloud erstellt (siehe Abbildung 17). Um alle Punkte vollständig zu erfassen, wurde jedoch hauptsächlich mit den Original-Antworten gearbeitet. Der größte Begriff in Bezug auf die Zufriedenheit war *nichts*, was dies verdeutlicht, dass der Großteil mit dem Prototyp bereits zufrieden war. Bei genauerer Betrachtung lassen sich jedoch einige Kritikpunkte feststellen:

- Übersichtlichkeit bei vielen Bubbles
- Platzierung von Bubbles wirkt willkürlich
- Beim Vergleich ist nicht klar erkennbar, welche Farbe dem ursprünglich ausgewählten Studiengang und welche dem Vergleichsstudiengang zugeordnet ist
- In der Funktion „Studiengang vergleichen“ sollten die Studiengänge entweder alphabetisch sortiert oder durch eine Suchfunktion filterbar sein.
- Legende für direkten Vergleich einfügen
- Die Vergleichsoption sollte auffälliger gestaltet werden

- Was passiert, wenn ein Studiengang inhaltlich in zwei übergeordnete Gruppen fällt?
- Erforderliche Soft Sk mit angeben
- Button „Mehr erfahren“ geht in der Übersicht unter
- Einstiegsgehalt in absoluten Zahlen

Zusammenfassend sollte die Platzierung der Bubbles erklärt werden. Diese erfolgt nicht willkürlich, sondern mithilfe des MDS-Algorithmus und der Inhalte des Studiengangs automatisch. Im finalen Produkt werden die Bubbles voraussichtlich durch einen beten Datensatz auch geschickter platziert, wodurch sich dieser Kritikpunkt vermutlich erledigen sollte.

Die Farbe des Vergleichsfeatures wurde mehrmals kritisiert. Im oment werden die Inhaltsbalken des zu vergleichenden Studiengangs mit einem neutralen Grau über die des original angeklickten Studiengangs gelegt. Eine mögliche Verbesserung wäre die Verwendung einer Farbänderung, eines zusätzlichen Hilfe-Dialogs oder einer Legende. Um eine schnellere Suche zu ermöglichen, könnten die zu vergleichenden Studiengänge vor der Auslieferung ans Frontend durch das Backend alphabetisch sortiert werden.

Ferner ist es kein Hindernis, wenn ein Studiengang inhaltlich in zwei Gruppen fällt, da immer der höchste Wert zur Gruppenzugehörigkeit gewählt wird. Wenn zwei Kategorien denselben Wert haben und gleichzeitig die höchsten Werte sind, wird die erste Kategorie verwendet. Außerdem ist es kein Problem, da die Wahrscheinlichkeit hoch ist, dass die Bubbles nahe beieinander liegen, wenn der Algorithmus berechnet wird. Selbst wenn sie sich in verschiedenen Supergruppen befinden, werden sie aufgrund ihrer Koordinaten als ähnliche Studiengänge betrachtet.

Es gestaltet sich eher schwierig, erforderliche Soft Skills für das Studium anzugeben, da es keine offizielle Dokumentation über die benötigten Soft Skills pro Studiengang gibt.

Der Button „Mehr erfahren“ geht in der Übersicht unter. Dieser Punkt wurde nur von einer Person geschrieben und die Schaltfläche ist reits sehr groß und deutlich sichtbar auf der Seite rechts oben platziert, weshalb dieser Punkt nicht weiter evaluiert wird.

Es wurde zweimal nach dem Einstiegsgehalt in konkreten Zahlen gefragt. Ein Nachteil dieser Implementierung besteht darin, dass die Werte regelmäßig überprüft und gewartet werden müssen, im Gegensatz zur aktuellen Implementierung mit den eichen Bewertungen „Überdurchschnittlich“, „Durchschnittlich“ usw. Deshalb wird dieser Punkt auch nicht weiter untersucht.



Abbildung 18: Wordcloud zu „Hast du neue Ideen für den Prototypen?“ (Umfrage Teil 3)

Quelle: Eigene Darstellung

Die in Abbildung 18 genannten Punkte ähneln sehr den Kritikpunkten, die in der vorherigen Frage „Was findest du an dem Konzept schlecht?“ behandelt wurden. Im Folgenden werden die Kritikpunkte auf Basis der Wordcloud und den vollen Antworten der Teilnehmenden gruppiert und zusammengefasst dargestellt. Die Original-Antworten sind, wie bei den vorherigen Fragen auch, im Anhang aufgeführt.

- Vergleichsfunktion optisch ansprechender gestalten
- Vergleichsfunktion über mehr als zwei Studiengänge
- Suchfunktion zur Vergleichsfunktion hinzufügen
- Persönlichkeitstest einführen (als Orientierung)
- Ähnliche Studiengänge anders platzieren/gestalten
- Absolute Werte bei Gehälter
- Mehr Informationen zu den einzelnen Fächern ggf. Bezug auf die Schule nehmen
- Farben ändern

Die optische Gestaltung der Vergleichsfunktion wurde mehrmals erwähnt. Die Farben und die Gestaltung des Buttons zur Aktivierung der Vergleichsfunktion sowie der Inhaltsbalken

sollten geändert werden. Außerdem wurde der Wunsch geäußert, dass man mehr als zwei Studiengänge gleichzeitig vergleichen kann. Es stellt sich jedoch die Frage, ob dies sinnvoll ist, da es auf mobilen Endgeräten schnell unübersichtlich werden könnte. Ein weiterer oft genannter Wunsch ist eine Suchfunktion innerhalb der Vergleichsfunktion. Dies könnte, wie bei der vorherigen Frage bereits beschrieben, entweder durch eine alphabetisch sortierte Liste oder durch ein Autocomplete-Textfeld gelöst werden. Ein Autocomplete-Textfeld schlägt Vorschläge vor, während Buchstaben eingegeben werden. Es ähnelt somit einem Suchfeld.



Außerdem wurde mehrfach ein Persönlichkeitstest gewünscht, der spezifische Fragen an den oder die Studieninteressierten stellt und daraus eine Empfehlung für das zu wählende Studium ableitet. Wie bereits in Unterabschnitt 2.2 erwähnt, bringt dies jedoch auch einige Limitierungen mit sich. Eine Möglichkeit für die Zukunft wäre, eine optionale Online-Umfrage anzubieten. Diese könnte die Ergebnisse aufgrund der eingegebenen Daten in StudyMap als eine Art Heatmap darstellen und die Zugehörigkeit zu einer bestimmten Position und somit zu einer Supergruppe bzw. Studiengang markieren. Eine Heatmap ist eine grafische Darstellung von Daten, bei der Werte in einer Matrix durch Farben codiert werden. Intensivere Farben repräsentieren höhere Werte, um Muster oder Trends visuell hervorzuheben. In diesem Fall werden die Zugehörigkeiten zur Kategorie „Wirtschaft“ dargestellt.



Es wurde erwähnt, dass die Sektion „Ähnliche Bachelor-Studiengänge“ nicht gut platziert ist. Dieser Eindruck könnte aufgrund der eher standardmäßigen Gestaltung (siehe Abbildung 5 ganz unten im Popup) entstehen. Eine Möglichkeit wäre, die Studiengänge mit Icons zu versehen oder sie wie bei dem Abschnitt „Firmen in Regensburg“ mit schwarzen, abgerundeten Rechtecken als Schaltflächen zu präsentieren.

Absolute Werte bei den Gehältern können aus den vorher genannten Gründen nicht umgesetzt werden.

Für die Kurzübersicht wäre es vermutlich zu umfangreich, Informationen zu den einzelnen Fächern bereitzustellen. Das Popup ist bereits sehr voll - mehr Inhalt könnte die Benutzererfahrung negativ beeinflussen. Ein zusätzlicher Reiter für alle Fächer wäre vermutlich zu viel. Um die einzelnen Fächer einzusehen, muss der Interessierte lediglich auf den Button „Mehr erfahren“ klicken und schließlich den Studienverlaufsplan des jeweiligen Studiengangs öffnen. StudyMap bietet einen Überblick und eine grobe Orientierungshilfe für das Studium. Für weitere Informationen steht Studieninteressierten die OTH-Website mit detaillierten Informationen zu allen Studiengängen zur Verfügung.

Wie bereits in Absatz 5.1.3.1 erläutert, werden die Bubbles in den Fakultätsfarben einge-

färbt. Dies ist eine Anforderung der Vizepräsidentin der OTH-Regensburg und der Fakultäten der Hochschule. Daher kann dem Wunsch nach anderen Farben nicht entsprochen werden. Andere Farben könnten zu Verwirrung führen, da nicht klar ist, warum eine bestimmte Bubble beispielsweise grün gefärbt ist. Die aktuelle Lösung definiert klar, welcher Studiengang welche Farbe erhält. So umfasst z.B. die Supergruppe Technik nicht nur die Studiengänge der Fakultät Maschinenbau, sondern auch die Studiengänge der Fakultät für Elektro- und Informationstechnik. Obwohl die Studiengänge in derselben Supergruppe sind, kann man sie durch die Farben leichter voneinander unterscheiden.

5.3 Zusammenfassung der Studien

Zusammenfassend wurde durch die Anwendung der Mockup-Studie ein positiv bewertetes Konzept für den Prototypen entwickelt. Durch den Prozess des User Centered Designs wurden die späteren Benutzer von Anfang an in den Entwicklungsprozess einbezogen. Dadurch wird sichergestellt, dass die Bedürfnisse der Zielgruppe im finalen Produkt möglichst umfassend erfüllt werden.

Da die Anforderungen der Zielgruppe von Anfang an klar definiert sind, kann durch den Einsatz von UCD die Entwicklungszeit deutlich verkürzt werden. Beide Studien erzielten ähnliche Ergebnisse. Der Hauptkritikpunkt an den Entwürfen bzw. dem Prototypen ist die mangelnde Übersichtlichkeit. Zusammenfassend lässt sich festhalten, dass alle Teile des Endprodukts durch Hilfedialoge erläutert werden sollten. Es muss auch geprüft werden, ob eine Legende für die Bubbles als Erklärung erforderlich ist. Der Fokus sollte also auf der Darstellung von möglichst vielen Informationen in möglichst übersichtlicher Form liegen, um den Nutzern einen Überblick und Vergleich aller Studiengänge zu ermöglichen.

Durch die erste Mockup-Studie konnten bereits einige Schwachstellen des Entwurfs behoben werden, die bei der Implementierung des Prototypen berücksichtigt wurden. Bereits in dieser Phase konnte Entwicklungszeit eingespart werden, da Änderungen mithilfe des Mockups vor der ersten Entwicklung geplant werden konnten. Das positive Feedback auf den Prototyp im Rahmen der Prototypenstudie von 37 Studieninteressierten war eine Bestätigung für den Bedarf an StudyMap als Instrument zur Studienorientierung.

Zusammenfassend ist festzuhalten, dass die Studien erfolgreich verlaufen sind. Als Ausblick in die Zukunft könnte ein Orientierungstest interessant sein. Dieser Test würde den Studierenden anhand einiger Fragen eine Position auf der StudyMap zuweisen, damit sie sich die Studiengänge in der Nähe dieser Position genauer ansehen können. Eine weitere Idee für die Zukunft ist, dass mehr als nur zwei Studiengänge inhaltlich miteinander verglichen werden können.

6 Implementierung und Deployment

6.1 Technologieauswahl und Implementierungsdetails

Die Auswahl der geeigneten Technologien für die Entwicklung des innovativen Studiengangsfilters wurde so gewählt,  eine effiziente und interaktive Lösung zu gewährleisten. Besonders wichtig ist die Nutzbarkeit auf Smartphones und Desktop-Geräten, wodurch sich außerdem neue Herausforderungen hinsichtlich der Bedienung ergeben. Im Folgenden werden die Hauptkomponenten des Technologiestacks und ihre jeweiligen Funktionen erläutert.

6.1.1 PixiJS - Interaktive Grafik

Für die Darstellung der interaktiven Grafik wurde PixiJS gewählt. PixiJS ist ein leistungsstarkes WebGL-Rendering-Framework, das eine schnelle und reibungslose Darstellung von Grafiken ermöglicht. (PixiJS, 2023) WebGL (Web Graphics Library) ist eine JavaScript-Schnittstelle zur Berechnung hochperformanter, interaktiver 2D- und 3D-Grafiken zur Anzeige im Browser. (Corporation, 2023c) Da nicht jeder Browser WebGL unterstützt, verwendet PixiJS als Ausweichlösung immer das klassische HTML5 Canvas ohne Hardwareunterstützung. (PixiJS, 2024b)

Die Entscheidung für PixiJS basiert auf seiner Effizienz bei der Verarbeitung komplexer 2D-Grafiken und seiner Fähigkeit, eine ansprechende Benutzererfahrung zu bieten. Neben PixiJS wurden verschiedene weitere Bibliotheken betrachtet:

- three.js: WebGL 3D-Framework
- D3.js: 2D-Datenvisualisierungsbibliothek
- Chart.js: HTML5-Bibliothek für Diagramme
- Paper.js: HTML5-Bibliothek für Animationen und interaktive Grafiken
- Fabric.js: 2D-Canvas Bibliothek

Der Studiengangsfinder erfordert eine 2D-Grafikdarstellung, weshalb *three.js*, das auf 3D-Visualisierungen spezialisiert ist, nicht als geeignete Grundlage gewählt wurde. (Three.js, 2023)

Die Entscheidung gegen die Verwendung von *D3.js* wurde aufgrund von Bedenken bezüglich der Dokumentation und der Wahrnehmung in der Entwicklergemeinschaft getroffen. Die



unvollständige Dokumentation und bestehende Forenthreads, die die Relevanz von D3.js in Frage stellen, könnten zu potenziellen Schwierigkeiten bei der Entwicklung und zukünftigen Wartungen führen. (Bostock & Observable, 2023)

Aufgrund der festgestellten Einschränkungen in der Flexibilität von *Chart.js* wurde gegen die Verwendung der Bibliothek entschieden. Obwohl *Chart.js* die Erstellung einer Vielzahl von Diagrammen ermöglicht, hat sich gezeigt, dass die Anpassbarkeit eingeschränkt ist. (etimberg et al., 2023a)



Basierend auf der wahrgenommenen Inaktivität des Projekts und den festgestellten Einschränkungen in Bezug auf Event-Handler wurde entschieden, *Paper.js* nicht zu verwenden. (Lehni & Puckey, 2023) Die begrenzten Event-Handler schränken die Interaktionsmöglichkeiten ein, was im Kontext des Studiengangsfilters, der eine umfassende Benutzerinteraktion für Smartphone und Desktop-Gerät erfordert, als unzureichend erachtet wurde. (etimberg et al., 2023b)

Obwohl *Fabric.js* als vielversprechende Alternative erschien, wurde PixiJS aufgrund mehrerer Faktoren bevorzugt (Zaytsev et al., 2023). Der professionellere Website-Auftritt von PixiJS trug dazu bei, das Vertrauen in die Zuverlässigkeit und Wartbarkeit des Frameworks zu stärken. Ein weiterer bedeutender Punkt ist die Anzahl der GitHub-Sterne in Relation zu der Anzahl an *offenen Issues*, die PixiJS aufweist. Eine höhere Anzahl an GitHub-Sternen mit gleichzeitig weniger offenen Issues, deutet oft auf eine größere und aktivere Entwicklergemeinschaft hin, was wiederum auf kontinuierliche Weiterentwicklung und Wartung schließen lässt. (Batista & Correia, 2023)

Mit Hilfe der in den folgenden Kapiteln vorgestellten Webschnittstelle werden Daten für die Visualisierung der Studiengänge vom PixiJS-Projekt abgerufen. Anschließend werden die Studiengänge auf einer Canvas-Fläche in Form von Bubbles platziert, die sich hinsichtlich ihrer Nähe durch die Ergebnisse des MDS-Algorithmus unterscheiden:

```
1 const x = ((renderedSize.width * 0.75) * data[i][4][0]) + renderedSize.width *  
  0.125;  
2 const y = ((renderedSize.height * 0.8) * (1 - data[i][4][1])) + renderedSize.height  
  * 0.10;
```

Die vorliegende Implementierung berechnet die x- und y-Positionen für die weißen Bubbles der Studiengänge. In diesem Fall enthält die Variable `renderedSize` die Breite des Canvas in Relation zur physischen Pixelhöhe. Die Variablen `data[i][4][0]` und `data[i][4][1]` enthalten die vom Algorithmus berechneten auf eins normierte Koordinaten. Durch Multiplikation

der Breite und Höhe der gezeichneten Fläche mit einem Pufferwert (z.B. 0.75) werden die Bubbles zentriert und nicht bis zum Rand gezeichnet. Ansonsten könnte es passieren, dass Bubbles genau am Rand stehen und dadurch aufgrund des zentralen Ankerpunkts des Kreises nur zur Hälfte sichtbar sind.



Abbildung 19: StudyMap: Tooltip über Bubble
Quelle: Eigene Darstellung

Die Implementierung beinhaltet außerdem die Funktionalität, dass ein Tooltip erscheint, wenn man mit der Maus über die Bubble fährt (siehe Abbildung 19). In der Entwicklung treten jedoch zwei große Herausforderungen auf: Wie funktioniert dieses Feature auf einem Smartphone ohne Maus? Was passiert, wenn das Tooltip außerhalb des Bildschirms gezeichnet wird?

Herausforderung 1: Bedienung

PixiJS verfügt über ein ereignisbasiertes System zur Verfolgung von Interaktionen mit der Zeichenfläche. Es gibt verschiedene Eventtypen, wie zum Beispiel `.click()`. Dieses Event wird ausgelöst, wenn auf das Objekt geklickt wird, auf dem der sogenannte Eventlistener aktiviert wurde. (PixiJS, 2024a)

Für das Tracking, ob der Benutzer mit der Maus über der Bubble ist, wird der Eventtyp `.onmouseover()` verwendet. Die Studiengänge werden, wie im Konzept vorgestellt, in Supergruppen eingeteilt. Alle Studiengänge, die zur gleichen Supergruppe gehören, werden durch ein Label mit dem Namen der Supergruppe und einen leichten Hintergrundnebel gekennzeichnet, um ihre Zugehörigkeit zu zeigen. Wenn der Benutzer keine Interaktion mit dem Canvas durchführt, wird kein Studiengang mit einem Label versehen, sondern lediglich als farbiger Kreis dargestellt (siehe Abbildung 20).



Abbildung 20: StudyMap: Supergruppe mit Label
Quelle: Eigene Darstellung

Beim Berechnen des Nebels der Supergruppen werden gleichzeitig die Minima und Maxima der X- Y-Koordinaten berechnet. Das Ergebnis ist ein unsichtbares Rechteck, das alle Studiengänge in dieser Inhaltsskategorie umfasst; dieses Rechteck hat die Funktion einer Hitarea. Auf dieser Hitarea wird ein `.onmouseover()`-Eventlistener platziert, der ausgelöst wird, sobald der Benutzer mit der Maus in die Nähe eines Studiengangs dieser Kategorie kommt.

Sobald dieser Fall eintritt, wird das Label der Supergruppe deaktiviert und alle Studiengangs-Bubbles erhalten als Label das Kürzel des jeweiligen Studiengangs (siehe Abbildung 21).

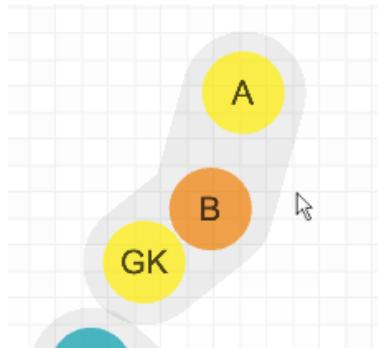


Abbildung 21: StudyMap: Supergruppe ohne Label - Hitarea
Quelle: Eigene Darstellung

Schließlich besitzt jede Bubble einen zusätzlichen `.onmouseover()`-Eventlistener. Wenn dieses Event ausgelöst wird, wird die Bubble vergrößert und immer gegenüber eventuell überschneidenden Bubbles in den Vordergrund gestellt. Die Labels der anderen Studiengänge in allen anderen Supergruppen werden deaktiviert, um den Fokus auf diese Inhaltsskategorie und explizit diese Bubble zu legen. Daraufhin erscheint darüber das Tooltip, das in Abbildung 19 zu sehen ist.

Da ein Smartphone in der Regel keine Maus angeschlossen hat, wird anstelle des für

Desktop-Benutzer verwendete `.onmouseover()`-Event das `.ontap()`-Event verwendet. Dieses Ereignis wird ausgelöst, wenn der Benutzer auf die Bubble des Studiengangs tippt. Die Hitarea des Supergruppen-Nebels wird dabei ignoriert.

Auch hier muss erneut differenziert werden. Wenn ein Desktop-Nutzer auf eine Bubble klickt, öffnet sich er Dialog mit den Studiengangsdetails gemäß dem Konzept. Wenn ein Smartphone-Nutzer auf die Bubble tippt, sollte sich das Popup nicht direkt öffnen. Es ist wichtig, dass der Nutzer zuvor das Tooltip gesehen hat, um zu wissen, welchen Studiengang er öffnet.

Aus diesem Grund wird beim `.ontap()`-Event auf eine Bubble zwischen drei Zuständen unterschieden:

1. **Bisher noch  nichts anderes angeklickt:** Da  wird der gleiche Hover-Effekt ausgeführt, den auch ein Desktop-Endgerät beim `.onmouseover()`-Event über eine Bubble auslösen würde.
2. **Es wurde bereits eine andere Bubble angeklickt:** In diesem Fall wird der Hover-Effekt für die bereits angeklickte Bubble deaktiviert und für diese Bubble aktiviert.
3. **Diese Bubble wurde bereits angeklickt:** Daraufhin wird das Tooltip entfernt und das Popup mit den Details zum Studiengang geöffnet. Dies ist das Äquivalent zu einem Doppelklick auf die Bubble.

Zur Implementierung dieser Funktionalität ist die Zwischenspeicherung eines Active State  erforderlich, der immer die zuletzt angeklickte Bubble enthält.

Herausforderung 2: Was passiert, wenn das Tooltip außerhalb des Bildschirms gezeichnet wird?

Die Zeichenoberfläche von PixiJS enthält keine vorgefertigten Anzeigeelemente oder Komponenten zur Erstellung einer Benutzeroberfläche. Das bedeutet, dass Elemente wie z.B. das Tooltip in Abbildung X selbst mit Hilfe von Polygonen gezeichnet werden müssen.

Das gezeigte Tooltip besteht aus einem abgerundeten Rechteck, das den Text darüber enthält, und einem Dreieck, das anzeigt, für welche Bubble das Tooltip geöffnet wurde. Das Rechteck und das dazugehörige Dreieck werden in Schwarz aneinander gesetzt, so dass das kleine Fenster wie eine Sprechblase aussieht.

Bei der Entwicklung der Klasse Tooltip wurde festgelegt, dass die Standardrichtung zum Öffnen des Tooltips nach oben zeigt. Der folgende Codeausschnitt zeigt, wie die Richtung festgelegt wird:

```
1 this.direction = 'top';
2 if (this.rootObjectYPosition - this.distance - this.container.height < 0) {
3     // if object is too high to display the tooltip, it will be drawn to the bottom
4     this.direction = 'bottom';
5 } else if (this.rootObjectXPosition - this.container.width / 2 < 0) {
6     // if object overlaps left world border, it will be drawn to the right
7     this.direction = 'right';
8 } else if (this.rootObjectXPosition + this.container.width / 2 > this.worldWidth) {
9     // if object overlaps right world border, it will be drawn to the left
10    this.direction = 'left';
11 }
```

Die Variablen `this.rootObjectXPosition` und `this.rootObjectYPosition` enthalten die x- und y-Koordinaten des Elements, an dem der Tooltip erscheinen soll. In diesem Fall sind dies die Koordinaten der jeweiligen Bubble. Die Variable `this.distance` enthält den vordefinierten Abstand, um den der Tooltip vom Element entfernt sein soll. Im vorliegendem Fall handelt es sich mindestens um den Radius des Studiengangskreises. Die Größe des Tooltips wird wiederum in den Variablen `this.container.height` und `this.container.width` gespeichert.

Die Richtung des Tooltip-Fensters wird nach dem Ausschlussverfahren festgelegt. Wenn das Tooltip mit der Standardrichtung 'top' (nach oben) aus der Zeichenfläche hinausragt, wird die Richtung nach unten ('bottom') festgelegt. Anschließend wird überprüft, ob das Element die linke Fenstergrenze überschreitet. Falls dies der Fall ist, wird die Richtung auf 'rechts' gesetzt. In diesem Fall wird die Richtung auf 'links' gesetzt. Analog dazu wird geprüft, ob das Objekt die rechte Grenze überschreitet.

Sobald die Ausrichtung des Tooltips festgelegt wurde, müssen die Elemente einzeln verschoben und gedreht werden. Führt die vorangegangene Bedingung dazu, dass die Richtungsvariable `direction` auf 'left' gesetzt wird, muss das Tooltip wie in der folgenden Abbildung 22 dargestellt links neben der Bubble erscheinen.



Abbildung 22: StudyMap: Tooltip links neben Bubble
Quelle: Eigene Darstellung

Der unten dargestellte Codeauszug zeigt die Platzierung des Tooltip-Containers auf der linken Seite. Zunächst wird die Rotation des Dreiecks geändert, sodass es sich um 90 Grad nach links dreht. Anschließend werden die x- und y-Koordinaten des Dreiecks so angepasst, dass es an der rechten Seite des abgerundeten Rechtecks anliegt. Schließlich wird der gesamte Container mithilfe der Zeilen sechs und sieben linksbündig und gleichzeitig vertikal zentriert neben der Bubble platziert. Sowohl das Dreieck als auch das abgerundete Rechteck und der Text sind im Container enthalten.

Platzierung des Tooltips auf der linken Seite:

```

1 if (this.direction == 'left') {
2     triangle.rotation = -Math.PI / 2;
3     triangle.y = this.container.height / 2 - triangleHeight / 2;
4     triangle.x = this.container.width;
5
6     this.container.x = -this.container.width - this.distance;
7     this.container.y = -background.height / 2;
8 }
```

Dieser Ansatz funktioniert jedoch nur bis zu einem gewissen Grad. Wenn das Tooltip breiter ist als die x-Achse des Endgeräts, wird es immer über die Seiten des Endgeräts hinausragen, da auf keiner der beiden Seiten genügend Platz für das Tooltip zur Verfügung steht.

6.1.2 Python - Berechnung der Positionen

Die Berechnung der Positionen für die Studiengänge basiert, wie im Abschnitt 3.1.3 erläutert, auf dem Multidimensionalen Skalierungsalgorithmus (MDS), der in Python implementiert ist. Der Algorithmus verarbeitet eine strukturierte CSV-Datei, die alle relevanten Informationen zu den Studiengängen, Feldern und Meta-Informationen (wie z.B. Fakultätszugehörigkeit) enthält. Dieser Datensatz bildet die Grundlage für die Positionsbestimmung im zweidimensionalen Raum.

Hierzu wird ein Python-Skript entwickelt, welches die CSV-Datei einliest und schließlich den MDS-Algorithmus auf den Daten anwendet. Im Verlauf der Arbeit wurde eine anfängliche Eigenimplementierung des MDS-Algorithmus in Betracht gezogen.

Zu diesem Zweck wurde eine Python-Klasse erstellt, die drei Methoden enthält:

1. calculate_distance_matrix(X)
2. read_data_from_csv()
3. mds(self)

Die Methode `calculate_distance_matrix` berechnet die Distanzmatrix, welche für den Ablauf des Programms essenziell ist. Der Code dazu wird an dieser Stelle ausgespart, da er bereits in Abschnitt 3.1.3 vollständig spezifiziert wurde. Außerdem ausgespart ist die Methode `read_data_from_csv()`, welche die Eingabedatei einliest, analysiert und schließlich die zur Berechnung benötigten Zahlen-Arrays zurückgibt. Für die Berechnung der mathematischen Gleichungen wird die Programmzbibliothek NumPy verwendet. (N. Team, 2023)

Folgender Ausschnitt eines Quellcodes beinhaltet die für diese Arbeit entwickelte Eigenimplementierung des MDS-Algorithmus, basierend auf dem im Abschnitt 3.1.3 genau erläuterten Ablauf:

```

1 def mds(self):
2     # Reads data from CSV file and transforms it to number rows
3     X = self.read_data_from_csv()
4
5     # Calculates distance matrix
6     M = self.calculate_distance_matrix(X)
7
8     # Double centered matrix
9     n = 11 # Number of subjects
10    I = np.identity(n) # Identity matrix
11    Jn = np.ones((n, n)) # Matrix Jn of all ones
12    C = I - (1/n) * Jn # Calculate centering matrix C
13    B = -0.5 * np.dot(np.dot(C, np.array(M)), C) # Calculate the double centered
14        matrix B
15
16    # Eigenvalues, Eigenvectors
17    m = 2 # number of dimensions (2D)
18    eigenvalues, eigenvectors = np.linalg.eig(B) # Perform eigenvalue decomposition
19
20    # Sort eigenvalues and corresponding eigenvectors in descending order
21    sorted_indices = np.argsort(eigenvalues)[::-1]
22    eigenvalues = eigenvalues[sorted_indices]
23    eigenvectors = eigenvectors[:, sorted_indices]
24
25    # Select the top m eigenvalues and corresponding eigenvectors
26    top_m_eigenvalues = eigenvalues[:m]
27    top_m_eigenvectors = eigenvectors[:, :m]
28
29    # Calculate the square root of the diagonal matrix
30    sqrt_Lambda_m = np.sqrt(np.diag(top_m_eigenvalues))
31
32    # Compute the matrix X using the formula
33    X_transformed = np.dot(top_m_eigenvectors, sqrt_Lambda_m)
34
35    return X_transformed

```

Die Entwicklung und Umsetzung dieser eigenen Lösung erwies sich jedoch als äußerst anspruchsvoll, da sie zahlreiche subtile mathematische Nuancen berücksichtigen müsste. Angesichts dieser Komplexität wurde die Entscheidung getroffen, auf die bewährte und leistungsstarke SciKit-Learn-Bibliothek zurückzugreifen. Die Verwendung dieser etablierten Implementierung ermöglicht eine präzise und effiziente Lösung, wobei der Fokus auf den zentralen Fragestellungen der Masterarbeit  liegt. Außerdem gewährleistet die Einbindung von SciKit-Learn neben der Robustheit der MDS-Umsetzung, auch die Reduktion der Entwicklungskomplexität im Hinblick auf die Algorithmusimplementierung. (developers scikit-learn, 2023)

Das Ergebnis des Skripts sind  Koordinaten für jeden Studiengang im zweidimensionalen Raum. Diese Koordinaten werden anschließend zusammen mit den vorher eingelesenen Meta-Daten zu den Studiengängen in JSON-Dateien gespeichert, die schließlich vom Backend über eine Webschnittstelle bereitgestellt werden.

Der Vorteil bei der Speicherung der berechneten Positionen in Form von JSON-Dateien liegt darin, dass die Daten wesentlich seltener geändert werden, als die Website aufgerufen wird. Somit wird effektiv Rechenleistung und Strom gespart. Gleichzeitig erhöht dieses Vorgehen die Performance der Website. Zur Festigung dieser These werden die Google Suchtrends von 2021 zum Suchbegriff „OTH Regensburg Studiengänge“ herangezogen (siehe Anhang A). Die Trends zeigen eine Jahressumme von 1194 Suchanfragen. Die Anzahl an Suchanfragen lassen sich nicht 1:1 in Seitenaufrufe projizieren, verdeutlichen aber dennoch einen ungefähren Richtwert. Die Datengenerierung hingegen wird nach mündlicher Zusage von Frau Rösel (Vizepräsidentin der OTH-Regensburg) nur bei größeren Änderungen an den Inhalten von Studiengängen, oder beim Hinzufügen bzw. Entfernen eines Studiengangs ausgelöst. Dies wiederum passiere meist insgesamt nur einmal pro Jahr.

Das Python-Skript hat neben der Berechnung und Speicherung der Koordinaten der Studiengänge noch eine weitere Aufgabe. Sobald der Benutzer auf der StudyMap eine Bubble anklickt, öffnet sich ein Popup mit Details zum entsprechenden Studiengang (siehe Konzept Abbildung ). Wie im weiteren Verlauf der Arbeit noch erläutert wird, können diese Details über die Administrationsoberfläche bearbeitet werden. Es ist jedoch notwendig, dass die Dateien mit diesen  Informationen bereits angelegt werden und sowohl die Werte der Inhaltskategorien für die Darstellung mit den Balken als auch die ähnlichen Studiengänge bereits in diese Dateien geschrieben werden. Diese Funktion übernimmt das Python-Skript.

Bei der Berechnung neuer Koordinaten wird für jeden eingelesenen Studiengang eine .json-Datei mit einem leeren Template erstellt. Wenn bereits eine Datei für den Studiengang aus einer vorherigen Datenquelle existiert, wird diese geöffnet und bearbeitet. Zur späteren Darstellung

der Inhaltskategorien werden anschließend die Originalwerte aus der Datenquelle in die .json-Datei kopiert (siehe folgendes Beispiel-JSON-Objekt `contents`).

Um ähnliche Studiengänge zu ermitteln, wird für jeden Studiengang anhand der zuvor berechneten Koordinaten der euklidische Abstand zu allen anderen Studiengängen ermittelt. Alle Studiengänge, die innerhalb einer bestimmten Grenze liegen, werden in eine Liste aufgenommen und schließlich in die .json-Datei des jeweiligen Studiengangs geschrieben (siehe folgendes Beispiel-JSON-Objekt `related_studies`).

Beispiel: Leeres Template des Studiengangs Architektur mit Inhalten und ähnlichen Studiengängen:

```
1  {
2      "name": "Architektur",
3      "abb": "A",
4      "supergroup": "",
5      "length": 0,
6      "course_url": "",
7      "starting_salary": "",
8      "description": "",
9      "local_companies": [],
10     "related_studies": [
11         { "name": "Bauklimatik", "abb": "GK" },
12         { "name": "Bauingenieurwesen", "abb": "B" }
13     ],
14     "contents": [
15         { "name": "Architektur", "score": 0.43 },
16         { "name": "Bau", "score": 0.25 },
17         { "name": "Design", "score": 0.09 },
18         { "name": "Digitalität", "score": 0.06 },
19         { "name": "Elektrotechnik", "score": 0.06 },
20         { "name": "Erneuerbare Energien", "score": 0.0 },
21         { "name": "Future Skills", "score": 0.03 },
22         ...
23     ]
24 }
```

Falls sich im Ordner weitere .json-Dateien befinden, die von Studiengängen stammen, die in der aktuellen Datenquelle nicht mehr aufgeführt sind, werden diese automatisch gelöscht.

6.1.3 Node.js - REST-API

Wie im vorherigen Abschnitt erläutert, ist neben dem Python-Programm auch eine Webschnittstelle erforderlich. In diesem Fall wird hierfür Node.js in Kombination mit der Bibliothek Express verwendet. Node.js ist eine JavaScript-Runtime-Umgebung, die auf der V8 JavaScript Engine von Google basiert und eine serverseitige Ausführung von JavaScript ermöglicht. (Foundation, 2023b) Express wiederum ist ein quelloffenes Webanwendungs-Framework für Node.js. Es erleichtert die Erstellung von Webanwendungen und APIs, indem es eine Reihe von Funktionen und Tools für den Aufbau von Webanwendungen bereitstellt. (Foundation, 2023a)

Der daraus entstehende Webservice ermöglicht den Zugriff, auf die vom Python-Skript generierten Dateien, über standardisierte REST-Endpunkte.

Eine REST-API (Representational State Transfer Application Programming Interface) ist eine zustandslose Schnittstelle, die es ermöglicht, über HTTP-Methoden, wie GET, POST oder PATCH, auf Ressourcen des Systems zuzugreifen und mit ihnen zu interagieren.

In der folgenden Tabelle 5 werden die REST-Endpunkte von StudyMap vorgestellt. Alle Einträge mit einem x in der Spalte **Auth.** setzen eine Authentifizierung, welche im späteren Unterabschnitt 6.2.1 näher erläutert wird, voraus:

Methode	Pfad	Rückgabewert	Auth.
GET	/bubbles/bachelor?mode=:mode	application/json	
GET	/details/:abb?mode=:mode	application/json	
GET	/data/history	application/json	x
PATCH	/data/history/restore/:id	application/json	x
GET	/data/generate?mode=:mode	application/json	x
GET	/data/download/:version	application/zip	x
POST	/data/upload	application/json	x
GET	/data/details	application/json	x
PATCH	/data/details/:filename	application/json	x

Tabelle 5: StudyMap-API: REST-Endpunkte

Bestimmte Endpunkte enthalten so genannte Query-Parameter. Query-Parameter sind Key-Value-Paare, die an das Ende einer URL angehängt werden können, um dem Webserver zusätzliche Informationen mitzuteilen. (branch, 2024) Tabelle 5 zeigt einige Endpunkte mit dem Query-Parameter mode.

Beispiel für eine Anfrage:

```
1 GET /bubbles/bachelor?mode=staging HTTP/1.1
2 Host: ...
```

Beschreibung der in Tabelle 5 aufgezählten Endpunkte:

GET /bubbles/bachelor?mode=:mode

Gibt alle gespeicherten Bachelor-Studiengänge in einem JSON-Positions-Array aus. Darin befindet sich der Name des Studiengangs, sein Kürzel, die zugehörige Supergruppe und die Farbe der Fakultät, um die Bubble entsprechend einzufärben. Außerdem enthält jeder Studiengang die relative Position in der StudyMap, die vom Algorithmus berechnet und auf 1 normiert wurde.



Querparameter: `:mode` ∈ 'staging', 'production'

Beispielausgabe:

```
1 {
2     "positions": [
3         [
4             "Architektur",
5             "AT",
6             "Architektur und Bau",
7             "#AOCCCC",
8             [
9                 0.4013795552124245,
10                1.0
11            ],
12            ...
13        ],
14    ]
```

GET /details/:abb?mode=:mode

Sobald alle Studiengänge abgerufen wurden und der Benutzer auf eine Bubbles klickt, öffnet sich ein Popup mit Details über den Studiengang. Der Klick ruft den Befehl `/details/:abb` auf, wobei `:abb` für das Kürzel des Studiengangs steht. Die Ausgabe ist erneut ein JSON und



enthält die Details des Studiengangs.

Queryparameter: `:mode` ∈ 'staging', 'production'

Beispielausgabe:

```
1 {
2     "name": "Informatik",
3     "abb": "IN",
4     "supergroup": "Bachelor of Science (B.Sc.)",
5     "length": 7,
6     "course_url": "https://www.oth-regensburg.de/studieren/...",
7     "starting_salary": "Überdurchschnittlich",
8     "description": "Sie möchten mit Informatik die Zukunft gestalten? Informatik
9         ist in allen Branchen ...",
10    "local_companies": [
11        { "name": "Vector", "url": "https://www.vector.com/" },
12        ...
13    ],
14    "related_studies": [
15        { "name": "International Computer Science", "abb": "ICS" },
16        ...
17    ],
18    "contents": [
19        { "name": "Architektur", "score": 0.0 },
20        { "name": "Bau", "score": 0.0 },
21        { "name": "Design", "score": 0.3 },
22        { "name": "Gesundheit", "score": 0.0 },
23        { "name": "Informatik", "score": 1.0 },
24        { "name": "Internationales", "score": 0.2 },
25        ...
26    ]
}
```

GET /admin/data/history

Gibt alle existierenden (maximal zehn) Datensicherungen in einem Array aus IDs aus.

Beispielausgabe:

```
1 ["1708014679710", "1708014600777"]
```

PATCH /admin/data/history/restore/:id

Stellt Datensicherung mit der ID :id wieder in die Staging-Umgebung her. :id ist ein Pfadparameter.



Beispielausgabe bei erfolgreicher Wiederherstellung (HTTP-Statuscode 200):

```
1 {  
2     "message": "Starte Bearbeitung: staging -> staging\\nBubbles wurden neu  
3         generiert.\\nFolgende .JSON-Vorlagen wurden nicht gefunden: B, ID, HK, PA,  
        IE, LP, SA, IW, EB, ISE, UI, MS, IR, REE, EI, ME, BE, MB, PT, ICS\\nEs wurden  
        insgesamt 4 Studiengangsvorlagen bearbeitet.\\nProzess erfolgreich beendet.\\n"  
3 }
```

Beispielausgabe bei Fehlerfall (HTTP-Statuscode 500):

```
1 {  
2     "message": "Fehler beim Wiederherstellen des letzten Standes."  
3 }
```

GET /admin/data/generate?mode=:mode

Erstellt anhand der bereits hochgeladenen Dateien eine neue Version und erzeugt in Falle einer Produktivversion eine Datensicherung.

Queryparameter: :mode ∈ 'staging', 'production'

Beispielausgabe bei Generierung einer Produktivversion (HTTP-Statuscode 200):

```
1 {  
2     "message": "Starte Bearbeitung: staging -> production\\nBubbles wurden neu  
3         generiert.\\nFolgende .JSON-Vorlagen wurden nicht gefunden: B, ID, HK, PA,  
        IE, LP, SA, IW, EB, ISE, UI, MS, IR, REE, EI, ME, BE, MB, PT, ICS\\nEs wurden  
        insgesamt 4 Studiengangsvorlagen bearbeitet.\\nProzess erfolgreich beendet.\\n"  
3 }
```

Beispielausgabe bei Fehlerfall beim Upload von neueren Dateien (HTTP-Statuscode 400):

```
1 {  
2     "message": "    self.dataset = Dataset(input_path)\n      File  
      '.../backend/generator/dataset.py', line 20, in __init__\n        for i, row in  
        enumerate(csvreader):\n          File '.../lib/python3.9/codecs.py', line 322, in  
          decode\n            (result, consumed) = self._buffer_decode(data, self.errors,  
            final)\nUnicodeDecodeError: 'utf-8' codec can't decode byte 0x9f in position  
            14: invalid start byte\n"  
3 }
```

GET /admin/data/download/:version

Erstellt aus Version :version eine .zip-Datei und erstellt daraus einen Download-Stream.
Für den Pfadparameter :version gilt :version ∈ 'staging', 'production', 'template'.

POST /admin/data/upload

Nimmt mehrere Dateien entgegen und speichert diese im Input des StudyMap-Algorithmus.
Die zu hochladenden Dateien müssen im Body der HTTP-Anfrage enthalten sein. Dabei gibt es die Regeln: dataset (1 Datei), faculties (1 Datei) und details (maximal 50 Dateien). Die hochgeladenen Dateien können anschließend per Aufruf von /admin/data/generate verarbeitet werden.

Beispielausgabe bei erfolgreichem Upload (HTTP-Statuscode 200):

```
1 {  
2     "message": "Dateien erfolgreich hochgeladen."  
3 }
```

Beispielausgabe bei fehlerhaften Download (HTTP-Statuscode 500):

```
1 {  
2     "message": "Fehler beim Dateiupload."  
3 }
```

Durch die in diesem Unterkapitel beschriebene Web-Schnittstelle kann das PixiJS-Frontend in Echtzeit die benötigten Informationen abrufen, um die interaktive Grafik der Studiengänge

zu erstellen. Die klare Trennung von Backend und Frontend gewährleistet eine effiziente Datenübertragung und ermöglicht eine dynamische Aktualisierung der Grafik bei Änderungen im Datensatz.

GET /admin/data/details

Dieser Endpunkt ruft die Dateinamen aller gespeicherten Studiengangsdetails ab und gibt sie aus. Wenn die  Datei per .csv-Datei hochgeladen werden, wird für jeden Studiengang, sofern noch nicht vorhanden, eine .json-Datei mit den Studiengangsdetails erstellt. Mithilfe dieses Endpunkts können die Dateinamen dieser Dateien abgerufen werden.

Beispielausgabe nach erfolgtem Upload von Studiengängen (HTTP-Statuscode 200):

```
1 [  
2   "A.json", "B.json", "BE.json", "BM.json", "DBM.json", "EI.json", "GK.json",  
3   "I.json", "IBM.json", "ICS.json", "ID.json", "IR.json", "ISE.json", "IT.json",  
4   "IW.json", "KI.json", "MB.json", "ME.json", "MS.json", "PA.json", "RE.json",  
5   "SC.json", "UI.json"  
6 ]
```

PATCH /admin/data/details/:filename

Nachdem alle verfügbaren Studiengänge mit  /admin/data/details abgerufen wurden, können mithilfe von /details/:abb die  Details zu einem einzelnen Studiengang abgefragt werden. Anschließend kann der Benutzer die  Informationen in der Administrationsoberfläche bearbeiten und die Änderungen mit diesem Endpunkt der API persistieren.

Der Request Body dieser Anfrage muss das bearbeitete JSON-Objekt der Studiengangsdetails sein. Der Request Body enthält die zu sendenden Daten der HTTP-Anfrage. (Parthiban, 2023)

Bei erfolgreicher Speicherung der Studiengangsdetails wird zusammen mit dem Statuscode 200 der neue Inhalt der Datei zurückgegeben. Ein Beispiel dazu befindet sich in der Schnittstellenbeschreibung für GET /details/:abb.

Wenn der Prozess fehlschlägt, gibt es zwei Arten von Fehlermeldungen zu unterscheiden:

- **Der Speicherprozess ist fehlgeschlagen:** In diesem Fall wird der Statuscode 500

zusammen mit  Fehlerbeschreibung als JSON ausgegeben: `{"message": "Fehler bei der Aktualisierung der Studiengangsdetails."}`.

- **Die angeforderte Datei :filename existiert nicht:** Der Statuscode 404 wird mit dem JSON `{"message": "Studiengangsdetails nicht gefunden."}` zurückgegeben.

6.1.4 Angular - Administrationsoberfläche

Um den im Unterabschnitt 4.3.1 aufgeführten Anforderungen wie Datensicherung und Trennung von Staging- und Produktivumgebung gerecht zu werden, ist die Implementierung einer Verwaltungssoftware notwendig.  Um die Entwicklung möglichst zukunftssicher, d.h. leicht erweiterbar zu gestalten, wird ein sogenanntes Webframework eingesetzt.

Ein Webframework ist eine Sammlung von Bibliotheken, Tools und Technologien und bietet Programmierern ein Grundgerüst für eine dynamische Webanwendung. Neben der Zeitersparnis durch die bereits vorhandenen Bausteine hat die Verwendung eines Webframeworks den Vorteil, dass es in der Regel eine standardisierte Struktur vorgibt, die es späteren Entwicklern erleichtert, an dem Projekt weiterzuarbeiten. Generell arbeiten die meisten Frameworks nach dem DRY-Prinzip (Don't repeat yourself), was den Code schlanker und damit weniger fehleranfällig macht. (DomainFactory, 2023)

Für die Entwicklung der Administrationsoberfläche wurde Angular als Webframework gewählt. Der Grund für die Wahl des Frameworks liegt in seiner Verbreitung, da Angular neben dem React-Framework und Vue.js eines der meistgenutzten Frontend-Frameworks ist. (Greif & Burel, 2022)

Angular ist ein komponentenbasiertes Framework zur Erstellung von skalierbaren Webanwendungen auf Basis der Programmiersprache TypeScript. Es enthält unter anderem Bibliotheken für das Routing zwischen Seiten, dynamische Formulare, Client-Server-Kommunikation und mehr (G. Inc., 2023).

Zusätzlich zum Angular-Framework wird ein CSS-Framework benötigt, um vordefinierte Styles und wiederverwendbare Komponenten wie z.B. Dialoge zu erhalten.

HTML steht für HyperText Markup Language und ist die Sprache des Internets. HTML ist eine Dokumentenbeschreibungssprache, mit der die Struktur von Webseiten definiert wird. Mithilfe von CSS wird dann das Layout und Aussehen der in HTML definierten Elemente (z.B. Schaltflächen) festgelegt. (Corporation, 2023a) CSS wiederum steht für Cascading Style Sheets und definiert Darstellungsregeln für HTML-Elemente. CSS-Regeln enthalten Definitionen zu

Größe, Form, Farbe, Animation und Layout der einzelnen HTML-Elemente einer Website. (Corporation, 2024b)

Im Fall von StudyMap wird Angular Material UI verwendet. Angular Material UI ist ein von Google entwickeltes CSS-Framework. Die Entscheidung für dieses Framework beruht darauf, dass es ebenso wie das Angular Framework von Google LLC entwickelt wurde und daher sehr gut parallel kombiniert werden kann. (LLC, 2024)

Die Implementierung der Authentifizierung in Angular ist vorerst nicht erforderlich, da diese bereits durch die Authentifizierungs-Middleware der Node.js-API abgedeckt ist.

Die Verwaltungssoftware hat in ihrer ersten Version fünf Seiten:



- Überblick
- Neue Daten einpflegen
- Studiengangsdetails
- Versionsverlauf
- Vorlagen

Überblick

Die Seite „Überblick“ bietet einen Überblick über das Gesamtsystem. Wie in Abbildung 23 zu sehen ist, enthält die Seite einen Link zur Vorschau der Staging-Umgebung sowie einen Link zur Vorschau der Produktivumgebung. Des Weiteren verfügt die Seite über einen roten Knopf. Bei Bestätigung wird die Staging-Version in die Produktivumgebung überführt und somit der Öffentlichkeit freigeschaltet.



Überblick

The screenshot shows the 'Überblick' (Overview) section of the Admin UI. It contains two main sections: 'Staging-Umgebung' and 'Produktiv-Umgebung'.
Staging-Umgebung: This section includes a note about testing new data without making it publicly visible, instructions to click 'Neue Daten einpflegen' in the sidebar, and a message that a successful upload and editing process makes the new version available via preview. It features two buttons: 'Vorschau' (Preview) and 'Staging -> Production' (with a warning icon).
Produktiv-Umgebung: This section notes that the name suggests the published version on the university website. It includes a note about preparing a new error-free version in the Staging environment for deployment to the Productive environment. It also features a 'Vorschau' button.

Abbildung 23: Admin UI: Überblick

Quelle: Eigene Darstellung

Direkt unterhalb befinden sich auf der Übersichtsseite außerdem zwei Architekturgrafiken, die dem Benutzer die Struktur von StudyMap und die damit verbundene Datensicherung über die Versionshistorie verdeutlichen sollen.

Neue Daten einpflegen

Auf der Seite zur Eingabe neuer Daten gibt es die Möglichkeit, zwei CSV-Dateien vom Computer auszuwählen (siehe Abbildung 24). Dies ist die Datenquelle, die in Kapitel X beschrieben wurde. Sie enthält alle Studiengänge und die dazugehörigen Werte der Inhaltskategorien. Die zweite Datei enthält die Abkürzungen der Fakultäten und den entsprechenden Farbcodes, um die Bubbles entsprechend den Fakultäten einzufärben.



Neue Daten einpflegen

Sobald der Upload abgeschlossen ist und die Überprüfung erfolgreich war, werden die Daten auf der [Staging-Version](#) zur Verfügung stehen.

Datenquelle hinzufügen

Bitte wählen Sie hier die .CSV-Datei mit den Studieninhalten aus:

bachelors.csv ×

Wähle Spreadsheet-Datei (CSV)

Fakultätsdatei hinzufügen

Bitte wählen Sie hier die .CSV-Datei mit den Fakultäten und zugehörigen Farben aus:

faculties.csv ×

Wähle Spreadsheet-Datei (CSV)

Hochladen

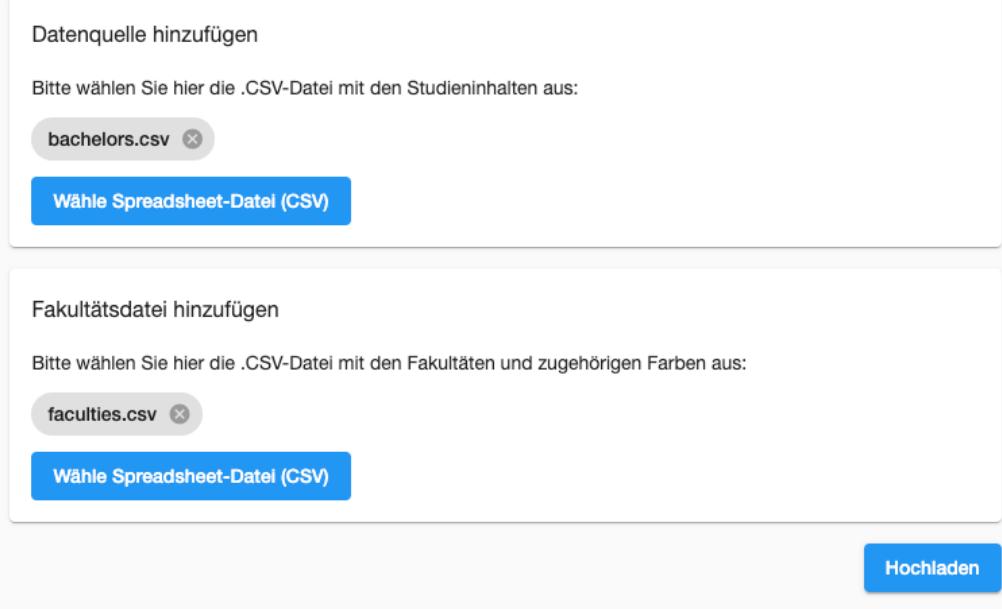


Abbildung 24: Admin UI: Neue Daten einpflegen

Quelle: Eigene Darstellung

Nach der Auswahl der beiden Dateien können diese mithilfe der Schaltfläche „Hochladen“ hochgeladen und verarbeitet werden. Sobald das Hochladen abgeschlossen ist, beginnt die Verarbeitung der Dateien. Anschließend erscheint ein Dialogfeld mit der Antwort des Servers (siehe Abbildung 25).

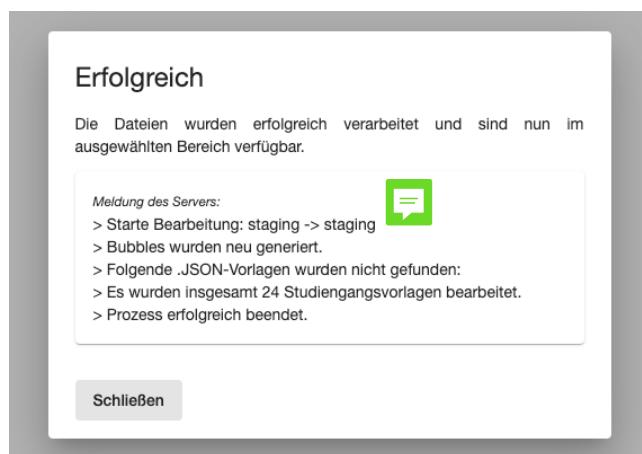


Abbildung 25: Admin UI: Neue Daten einpflegen - Ergebnisdialog

Quelle: Eigene Darstellung

Neue Daten einpflegen

Studiengangsdetails

Hier können von allen übermittelten Studiengängen die Details für den Popup-Dialog aktualisiert werden.

Wählen Sie einen Studiengang zur Bearbeitung aus:

Kürzel auswählen
I

Name des Studiengangs *
Informatik

Link des Studiengangs *
<https://www.oth-regensburg.de/studieren/studienganguebersicht/detailansicht/informatik-studieren-bachelor>

Studienart *
Bachelor of Science (B.Sc.)

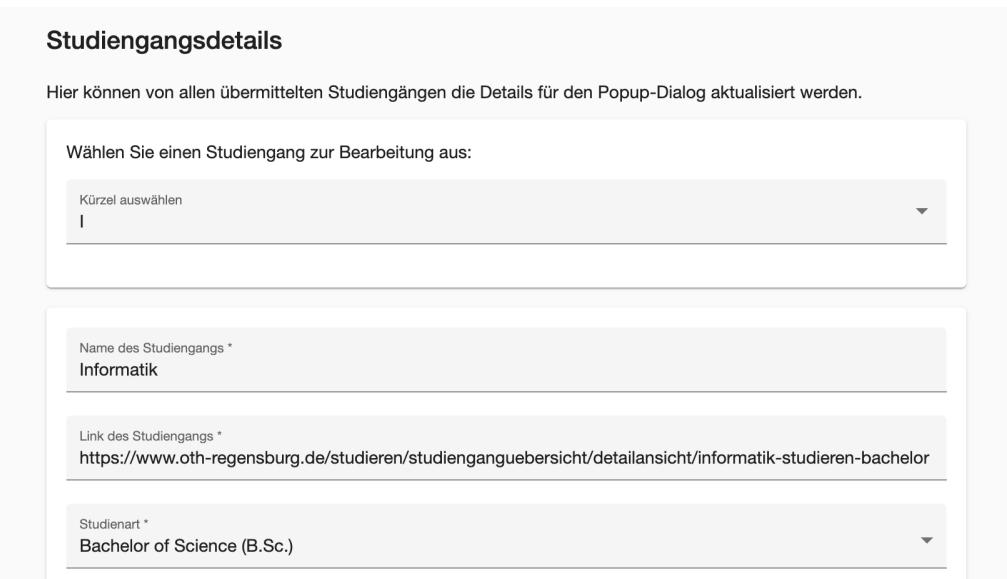


Abbildung 26: Admin UI: Studiengangsdetails verwalten (Ausschnitt)
Quelle: Eigene Darstellung

Sobald die Dateien hochgeladen wurden, wird für jeden Studiengang eine Studiengangsdetails-Datei angelegt. Diese kann mithilfe eines grafischen Formulars in der Administrationsoberfläche bearbeitet werden. Abbildung 26 zeigt einen Ausschnitt dieses Formulars. Zunächst muss der zu bearbeitende Studiengang oben im Auswahlfenster ausgewählt werden. Anschließend können alle statischen Inhalte des Popups, das sich beim Klick auf eine Bubble in der StudyMap öffnet, bearbeitet werden. Die Liste der lokalen Unternehmen ist ebenfalls Teil davon.



6.2 Softwarearchitektur

Das Gesamtbild StudyMap setzt sich aus den vorher genauer erläuterten Komponenten REST-API, PixiJS, Python-Backend und Node.js als Webserver für alle Dienste zusammen. Dieses Kapitel beschäftigt sich mit der Architektur und der Kommunikation der einzelnen Komponenten untereinander.

Obwohl die Softwarearchitektur erst in den letzten 30 Jahren an Bedeutung gewonnen hat, ist sie ein fester Bestandteil der Softwareentwicklung und definiert wesentliche tragende Säulen des Produkts. Dabei geht die Softwarearchitektur nicht auf die internen Details einer bestimmten Komponente ein. (Vogel et al., 2009) Sie soll komplexe Zusammenhänge übersichtlich darstellen und Fragen wie folgende beantworten:

- Wie hängen die Systembausteine miteinander zusammen?
- Welche Schnittstellen haben die Komponenten?
- Worauf sind Strukturierungen und Entscheidungen zurückzuführen?

Für StudyMap ist eine Darstellung der Softwarearchitektur aus mehreren Gründen essenziell.

Grund 1: Komplexität

Die Komplexität der Software wird übersichtlicher dargestellt. Da bei der Verwendung mehrerer Programmiersprachen wie JavaScript, TypeScript und Python schnell die Übersicht über die einzelnen Komponenten verloren gehen kann, besteht die Gefahr, ineffizienten oder im schlimmsten Fall fehlerhaften Code zu produzieren. Ein weiterer Grund ist die Einarbeitung zukünftiger Entwickler.

Grund 2: Einarbeitung

Da StudyMap im Rahmen dieser Masterarbeit entsteht, ist es abzusehen, dass es immer wieder wechselnde Stakeholder und Weiterentwickler für das Projekt geben wird. Diese müssen zwangsläufig eingearbeitet werden und das Gesamtprojekt verstehen. Ein Entwickler, dem der gesamte Workspace ohne Strukturgrafik mit vier miteinander kommunizierenden Projekten

präsentiert wird, kann  überfordert sein. Deshalb dient die Softwarearchitektur der effizienten Einarbeitung.

Grund 3: Sicherheit

Schließlich ist die Strukturgrafik für die Sicherheit der Hochschule von Bedeutung. StudyMap enthält einen geschützten Administrationsbereich. Es sollte im Vorfeld genau festgelegt werden, dass z.B. der Administrationsbereich nur über das VPN der Hochschule erreichbar ist, um die Wahrscheinlichkeit eines Hackerangriffs zu reduzieren. Solche Beziehungen lassen sich mithilfe der Definition einer Softwarearchitektur präzise festlegen. Außerdem benötigt man diese Darstellung explizit an der OTH-Regensburg, um einen Server vom Rechenzentrum für das Hosting von StudyMap zu erhalten. Weitere Informationen dazu finden Sie im Unterabschnitt 6.3.

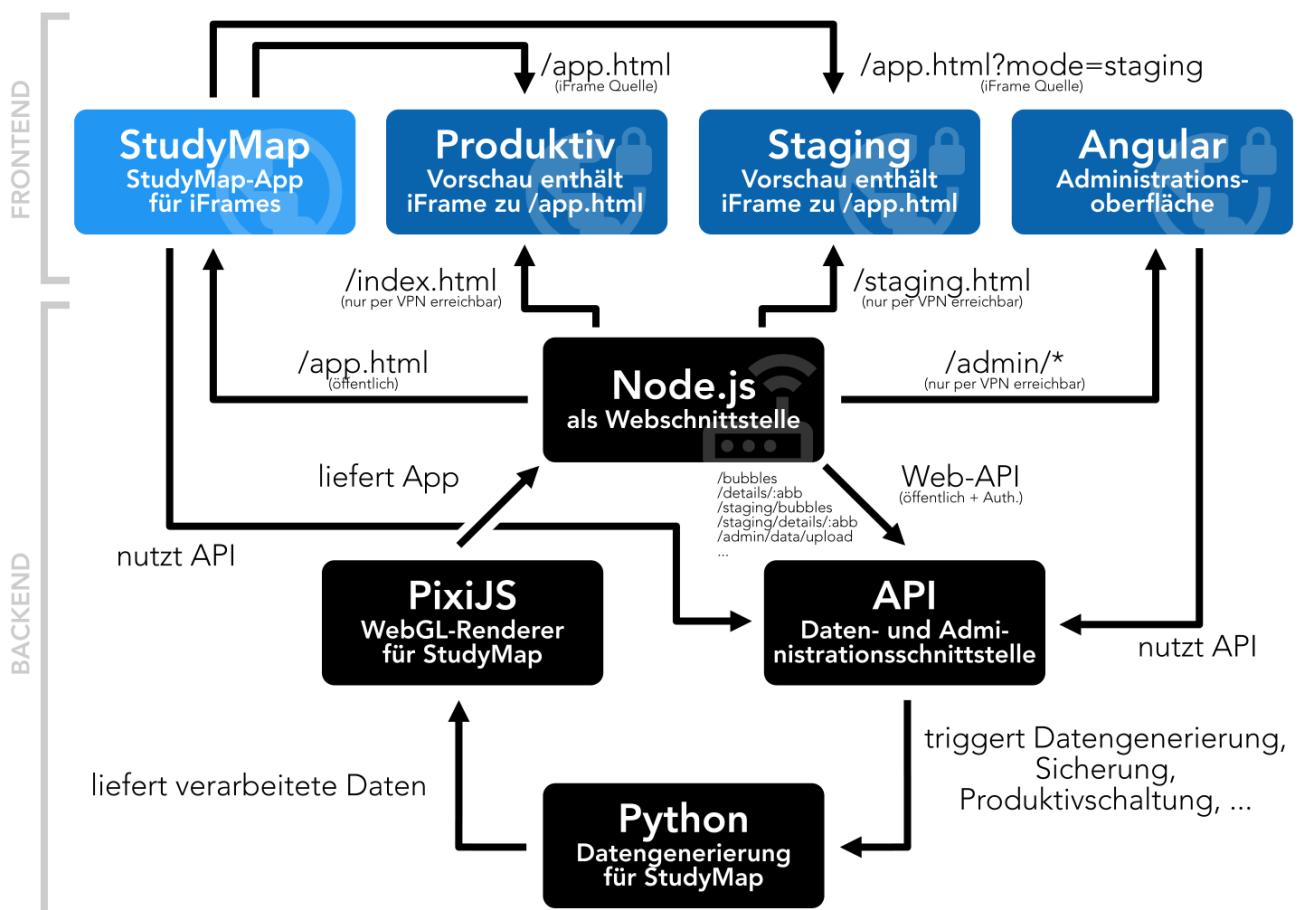


Abbildung 27: Softwarearchitektur von StudyMap
Quelle: Eigene Darstellung

Abbildung 27 zeigt die Softwarearchitektur von StudyMap. Die Anwendungskomponenten

werden durch abgerundete Rechtecke beschrieben. Die schwarzen Rechtecke repräsentieren die Backend-Komponenten, während die blauen die Frontend-Komponenten darstellen. Die Beziehungen zwischen den Komponenten werden durch Pfeile visualisiert.

6.2.1 Frontend-Komponenten

Alle Frontend-Komponenten werden über die Node.js-Webschnittstelle gehostet. Node.js wird mithilfe von Express genutzt, um neben der in Unterabschnitt 6.1.3 erläuterten REST-API auch die restlichen Webseiten wie die Anwendung selbst und die Administrationsoberfläche bereitzustellen. Die Frontend-Komponenten in Dunkelblau sind ausschließlich über eine aktive VPN-Verbindung der Hochschule erreichbar.

StudyMap

Die Frontend-Komponente StudyMap ist das fertige Produkt. Die StudyMap-App besteht aus dem PixiJS-Canvas, der mit Bootstrap verbunden ist, um den Studiengangsdetails-Dialog anzuzeigen (siehe Abbildung 5). Es handelt sich um eine reine Web-App ohne weitere Design-Elemente rund um den Canvas herum. Der Grund dafür ist, dass die StudyMap-Anwendung als iFrame in die Website der Hochschule eingebunden wird, d.h. die Anwendung muss den gesamten Bildschirminhalt ausfüllen, damit sich der iFrame später nahtlos in die Hochschulwebsite einfügt. Das HTML-Element iFrame erlaubt die Einbindung externer Webseiten in eine Seite. Ein klassisches Beispiel hierfür ist die Einbindung einer Anfahrtskarte eines Kartendienstleisters wie OpenStreetMap auf Unternehmenswebsites. (Corporation, 2024a) In dieser Arbeit wird der Studiengangsfinder in die Hochschulwebsite eingebettet.

Webschnittstelle: /app.html

Produktiv

Wie in Abbildung 27 dargestellt, ist die Produktiv-Schnittstelle eine HTML-Website, die die StudyMap-App per iFrame einbettet. Das Bestreben ist es, ein möglichst realitätsnahees Produktivsystem nachzubilden, weshalb die Produktivkomponente einen ähnlich breiten Platz für den iFrame verwendet, wie er auch auf der Website der OTH zur Verfügung steht. Die Produktivkomponente ist ausschließlich über das VPN der Hochschule erreichbar.

Webschnittstelle: /index.html

Staging

Die Staging-Umgebung ist identisch zur Produktivumgebung. Der einzige Unterschied besteht darin, dass die StudyMap-App nicht mit `/app.html`, sondern mit `/app.html?mode=staging` eingebunden wird. StudyMap soll die Daten aus dem Staging-Bereich des Backends abrufen und darstellen.

Mit dieser Testseite können neue Daten vor der Veröffentlichung getestet und optimiert werden. Die Staging-Komponente ist ebenfalls nur per VPN erreichbar.

Webschnittstelle: `/staging.html`

Angular

Die vierte und letzte Komponente ist die Angular-Administrationsoberfläche. Offensichtlich darf auch dieser Teil der Anwendung nur über eine authentifizierte VPN-Verbindung erreichbar sein. Der gesamte Administrationsbereich ist außerdem durch eine Basic-Auth-Middleware geschützt.

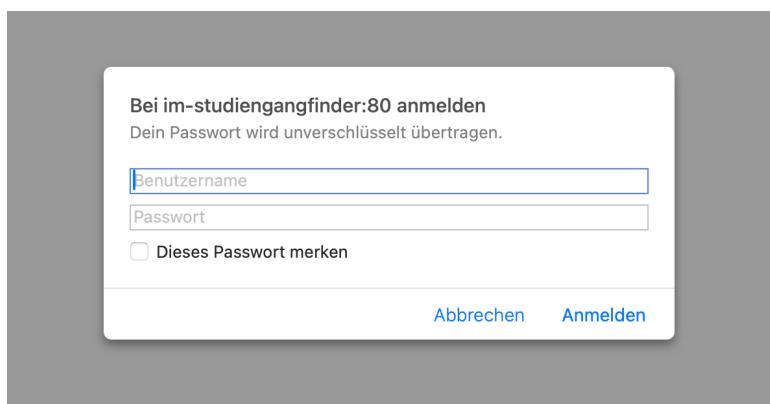


Abbildung 28: Basic-Auth-Middleware von StudyMap
Quelle: Eigene Darstellung

Basic-Auth ist ein allgemeines Authentifizierungsframework, das standardmäßig in der HTTP-Definition enthalten ist. (Corporation, 2023b) Es wird von allen gängigen Browsern unterstützt und erfordert nicht die Implementierung eines eigenen Authentifizierungsverfahrens. (Fyrd & Lensco, 2024) Wenn ein Webserver, in diesem Fall das Node.js-Webinterface, eine Authentifizierung anfordert, können Browser diese Anforderung verarbeiten und ein generisches Anmeldeformular anzeigen (siehe Abbildung 28).

In Abbildung 28 fehlt ein SSL-Zertifikat, wodurch keine verschlüsselte Verbindung mög-

lich ist. SSL-Zertifikate sind kleine Datendateien, die kryptografische Schlüssel digital an eine Organisation binden. Somit können HTTPS-Verbindungen genutzt werden, welche wiederum die Browserverbindungen verschlüsseln und somit auch die eingegebenen Daten vor Mitlesern schützen. (GlobalSign, 2023)

Daher wird auch eine Basic-Auth-Authentifizierung als unsicher eingestuft, wenn kein SSL-Zertifikat vorhanden ist. Um die Sicherheitsbedenken von StudyMap auszuschließen, sollte in Zukunft die Authentifizierung über einen externen Authentifizierungsprovider der Hochschule erfolgen. Dadurch muss StudyMap keine Anmeldedaten speichern und es besteht kein Risiko mehr für einen Hackangriff.

Nach Abschluss der Authentifizierung kann der Nutzer die Administrationsoberfläche nutzen. Die Web-App dient der Verwaltung der Anwendung und ihrer Daten. Hierfür werden HTTP-Anfragen genutzt, um die REST-API aufzurufen.

Webschnittstelle: /admin/*

6.2.2 Backend-Komponenten

Das Backend der Softwarearchitektur besteht neben der bereits im Unterabschnitt 6.2.1 erläuterten Node.js Webschnittstelle aus drei weiteren Komponenten:

1. PixiJS: WebGL-Renderer für StudyMap
2. API: Daten- und Administrationsschnittstelle
3. Python: Datengenerierung für StudyMap

PixiJS

Das PixiJS-Projekt ist ein eigenständiges JavaScript-Projekt, das die vorher erläuterte HTML-Datei `app.html` für die Frontend-Komponenten generiert und darin den WebGL-Renderer für StudyMap einbindet. Dieses Projekt enthält HTTP-Anfragen zur REST-API, um die Daten der Studiengänge abzurufen. Es ist jedoch wichtig zu beachten, dass diese Anfragen durch den Browser-Client bzw. die Frontend-Komponenten aufgerufen werden und somit streng genommen kein Teil der Backend-Komponente ist.

API

Die API (Application Programming Interface) bietet nicht nur die Lieferung von Positions- und Studiengangsdetailsdaten, sondern auch die Möglichkeit, die im Server gespeicherten Daten zu verwalten und zu ändern (siehe Unterabschnitt 6.1.3). Aus diesem Grund ist die Administrationsoberfläche sowie die StudyMap-App selbst mit der API verbunden.

Wenn jemand in der Administrationsoberfläche neue Dateien hochlädt, wird eine Anfrage an die API gesendet. Diese speichert die Dateien an der richtigen Stelle und löst schließlich die Generierung neuer Positionsdaten durch eine Anfrage an die Python-Komponente aus. Die Python-Komponente ist folglich ebenfalls mit der API angebunden.

Python

Die Python-Komponente hat im Wesentlichen nur eine **wirkliche** Verbindung zu einer anderen Komponente, nämlich zur REST-API, die Python aufruft. Die Python-Komponente enthält die Logik für die Generierung der auszuliefernden Dateien sowie für die Datensicherung. **Das** Python-Skript implementiert den MDS-Algorithmus zur Berechnung der genauen Positionsdaten anhand der Ähnlichkeiten zwischen den Studiengängen. Beide Anwendungsfälle werden durch die Kommandozeile der REST-API getriggert.

Es besteht eine weitere implizite Verbindung zwischen PixiJS und den Daten des Python-Skripts für die Anzeige der Bubbles. Die eigentliche Kommunikation zwischen den Komponenten erfolgt jedoch über die Programmierschnittstelle.



6.3 Software Deployment

Nachdem die Softwarearchitektur geklärt und implementiert wurde, folgt in diesem Abschnitt das Deployment. Software Deployment (Softwareverteilung) bezeichnet den Prozess der Konfiguration und Installation auf dem Zielserver. Dieser Prozess besteht in der Regel aus mehreren Schritten, wie Planung, Design, Testen, Terminplanung und Deployment. (A. Team, 2023) Im Falle von StudyMap beschränkt sich der Prozess auf die Schritte: Planung, Testen und Bereitstellung.

6.3.1 Planung

Der erste Schritt des Softwareverteilungs Prozesses ist die Planung, welcher diverse W-Fragen beantwortet. Es ist beispielsweise wichtig zu berücksichtigen, wie viele Nutzer die Anwendung später haben werden, welche Risiken zu erwarten sind und wie eine Authentifizierung aussehen könnte. (A. Team, 2023)

Bei dieser Arbeit waren bereits einige Planungspunkte vorgegeben, da an der Hochschule Regensburg ein spezielles IT-Verfahren zum Schutz der Hochschule eingesetzt wird. Um das Projekt StudyMap auch nach der Übergabe der Masterarbeit weiterführen zu können, wurde eine gewisse Normbeschreibung erstellt. In der folgenden Liste sind einige ausformulierte Punkte und Details des Planungsschrittes aufgeführt.

Wer sind die Verantwortlichen für die Software?

Die Verfahrensverantwortliche Person ist Prof. Dr.-Ing. Birgit Rösel, Vizepräsidentin der OTH-Regensburg. Sie hat den Bedarf der Studienorientierung erkannt und das Projekt initiiert. Während der Entwicklung ist Andreas Huber, der Autor dieser Arbeit, die technisch verantwortliche Person.

Die technisch Verantwortliche Person ist für die Wartung und das Patchmanagement zuständig. Es ist erforderlich, alle Pakete und Programme auf dem neuesten Stand zu halten, um die Wahrscheinlichkeit eines Hackerangriffs zu minimieren. Nach Abgabe dieser Arbeit wird das Amt an eine zum Zeitpunkt des Schreibens noch unbekannte Person übertragen.

Wie viele Benutzer wird die Anwendung haben?

Diese Frage ist insbesondere für die Leistung und die benötigten Serverressourcen von Bedeutung. Zum Zeitpunkt der Entwicklung liegen keine konkreten Besucherzahlen vor. Die Softwarearchitektur ist jedoch so geplant, dass die beschriebene REST-API lediglich .json-Dateien liest und zurückgibt - dies erfordert wenig Rechen- und Netzwerkleistung. Außerdem wird die interaktive Grafik clientseitig berechnet, also auf dem Gerät des Benutzers.

Aus den genannten Gründen wird ein leistungsschwacher virtueller Server mit den folgenden Spezifikationen vom Rechenzentrum angefordert:

- Prozessorkerne: 1 Kern
- Arbeitsspeicher: 8 GB
- Festplattenspeicher: 100 GB
- Betriebssystem: Debian 12

Wie kritisch ist ein Ausfall des Systems?

Ein Ausfall des Projekts ist weniger schwerwiegend, da es sich nur um ein optionales Feature der Hochschulwebsite handelt. Aus diesem Grund reicht es, das System innerhalb einer Wochenfrist wiederherzustellen. Darüber hinaus wird die Gesamtfunktion nicht ständig überwacht.

Werden personenbezogene Daten gespeichert?

Um rechtliche Fragen abzusichern, muss geklärt werden, ob im Falle eines Angriffs das Risiko besteht, persönliche Informationen zu verlieren. Daher ist es wichtig zu ermitteln, ob personenbezogene Daten gespeichert werden. StudyMap ist eine unabhängige Software, die Informationen über Studiengänge enthält, welche bereits veröffentlicht wurden. Das Hochladen und Speichern dieser Informationen in StudyMap durch eine verantwortliche Person ist ohne die Eingabe personenbezogener Daten möglich.

Aus diesem Grund gehören die verarbeiteten Informationen zur Hochschul-Informationsklasse „V0 - öffentlich“. Dies bedeutet, dass keine der verarbeiteten oder gespeicherten Informationen im Falle eines Diebstahls als kritisch eingestuft werden würde.

Wie wird sichergestellt, dass gewisse Bereiche nur per VPN zugänglich sind?



Wie wird die Authentifizierung sichergestellt?

Wie bereits in Unterabschnitt 6.2.1 erläutert, wird die Authentifizierung mittels einer Middleware im Webserver sichergestellt. Zudem wird vom Rechenzentrum eine separate IP-Adresse angefordert, um den Zugriff auf bestimmte Bereiche, wie beispielsweise den Administrationsbereich, nur mit aktiver VPN-Verbindung zu ermöglichen.

Welche Rollen sind vorgesehen?

In engem Zusammenhang mit der Frage der Authentifizierung steht die Frage, welche Rollen in der Anwendung vorgesehen sind. Im Studiengangsfinder gibt es lediglich zwei Rollen:

1. Administrator
2. Besucher

Die Besucher-Identität hat ausschließlich Zugriff auf die Frontend-Komponente `app.html`, welche die interaktive Grafik enthält. Eine Authentifizierung ist nicht erforderlich. Die Administrator-Identität hingegen hat Zugriff auf alle weiteren Frontend-Komponenten und kann somit die gespeicherten Daten verwalten.

Wer kümmert sich um die Aktualität der Daten?

Frau Rösel ist für die Aktualität der Daten verantwortlich, während der Entwicklung und nach der Übergabe des Projekts für einen unbestimmten Zeitraum. Nach Abgabe dieser Arbeit wird Andreas Huber, der technische Verantwortliche, Sie unverzüglich über den Prozess der Datenverwaltung durch die Administrationsoberfläche instruieren. Wer die Daten langfristig pflegt bleibt noch offen.

Werden regelmäßige System- und Datensicherungen durchgeführt?

Der Server von StudyMap befindet sich im zentralen Serverraum des Rechenzentrums der OTH-Regensburg. Dadurch wird sichergestellt, dass die Organisation täglich eine Systemsicherung durchführt.

Eine Datensicherung erfolgt nur beim Überführen eines Datenstands von Staging in die Produktivumgebung. Wie bereits beschrieben, speichert StudyMap immer die letzten 10 Produktivstände, um im Falle von korrupten Daten eine Datensicherung wiederherstellen zu können. Es gibt keine zeitgesteuerten Intervall-Datensicherungen.



Was ist der Ablauf bei einem eintretenden Sicherheitsvorfall?

Sollte es zu einer Kompromittierung des Servers kommen, muss die verantwortliche Person sofort Kontakt mit security@oth-regensburg.de aufnehmen und diese über den Vorfall informieren.

Wer stellt ein SSL-Zertifikat zur verschlüsselten Übertragung der Daten?



Damit die Verbindung zu StudyMap funktioniert, wird ein SSL-Zertifikat sowohl für die internen (VPN-Bereich) als auch für die externen Anfragen benötigt.

Wenn kein SSL-Zertifikat auf dem Webserver installiert ist, werden die Login-Anfragen unverschlüsselt verschickt. Ein Angreifer könnte dadurch die Zugangsdaten mitlesen und daraus einen weiteren Hackangriff starten.

Obwohl keine sicherheitsrelevanten Informationen beim Öffnen der Besucheransicht geteilt werden, benötigt StudyMap auch für den öffentlichen Bereich ein SSL-Zertifikat. Der Grund dafür ist, dass die Hochschulwebsite eine https-Verbindung erzwingt. Moderne Browser erlauben es nicht, einen iFrame mit einer unverschlüsselten Website auf einer verschlüsselten Website einzubinden. (Vyas, 2013)

6.3.2 Testen

In dieser Phase des Deployment-Prozesses wird eine Testumgebung erstellt, um zu überprüfen, ob alles wie vorgesehen funktioniert. Ziel ist die Schaffung einer möglichst realitätsnahen Umgebung, damit der endgültige Prozess der Bereitstellung auf dem Server der Universität mit möglichst wenigen unvorhersehbaren Komplikationen verbunden ist.

Verwendung von Docker zur Auslieferung

StudyMap wird auf dem Produktivserver in einem Docker-Container gehostet. Docker ist eine Open-Source-Plattform zur Erstellung und Bereitstellung von Anwendungen. Docker arbeitet

mit sogenannten Containern, die alle Pakete, benötigten Abhängigkeiten, Werkzeuge und Code enthalten, die für die Ausführung der Software notwendig sind. (Amazon Web Services, 2023)

Docker-Container visualisieren das Betriebssystem eines Servers. Jeder Container basiert auf einem Image, wie zum Beispiel einem Linux-Derivat. Auf diesem werden alle benötigten Pakete installiert. Docker-Images sind leichtgewichtige und dennoch effiziente eigenständige Pakete, da sie denselben System-Kernel verwenden und somit nicht für jede Anwendung ein vollständiges Betriebssystem benötigen. (D. Inc., 2024b)

Neben der Effizienz ist Sicherheit ein weiterer Vorteil. Anwendungen, die in Containern ausgeführt werden, sind voneinander isoliert und erlauben nur die Kommunikation, die durch die Konfiguration erlaubt ist. (D. Inc., 2024b)

 weiterer Vorteil ist die Möglichkeit, Docker-Container schnell von einem Server auf einen anderen zu übertragen, ohne den Server für die neue Anwendung konfigurieren zu müssen. Der Docker Container enthält alle notwendigen Netzwerk- und Software-Konfigurationen, die auf dem Zielserver unter Umständen nicht vorhanden sind. Dadurch können Probleme durch beispielsweise unterschiedliche Softwareversionen vermieden werden. Unterschiedliche Anwendungen benötigen oft unterschiedliche Paketversionen, wie zum Beispiel Python oder Python3. Solche Situationen können zu aufwendigen Konfigurationsprozessen führen, während mit Docker alles isoliert in einem Container genau für die Anwendung enthalten ist. (Amazon Web Services, 2023)

Docker-Image für StudyMap

Um StudyMap in einem Docker-Container auszuführen, wird ein Docker-Image benötigt, das alle Komponenten der Softwarearchitektur enthält. Für die Entwicklung dieses Images wird als Basisimage ein Node.js-Image verwendet. Auf diesem Basisimage werden alle Konfigurationen und zusätzlichen Komponenten festgelegt, die dann beim Starten des Containers automatisch installiert werden. Da Node.js als Webschnittstelle für den Studiengangsfinder dient, bildet es die ideale Basis für das Docker-Image. Nachfolgend ist das für StudyMap entwickelte Dockerfile abgebildet:

```

1 # Node version 18 as base image
2 FROM node:18
3
4 # Install python
5 RUN apt-get update && apt-get install -y python3 python3-pip
6
7 # Set work directory
8 WORKDIR /app
9
10 # Copy dir
11 COPY . .
12
13 # Install dependencies
14 RUN npm install
15 WORKDIR /app/admin
16 RUN npm install
17 RUN npm run build
18 WORKDIR /app/frontend
19 RUN npm install
20 RUN npm run build:prod
21 WORKDIR /app/backend/generator
22 RUN pip3 install -r requirements.txt --break-system-packages
23 WORKDIR /app
24
25 # Expose ports
26 EXPOSE 8080
27
28 # Launch
29 CMD ["npm", "start"]

```

Docker kann mithilfe einer Dockerfile in Form eines Textdokuments ein Image erstellen. In der Dockerfile befinden sich alle Befehle, um ein Image zusammenzustellen. (D. Inc., 2024a) Das für StudyMap entwickelte Dockerfile beginnt mit dem Befehl `FROM node:18`. Dies bedeutet, dass das Base Image `node` in Version 18 als Grundlage für dieses Image verwendet wird. Eine Dockerfile, die lediglich die erste Zeile zur Festlegung des Base Images enthält, würde dement sprechend ein bestehendes Image ohne Veränderung 1:1 kopieren. (T. N. D. Team, 2024)

In Zeile fünf wird die Anweisung gegeben, die Paketlisten des Systems zu aktualisieren und `python3` sowie `python3-pip` zu installieren, den offiziellen Paketmanager der Program-

miersprache Python. (pip developers, 2024) Mit dem **RUN**-Befehl in Dockerfiles kann der Container angewiesen werden, diese Anweisung in der Befehlszeile des Containers auszuführen. Das Node-Image basiert auf dem Alpine-Linux-Image, welches das Linux-Paketverwaltungswerkzeug APT (Advanced Packaging Tool) enthält. (T. N. D. Team, 2024) Dieses Werkzeug ermöglicht es, Pakete, wie z.B. `python3`, auf einfache Art und Weise per Kommandozeile mit dem Befehl `apt-get install` zu installieren. (Ubuntu, 2024)

In Zeile acht wird das Arbeitsverzeichnis des Containers auf den Pfad `/app` gesetzt, indem der Befehl `WORKDIR` verwendet wird. Alle folgenden Befehle werden relativ zu diesem Ordner ausgeführt. Falls der angegebene Pfad nicht vorhanden ist, wird Docker ihn im Dateisystem des Containers erstellen. (D. Inc., 2024a)

Mit **COPY** wird der gesamte Inhalt des Ordners, in dem sich die Dockerfile befindet, in den Ordner kopiert, in dem sich der Container gerade befindet. (D. Inc., 2024a) Die Dockerfile befindet sich im äußersten Verzeichnis des StudyMap-Git-Repositorys. Das bedeutet, dass alle Dateien und Unterordner des Studiengangsfinders in den Container kopiert werden. Nachdem das `WORKDIR` auf `/app` gesetzt wurde, werden alle Dateien dementsprechend nach `/app` kopiert.

Zeilen 14 bis 23 installieren alle Abhängigkeiten der einzelnen Subprojekte und komplizieren die Dateien zu ausführbaren Anwendungen. In Zeile 14 wird eine Version der Node.js-Webschnittstelle und REST-API erstellt. Anschließend wird in Zeile 15 das Arbeitsverzeichnis auf den Subordner der Administrationsoberfläche gesetzt und mit `RUN npm install` und `RUN npm build` dessen Abhängigkeiten installiert und kompiliert. Schließlich werden in den Zeilen 18 bis 20 des PixiJS-Projekts dieselben Schritte durchgeführt. Abschließend werden die Python-Abhängigkeiten mithilfe des zuvor vorgestellten Paketmanagers `pip3` und einer im Unterordner liegenden Abhängigkeitsliste `requirements.txt` installiert. Python benötigt keinen Build-Prozess, da das Skript zur Laufzeit von der Python-Engine gelesen und ausgeführt wird. Zum Schluss wird in Zeile 23 das Arbeitsverzeichnis auf `/app` zurückgesetzt.

In Zeile 26 wird mithilfe des Kommandos `EXPOSE 8080` Docker darüber informiert, dass der Netzwerkport 8080 der zu hörende Port ist. Ein Netzwerkport ist eine definierte Nummer, die die Kommunikation für einen bestimmten Dienst empfängt oder überträgt. (Wright, 2022) Ein Beispiel aus dem Alltag wäre die Wohnungsnummer in einem Mehrparteienhaus, damit die Post den Brief zur richtigen Wohnung innerhalb des Hauses zustellen kann. Der `EXPOSE`-Befehl informiert lediglich über den Port, schaltet ihn jedoch nicht frei. Um den Port freizugeben, muss beim Start des Containers der Parameter `-p` für Port angegeben werden. (D. Inc., 2024a)

Die letzte Zeile enthält lediglich den Startbefehl für das Node.js Projekt, welches anschlie-

ßend alle weiteren Projekte hostet. Wenn der Container gestartet wird, wird der Kommandozeilenbefehl ausgeführt, der mit dem **CMD**-Befehl definiert wurde. Aus diesem Grund darf es im Dockerfile auch nur einen **CMD**-Befehl geben. (D. Inc., 2024a)

Um das Docker-Image mithilfe des Dockerfiles zu bauen und auszuführen, müssen nun folgende zwei Befehle in der Kommandozeile ausgeführt werden:

```
1 docker build -t studymap .
2 docker run -p 8080:8080 studymap
```

Der erste Befehl führt den Build-Prozess anhand der Dockerfile im aktuellen Verzeichnis aus und benennt das Image als **studymap**. Der zweite Befehl erstellt einen Container mit dem gerade erstellten Image und leitet den Port 8080 aus dem Container auf den Port 8080 des Betriebssystems des Servers weiter.

Persistierung der Daten durch Docker Volumes und Docker Compose

Wie bereits erläutert, werden in StudyMap Dateien von außen in den Container hochgeladen und verarbeitet. Wenn ein Docker-Container jedoch neu gestartet wird, wird sein Zustand zurückgesetzt. Das bedeutet, dass alle veränderten Dateien wieder im Originalzustand sind und alle neu hinzugefügten Dateien gelöscht werden. Aus diesem Grund ist die Verwendung von Docker Volumes zur Persistenz über einen Neustart hinaus notwendig.

Docker Volumes ermöglichen es, Daten von Containern betriebssystemunabhängig zu persistieren. Sie ermöglichen die Erstellung von Container-übergreifendem Speicher und sind dabei deutlich effizienter und performanter als Bind-Mounts eines Betriebssystems. (D. Inc., 1b v. u. Z.)

Für die Verwendung von Docker-Volumes muss der Kommandozeilenbefehl zum Starten des Containers angepasst werden. Eine Alternative hierzu ist die Verwendung von Docker Compose. Docker Compose wurde entwickelt, um mehrere Container effizient zu steuern und eine klare Struktur von vielen Containern zu ermöglichen. Hier gibt es neben den Dockerfiles auch eine `docker-compose.yml`-Datei, die alle Informationen zu den Containern bezüglich Netzwerk, Speicher und freizugebenden Ports enthält. Da der Kommandozeilenbefehl zum Starten des StudyMap-Docker-Containers aufgrund der Ports und Volumes dokumentiert werden müsste, bietet sich die Verwendung von Docker Compose an. Die benötigten Volumes sowie der freizugebende Port 8080 werden in einer Datei festgehalten. (D. Inc., 1a v. u. Z.) Der Administrator kann die StudyMap-Anwendung mit folgenden konsistenten Befehlen verwalten:

```

1 docker compose up -d      # Startet den Container im Hintergrund (-d)
2 docker compose down       # Stoppt den Container
3 docker compose restart    # Startet den Container neu

```

Die im vorherigen Abschnitt erwähnte Datei `docker-compose.yml` sieht für den Studiengangsfinder folgendermaßen aus:

```

1 version: '3.8'

2

3 services:
4   studymap:
5     build: .
6     ports:
7       - "80:8080"
8     volumes:
9       - generated-data:/app/gdata
10    command: ["npm", "start"]

11

12 volumes:
13   generated-data:
14   input-files:

```

Die Container-Definition für den Studiengangsfinder befindet sich in den Zeilen vier bis elf. In Zeile vier wird der Name des Containers angegeben und in Zeile fünf das Image, das der Container verwenden soll. In diesem Fall ist **das ein wichtiger Punkt**, denn die `docker-compose.yml`-Datei liegt im selben Verzeichnis wie die Dockerfile. Das bedeutet, dass Docker Compose automatisch die Dockerfile sucht, das Image baut und als Grundlage für den Container verwendet. In Zeile sieben wird der freizugebende Port festgelegt. Der Container gibt den Port 8080 frei, welcher dann auf den Systemport 80 geleitet wird. In Zeile acht bis zehn werden die Volumes festgelegt. Hierbei handelt es sich um zwei Verzeichnisse. Einmal das Verzeichnis, in dem die hochgeladenen Dateien des Benutzers landen und die Datensicherungen angelegt werden, und das zweite Volume, welches die durch den Algorithmus generierten Dateien enthält. Zeile elf ist äquivalent zum `CMD`-Befehl aus der bereits erläuterten Dockerfile. In den Zeilen 13 bis 15 werden abschließend die verwendeten Volumes definiert.

Mithilfe der genannten Docker-Dateien kann die Anwendung und das Deployment nun entweder lokal auf dem Entwicklungsrechner oder auf Testservern getestet werden. Nachdem

der Container erfolgreich gestartet werden kann, kann mit dem letzten Schritt des Deployments der eigentlichen Bereitstellung begonnen werden.

6.3.3 Bereitstellung

Für eine reibungslose und fehlerfreie Bereitstellung auf dem Hochschulserver ist eine sorgfältige Planung erforderlich. Die folgenden Schritte geben eine Reihenfolge und eine Struktur für die Durchführung des Deployments vor:

1. Server aktualisieren
2. Speicherort für StudyMap festlegen
3. Docker und Docker Compose installieren
4. StudyMap auf den Server kopieren
5. Docker Compose ausführen

1. Server aktualisieren

Zuerst müssen der Server und seine Paketlisten aktualisiert werden. In Debian kann dies mit den folgenden Befehlen durchgeführt werden:

```
1 apt-get update  
2 apt-get upgrade
```

2. Speicherort für StudyMap festlegen

Es muss anschließend der Speicherort für StudyMap festgelegt werden. Der Standardordner für Docker-Container unter Linux ist `/var/lib/docker`. Da die Partition auf dem vom Rechenzentrum bereitgestellten Server nur 3,9 Gigabyte Speicherplatz hat, muss auf das Home-Verzeichnis mit 13 Gigabyte ausgewichen werden. Für diese Änderung ist eine Anpassung der Datei `/etc/docker/daemon.json` erforderlich:

```
1 {  
2     "data-root": "/home/docker-data"  
3 }
```

3. Docker und Docker Compose installieren

Der nächste Schritt besteht in der Installation der Docker Engine und dem Plugin Docker Compose. Die Pakete hierfür sind nicht in den Standard-Paketlisten von APT enthalten. Aus diesem Grund wird auf der Docker Installationswebsite die Step-by-Step-Anleitung befolgt, um die Paketlisten zu erweitern und Docker schließlich per Paketmanager zu installieren.

Offizielle Installationsanleitung für Debian: <https://docs.docker.com/engine/install/debian/>

4. StudyMap auf den Server kopieren

Um das Projekt auf den Server zu übertragen, wird das gesamte Git-Repository, einschließlich des Dockerfiles und der docker-compose.yml-Datei, per SCP auf den Server kopiert. Der Quellcode befindet sich derzeit auf der Plattform GitHub. Es ist möglich, Repositories direkt als .zip-Datei herunterzuladen. Falls es nicht möglich ist, das Repository als .zip-Datei herunterzuladen, kann alternativ das lokale Entwicklungsprojekt als Zip-Datei komprimiert und übertragen werden. In diesem Fall ist es jedoch wichtig, alle Build-Dateien, Bibliotheken und Abhängigkeiten zu entfernen, die sowieso nachinstalliert werden müssen, um unnötig große Datentransfers zu vermeiden.

Anschließend genügt ein Kommandozeilenbefehl, um die Anwendung sicher auf den Server zu übertragen:

```
1 scp studymap.zip root@im-studiengangfinder:/home
```

Der Befehl lädt die Datei `studymap.zip` auf den Server mit dem Hostnamen `im-studiengangfinder` hoch und verwendet als Login-Username den Benutzer `root`. Das Zielverzeichnis ist `/home`. Dieser Prozess erfordert eine aktive VPN-Verbindung zur Hochschule und einen autorisierten SSH-Key.

5. Docker Compose ausführen

Nachdem die Datei hochgeladen und entpackt wurde, kann das Docker-Image und der Docker-Container mithilfe des zuvor erklärten Docker-Compose-Befehls über eine SSH-Verbindung zur Kommandozeile des Servers erstellt und gestartet werden.

7 Diskussion und Ausblick

- 7.1 Zusammenfassung der Ergebnisse und wichtigsten Erkenntnisse**
- 7.2 Vergleich mit anderen Studienorientierungs-Tools**
- 7.3 Potenzielle Erweiterungen und zukünftige Anwendungen**

Quellenverzeichnis

- Amazon Web Services, I. (2023). *Was ist Docker? / AWS* [Was ist Docker?]. Verfügbar 17. Februar 2024 unter <https://aws.amazon.com/de/docker/>
- Batista, D. & Correia, M. (2023). *Github Compare*. <https://www.githubcompare.com/pixijs/pixijs+paperjs/paper.js>
- Beckmann, J., Piepenburg, J. G., Fervers, L. & Jacob, M. (2021). *Verbesserung der Studienorientierung von unentschiedenen Schülerinnen und Schülern durch Studienberatung? Ergebnisse aus dem Projekt „Frühe Prävention von Studienabbruch“*. Verfügbar 11. April 2023 unter https://link.springer.com/chapter/10.1007/978-3-658-32892-4_14
- Bostock, M. & Observable. (2023). *D3.js - The JavaScript library for bespoke data visualization*. <https://d3js.org/>
- branch. (2024). *Query parameters* [Branch]. Verfügbar 15. Februar 2024 unter <https://www.branch.io/glossary/query-parameters/>
- Commission, E. (n. d.). *Europäisches System zur Übertragung und Akkumulierung von Studienleistungen (ECTS) / European Education Area* [Europäisches System zur Übertragung und Akkumulierung von Studienleistungen (ECTS)]. Verfügbar 19. Februar 2024 unter <https://education.ec.europa.eu/de/education-levels/higher-education/inclusive-and-connected-higher-education/european-credit-transfer-and-accumulation-system>
- Company, T. M.-B. (2023). *MBTI® Persönlichkeitstypen*. Verfügbar 12. Februar 2023 unter <https://eu.themeyersbriggs.com/de-DE/tools/MBTI/MBTI-personality-Types>
- Corporation, M. (2023a, 18. Juli). *HTML: HyperText markup language / MDN* [HTML: HyperText markup language]. Verfügbar 11. Februar 2024 unter <https://developer.mozilla.org/en-US/docs/Web/HTML>
- Corporation, M. (2023b, 20. Dezember). *HTTP authentication - HTTP / MDN*. Verfügbar 15. Februar 2024 unter <https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication>
- Corporation, M. (2023c, 25. Oktober). *WebGL: 2d and 3d graphics for the web - web APIs / MDN*. Verfügbar 15. Februar 2024 unter https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API
- Corporation, M. (2024a, 10. Februar). *<iframe>: The inline frame element - HTML: HyperText markup language / MDN*. Verfügbar 15. Februar 2024 unter <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe>
- Corporation, M. (2024b, 1. Januar). *What is CSS? - learn web development / MDN* [What is CSS?]. Verfügbar 11. Februar 2024 unter https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS

- developers scikit-learn, s.-l. (2023). *scikit-learn: Machine Learning in Python*. Verfügbar 12. Februar 2023 unter <https://scikit-learn.org>
- DomainFactory. (2023, 15. November). *Die beliebtesten Web Application Frameworks im Überblick*. Verfügbar 11. Februar 2024 unter <https://www.df.eu/blog/die-beliebtesten-web-application-frameworks/>
- etimberg, kurkle, benmccann, tannerlinsley & more. (2023a). *Chart.js - Simple yet flexible JavaScript charting library for the modern web*. <https://www.chartjs.org/>
- etimberg, kurkle, benmccann, tannerlinsley & more. (2023b). *Paper.js - MouseEvent*. <http://paperjs.org/reference/mouseevent/>
- Foundation, O. (2023a). *Express - Node.js web application framework*. Verfügbar 12. Februar 2023 unter <https://expressjs.com>
- Foundation, O. (2023b). *Node.js*. Verfügbar 12. Februar 2023 unter <https://nodejs.org>
- Fyrd & Lensco. (2024, 7. Januar). *headers HTTP header: WWW-Authenticate: ‘Basic’ authentication / Can I use... Support tables for HTML5, CSS3, etc* [Can I use: Basic authentication]. Verfügbar 15. Februar 2024 unter https://caniuse.com/mdn-http_headers_www-authenticate_basic
- GlobalSign. (2023, 23. Januar). *Was ist ein SSL Zertifikat* [GlobalSign]. Verfügbar 15. Februar 2024 unter <https://www.globalsign.com/de-de/ssl-information-center/was-ist-ein-ssl-zertifikat>
- Greif, S. & Burel, E. (2022). *State of JavaScript 2022: Front-end frameworks* [State of JavaScript 2022]. Verfügbar 11. Februar 2024 unter <https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/>
- Hachmeister, C.-D., Harde, M. E. & Langer, M. F. (2023, 15. Dezember). Einflussfaktoren der Studienentscheidung.
- Heublein, U., Hutzsch, C., Schreiber, J., Sommer, D. & Besuch, G. (2010). *Ursachen des Studienabbruchs in Bachelor- und in herkömmlichen Studiengängen*. Verfügbar 11. April 2023 unter <https://hsdbs.hof.uni-halle.de/documents/t1944.pdf>
- Inc., D. (1a v. u. Z.). *Docker compose overview* [Docker documentation]. Verfügbar 18. Februar 2024 unter <https://docs.docker.com/compose/>
- Inc., D. (1b v. u. Z.). *Volumes* [Docker documentation]. Verfügbar 18. Februar 2024 unter <https://docs.docker.com/storage/volumes/>
- Inc., D. (2024a). *Dockerfile reference* [Dockerfile reference]. Verfügbar 17. Februar 2024 unter <https://docs.docker.com/engine/reference/builder/>
- Inc., D. (2024b). *What is a container? / docker* [What is a container?]. Verfügbar 17. Februar 2024 unter <https://www.docker.com/resources/what-container/>
- Inc., G. (2023, 15. August). *Angular - What is Angular?* [Angular - What is Angular?]. Verfügbar 11. Februar 2024 unter <https://angular.io/guide/what-is-angular>

- Lehni, J. & Puckey, J. (2023). *Paper.js - The Swiss Army Knife of Vector Graphics Scripting*.
<http://paperjs.org/>
- LLC, G. (2024). *Angular material UI* [Angular material UI]. Verfügbar 11. Februar 2024 unter
<https://material.angular.io/>
- Parthiban, P. (2023, 9. Juni). *Essential guide to HTTP POST request method* [Atatus blog - for DevOps engineers, web app developers and server admins.]. Verfügbar 22. Februar 2024 unter <https://www.atatus.com/blog/http-post-request-method/>
- pip developers, T. (2024). *pip documentation v24.0* [pip]. Verfügbar 17. Februar 2024 unter
<https://pip.pypa.io/en/stable/>
- PixiJS. (2023). *PixiJS - The HTML5 Creation Engine*. <https://pixijs.com/>
- PixiJS. (2024a). *Interaction / PixiJS* [Interaction]. Verfügbar 22. Februar 2024 unter <https://pixijs.com/guides/components/interaction>
- PixiJS. (2024b, 27. Januar). *PixiJS API Documentation* [PixiJS API Documentation]. Verfügbar 15. Februar 2024 unter <https://pixijs.download/dev/docs/index.html>
- Preim, B. & Dachselt, R. (2010, 8. September). Interaktive Systeme: Band 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung.
- Regensburg, O. T. H. (2023). *Die OTH Regensburg - Hochschulprofil*. <https://www.oth-regensburg.de/die-oth/ueber-uns>
- Team, A. (2023, 18. Mai). *Was ist Software Deployment? / Ateras Blog* [Was ist Software Deployment?]. Verfügbar 16. Februar 2024 unter <https://www.atera.com/de/blog/was-ist-software-deployment/>
- Team, N. (2023). *NumPy - The fundamental package for scientific computing with Python*. Verfügbar 12. Februar 2023 unter <https://numpy.org>
- Team, T. N. D. (2024). *node - Official Image / Docker Hub* [node - Official Image]. Verfügbar 17. Februar 2024 unter https://hub.docker.com/_/node/
- Three.js. (2023). *Three.js - JavaScript 3D Library*. <https://threejs.org/>
- Ubuntu, C. L. (2024). *Package management* [Package management]. Verfügbar 17. Februar 2024 unter <https://ubuntu.com/server/docs/package-management>
- Uniturm.de. (2023). *Studiumsfinder*. Verfügbar 12. Mai 2023 unter <https://www.uniturm.de/studienwahl/studiumsfinder>
- Vogel, O., Arnold, I., Chughtai, A., Ihler, E., Kehrer, T., Mehlig, U. & Zdun, U. (Hrsg.). (2009). Einleitung. In *Software-Architektur: Grundlagen — Konzepte — Praxis* (S. 1–21). Spektrum Akademischer Verlag. https://doi.org/10.1007/978-3-8274-2267-5_1
- Vyas, T. (2013, 10. April). *Mixed content blocking enabled in firefox 23!* [Tanvi vyas]. Verfügbar 17. Februar 2024 unter <https://blog.mozilla.org/tanvi/2013/04/10/mixed-content-blocking-enabled-in-firefox-23/>

Wright, G. (2022, November). *Was ist Port? - Definition von Computer Weekly* [Was ist Port?]. Verfügbare 18. Februar 2024 unter <https://www.computerweekly.com/de/definition/Port>

Zaytsev, J., Kienzle, S. & Bogazzi, A. (2023). *Fabric.js is a powerful and simple Javascript HTML5 canvas library*. <http://fabricjs.com/>

Anhang

A Google Search Trends 2021: „OTH Regensburg Studiengänge“

Woche	„OTH Regensburg Studiengänge“
2021-01-03	0
2021-01-10	0
2021-01-17	0
2021-01-24	0
2021-01-31	54
2021-02-07	28
2021-02-14	37
2021-02-21	0
2021-02-28	0
2021-03-07	0
2021-03-14	39
2021-03-21	39
2021-03-28	34
2021-04-04	0
2021-04-11	0
2021-04-18	0
2021-04-25	28
2021-05-02	0
2021-05-09	33
2021-05-16	0
2021-05-23	72
2021-05-30	0
2021-06-06	0
2021-06-13	0
2021-06-20	0
2021-06-27	39
2021-07-04	0
2021-07-11	41
2021-07-18	52
2021-07-25	0
2021-08-01	71
2021-08-08	75
2021-08-15	59
2021-08-22	0
2021-08-29	30
2021-09-05	0
2021-09-12	33
2021-09-19	48
2021-09-26	0
2021-10-03	0
2021-10-10	0
2021-10-17	33
2021-10-24	67
2021-10-31	0
2021-11-07	100
2021-11-14	0
2021-11-21	50
2021-11-28	0
2021-12-05	0
2021-12-12	93
2021-12-19	39
2021-12-26	0

C Studienverlaufsplan: International Computer Science

Curriculum International Computer Science B.Sc.																															
	ECTS																														
Semester	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Sum ECTS
1	Mathematics 1							Programming 1							Automata, Formal Languages, and Computation				Business Administration				Language Track 1 A or B *)				30				
2	Mathematics 2							Programming 2							Computer Systems				Networking				Language Track 2 A or B *)				30				
3	Databases							Global Software Engineering							Operating Systems				Statistics				Language Track 3 A or B *)				30				
4	Web Technology Project				Software for the Global Market			Algorithms and Data Structures							Computer Architecture							Language Track 4 A or B *)				30					
5	Industrial Placement + Seminar																											30			
6	Core Module 1 Scientific Seminar				Core Module 2 Int. Software Development *)				Core Module 3 Information Security *)				Core Module 4 Big Data Technology *)				FWPM 1				FWPM 2				30						
7	Bachelor Thesis											Bachelor Seminar		Core Module 5 Advanced Algorithms *)				Core Module 6 Stochastic Algorithms *)				FWPM 3				30					

*) Core Modules 2 - 6:
The designated course modules are intended to provide an orientation to the planned course offerings in the 3rd stage of study. The list of course modules mentioned can still be extended at any time within the program growth.

*) Language Track 1 -
Track A: German
Track B: Spanish