

Bachelorarbeit

Tim-Lukas Maeke

Entwurf und prototypische Umsetzung eines Tools
zum Forschungsmanagement von Professoren

Tim-Lukas Maeke

Entwurf und prototypische Umsetzung eines Tools zum Forschungsmanagement von Professoren

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Bachelor of Science Wirtschaftsinformatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Ulrike Steffens
Zweitgutachter: Prof. Dr. Olaf Zukunft

Eingereicht am: 07. Juni 2019

Tim-Lukas Maeke

Thema der Arbeit

Entwurf und prototypische Umsetzung eines Tools zum Forschungsmanagement von Professoren

Stichworte

Forschungsmanagement, Webanwendung

Kurzzusammenfassung

Diese Bachelorthesis befasst sich mit der prototypischen Umsetzung eines Forschungsmanagement Tools zur Verbesserung des Managements von Professoren. Es wird sich damit beschäftigt, die verteilten Informationen von Forschungsthemen und Forschungsprojekten zu bündeln und so aufzubereiten, dass Informationen und Zusammenhänge zwischen Projekten oder Themen schnell abrufbar und leicht verständlich sind.

Tim-Lukas Maeke

Title of Thesis

Design and prototypical implementation of a research management tool for professors

Keywords

Research management, Web application

Abstract

This Bachelor's thesis deals with the prototypical implementation of a research management tool to improve the management of professors. It is concerned with bundling the information of research topics and research projects, which is still very distributed, and with preparing it in such a way that information and connections between projects or topics can be quickly retrieved and easily understood.

Inhaltsverzeichnis

Abbildungsverzeichnis	v
Tabellenverzeichnis	vi
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	2
1.3 Gliederung	2
2 Anforderungsanalyse	4
2.1 Zielgruppen	4
2.1.1 Professor	4
2.1.2 Mitarbeiter	5
2.1.3 externe Personen	5
2.1.4 Studierende	6
2.1.5 Entwickler	7
2.2 Anforderungen	7
2.2.1 Funktionale Anforderungen	8
2.2.2 Nichtfunktionale Anforderungen	11
2.2.3 Zusammenfassung	12
2.3 Technologieentscheidungen	13
2.3.1 Auswahl der Werkzeuge und der Persistenzlösung	13
2.3.2 Marktanalyse	14
3 Analyse des Ist-Zustands	16
3.1 Listenverwaltung	16
3.2 Programme zur Bewahrung der Übersicht	16
3.3 Literaturverzeichnisse	17
3.4 Kommunikationsmittel	17

3.5	Terminverwaltung	18
3.6	Abschlussarbeiten	18
3.7	Konferenzen	18
3.8	Publikationen	19
4	Umsetzung	20
4.1	Aufbau der Applikation	21
4.1.1	Themen und Projekten	21
4.1.2	Die öffentlichen Ansichten	24
4.1.3	Aufbau aus Sicht des Benutzers	25
4.1.4	Hinzufügen, Updaten, Verschieben	25
4.2	Struktur der Datenbank	26
4.3	Betrieb der Webseite	27
5	Realisierung	28
5.1	Erfüllte und nicht erfüllte funktionale Anforderungen	28
5.2	Erfüllte und nicht erfüllte nichtfunktionale Anforderungen	31
6	Erweiterungsmöglichkeiten	32
6.1	Verbesserungsmöglichkeiten	32
6.2	Nicht erfüllte Anforderungen	32
6.3	Ideen und Wünsche der Professoren	33
6.4	Weitere Erweiterungsmöglichkeiten	34
7	Fazit	35
A	Anhang	38
A.1	Aufzeichnung Interviews mit den Professoren	38
A.2	Screenshots der Applikation	42
	Selbstständigkeitserklärung	45

Abbildungsverzeichnis

2.1	Fachliches Datenmodell	12
4.1	Komponentendiagramm	20
4.2	Verschachtelung von Themen und Projekten	22
4.3	Screenshot von der Projektseite mit geöffnetem Reiter 'Vorschläge für Abschlussarbeiten'	24
4.4	Screenshot des Dialogfensters beim Erstellen eines neuen Kontakts in einem Projekt oder Thema.	26
A.1	Erster schneller Entwurf der Verknüpfungen zwischen Themen und Projekten während eines Interviews mit einem Professor	41
A.2	Screenshot der Homepage und des Sidenavs des Programms	42
A.3	Screenshot vom offenen Sidenav mit bestehenden Projekten, Themen . . .	43
A.4	Screenshot der öffentlichen Seite von Ideen für Abschlussarbeiten	44

Tabellenverzeichnis

2.1	Persona eines Professors	5
2.2	Persona eines Mitarbeiters	5
2.3	Persona einer externen Person	6
2.4	Persona eines Studierenden	6
2.5	Persona eines Entwicklers	7

1 Einleitung

Die Arbeit von Professoren¹ ist vielseitig, sie zu organisieren und den Überblick zu behalten jedoch schwierig. Die Aufgabenschwerpunkte der Professoren liegen vor allem in der Lehre und der Forschung. Dazu kommt noch das Beantragen von Fördergeldern, der Datenaustausch mit Studierenden, wissenschaftlichen Mitarbeitern oder anderen Professoren. Die Organisation dieser Bereiche nimmt viel Zeit in Anspruch, die ggf. an anderer Stelle fehlt. Vor allem wenn die Unterlagen verstreut auf dem Computer, der Cloud oder auf dem Schreibtisch liegen. Die Organisation der Arbeit von Professoren geschieht in der Regel bisher mit einer Vielzahl von Programmen, die alle für einen bestimmten Teil der Aufgaben sehr gut geeignet sind, sie allerdings nicht zusammenfassen. Mit dieser Arbeit soll ein Prototyp für ein Forschungsmanagement Tool entworfen und erstellt werden, welches einen Überblick darüber gibt, wie die Forschung vernetzt und die Aufgaben verteilt sind. Die Anwendung soll eine Übersicht von allen Teilen der Forschung geben und zeigen wie Forschungsthemen oder Forschungsprojekte miteinander verknüpft sind.

1.1 Motivation

Der Ursprung dieser Arbeit war der Wunsch von mehreren Professoren der Hochschule für Angewandte Wissenschaften, eine Applikation zu erhalten, welches ihnen das Management ihrer Forschung vereinfacht und zusammenfasst. Die Professoren verwalten einzelne Komponenten der Forschung, wie Literaturverzeichnisse, Publikationsverzeichnisse und Abschlussarbeiten, in unterschiedlichen Programmen. Um diese Verzeichnisse mit anderen Teilen zu können werden meist einfach die URL zu den Programmen geteilt, sodass andere Mitarbeiter, Professoren oder Studenten diese einsehen können. Diese Methode ist allerdings sehr zeitaufwändig, da die Links zu den Programmen oft nur in den E-Mails

¹Aus Gründen der besseren Lesbarkeit wird in dieser Arbeit ausschließlich die männliche Form verwendet. Es wird darauf hingewiesen, dass diese Form sich auf Personen beiderlei Geschlechts bezieht.

oder Chatverläufen gespeichert sind. Es stellte sich heraus, dass es kein Programm gibt, welches alle Komponenten der Forschung zusammenfasst und übersichtlicher darstellt.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist es, eine prototypische Applikation zum Forschungsmanagement zu erstellen, welches die einzelnen Komponenten des Managements zusammenführt. Einige Komponenten, wie Termine oder Kontakte, sollen im Tool selbst erstellt werden. Bei anderen Komponenten, wie Abschlussarbeiten, sollen nur die wichtigsten Daten erstellt und die Abschlussarbeiten selbst verlinkt werden. Ähnlich wie bei Abschlussarbeiten sollen auch Kommunikationsmittel, wie Gruppenchats eines Projekts, lediglich verlinkt werden.

1.3 Gliederung

Die Arbeit ist in sieben Kapitel gegliedert, diese sind folgendermaßen aufgebaut: Kapitel 1, Einleitung, führt in das Thema ein und gibt einen kurzen Überblick über die Thematik.

Kapitel 2, Anforderungsanalyse, am Anfang des Kapitels werden die Zielgruppen anhand von Personas aufgezeigt. Im Anschluss werden die Anforderungen an die neue Anwendung beschrieben. Danach wird darauf eingegangen, welche Technologien für die Entwicklung der Anwendung genutzt werden. Zum Schluss wird eine Marktanalyse durchgeführt.

Kapitel 3, Analyse des Ist-Zustands, beschreibt welche Programme bis jetzt für das Forschungsmanagement von den Professoren genutzt werden und vergleicht diese mit den Anforderungen an die prototypische Applikation.

Kapitel 4, Umsetzung, beschreibt den Aufbau der Applikation. Im Anschluss wird die Speicherstruktur der Persistenzlösung beschrieben. Zum Schluss wird darauf eingegangen, wie die Webseite betrieben werden soll.

Kapitel 5, Realisierung, zeigt welche Anforderungen aus Kapitel 2 erfolgreich umgesetzt wurden und welche noch nicht umgesetzt werden konnten.

Kapitel 6, Erweiterungsmöglichkeiten, geht zunächst darauf ein welche Möglichkeiten es gibt die erarbeitete Anwendung zu verbessern. Im Anschluss wird darauf eingegangen, welche Möglichkeiten es gibt, die Applikation zu erweitern.

1 Einleitung

Kapitel 7, Fazit, gibt eine Zusammenfassung des Inhalts dieser Arbeit wieder und bewertet ihre Ergebnisse.

2 Anforderungsanalyse

Dieses Kapitel befasst sich mit der Analyse der Anforderungen, die an die Applikation gestellt werden sowie mit den grundlegenden Entscheidungen welche Technologien genutzt werden sollen. Am Ende des Kapitels wird eine Marktanalyse durchgeführt, um zu prüfen, welche vergleichbaren Applikationen bereits bestehen. Im Folgenden werden die Anforderungen systematisch erarbeitet und dargestellt. Die erforderlichen Informationen zur Erfassung der Anforderungen wurden mit der bereits bekannten Zielgruppe, den Professoren, durch Befragungen in Erfahrung gebracht.

2.1 Zielgruppen

Im Rahmen dieser Arbeit gibt es fünf verschiedene Gruppen von Stakeholdern. Neben der bereits bekannten Gruppe der Professoren gibt es noch die Gruppen der Studenten, der wissenschaftlichen Mitarbeiter oder allgemein Mitarbeiter, der externen Personen und der Entwickler. Um diese Stakeholder repräsentativ darzustellen und sich in diese eindenken zu können, wird im Folgenden jeder Gruppe durch eine Persona (nach Johannes Bergsmann[4]) beschrieben.

2.1.1 Professor

Name	Jürgen Bruns
Repräsentant für:	Benutzergruppe der Professoren
Alter	51
Land	Deutschland
Soziales Leben	Viel Freunde die auch jünger sind als er. Liebt es mit seinem Hund und seiner Frau am Strand spazieren zu gehen.
Wünsche	Würde gerne irgendwann mal eine Weltreise machen.

IT Nutzung	Nutzt einen MacBook, Arbeitet viel mit Safari.
Arbeitsleben	Als Professor arbeitet er nun schon 10 Jahre.
Verantwortlich für:	Er ist Verantwortlich für die Forschung im Bereich effizientere Erstellung von Software
Erwartung an die Software:	Möchte die Software haben, damit er eine schnellere und detaillierte Übersicht über sein Forschungsgebiet erhält.

Tabelle 2.1: Persona eines Professors

2.1.2 Mitarbeiter

Name	Harald Bremer
Repräsentant für:	Benutzergruppe der Mitarbeiter
Alter	38
Land	Deutschland
Soziales Leben	Fährt gerne mit seinem besten Freund Fahrrad.
Wünsche	Er möchte irgendwann seine eigene Firma leiten.
IT Nutzung	Harald nutzt Ubuntu. Er präferiert Firefox nutzt aber auch noch oft Microsoft Edge.
Arbeitsleben	Er strebt eine Promotion an und forscht am menschlichen Gehirn.
Verantwortlich für:	Er ist in seinem Projekt verantwortlich dafür die Treffen mit allen am Projekt beteiligten Personen zu organisieren und zu leiten.
Erwartung an die Software:	Möchte möglichst allen Beteiligten so die neusten Termine seines Projekts mitteilen können und Bescheid bekommen, welche anderen Personen von dem Professor für sein Projekt hinzugefügt werden.

Tabelle 2.2: Persona eines Mitarbeiters

2.1.3 externe Personen

Name	Julia Jäger
Repräsentant für:	Benutzergruppe der externen Personen

Alter	33
Land	Deutschland
Soziales Leben	Spielt Tennis mit ihrer besten Freundin. Ist deprimiert weil ihr Freund sie verlassen hat und will sich am Wochenende den Frust in der Disco raus tanzen.
Wünsche	Möchte Karriere in ihrem Unternehmen machen indem sie die Innovationsabteilung übernimmt.
IT Nutzung	Arbeitet am liebsten mit Windows und nutzt meistens Edge zum surfen. Macht aber möglichst viel über ihr Tablet.
Arbeitsleben	Sie arbeitet für eine Firma, die viele Forschungsprojekte zusammen mit Universitäten und Hochschulen vorantreibt.
Verantwortlich für:	Sie ist verantwortlich für die Kommunikation mit Projektpartnern und darf in Projekten nicht den Überblick verlieren.
Erwartung an die Software:	Möchte die Software, damit sie nicht andauernd ihren anderen externen Projektpartnern für Informationen hinterher laufen muss.

Tabelle 2.3: Persona einer externen Person

2.1.4 Studierende

Name	Patrick Mueller
Repräsentant für:	Benutzergruppe der Studenten
Alter	21
Land	England
Soziales Leben	Kommt gerade aus der Schule und studiert jetzt in Deutschland. Hat hier allerdings noch nicht viele Kontakte.
Wünsche	Würde gerne Deutsch lernen und in Deutschland bleiben.
IT Nutzung	Er nutzt ein MacBook aber liebt es mit Chrome zu surfen.
Arbeitsleben	Studentische Hilfskraft an seiner Hochschule.
Verantwortlich für:	Bau einer App für ein Projekt seines Professors.
Erwartung an die Software:	Hofft dadurch besseren Überblick zu bekommen was in seinem Projekt gerade wichtig ist.

Tabelle 2.4: Persona eines Studierenden

2.1.5 Entwickler

Name	Torge Kaufmann
Repräsentant für:	Benutzergruppe der Entwickler
Alter	28
Land	Deutschland
Soziales Leben	Er grillt gerne mit Freunden und feiert mindestens einmal die Woche.
Wünsche	Er würde sich gerne ein Haus kaufen und 3 Kinder mit seiner Frau bekommen.
IT Nutzung	Er arbeitet primär mit Windows und surft gerne über Chrome.
Arbeitsleben	Als Entwickler arbeitet er jetzt schon seit 4 Jahren.
Verantwortlich für:	Er ist in seiner Firma dafür verantwortlich die Frontends für Programme zu schreiben.
Erwartung an die Software:	Er möchte sie möglichst schnell schreiben und sie sollte möglichst wenig Ressourcen verbrauchen.

Tabelle 2.5: Persona eines Entwicklers

2.2 Anforderungen

In diesem Abschnitt geht es darum die Anforderungen für die neue Applikation zu beschreiben. Die Anforderungen werden aus Informationen gebildet. Diese Informationen stammen aus Befragungen die mit einzelnen Professoren der Hochschule für Angewandte Wissenschaften durchgeführt wurden. Anhang A.1 zeigt die vorher festgelegten Fragen, die den Professoren gestellt wurden. Außerdem zeigt es die gegebenen Antworten der Professoren stichwortartig auf.

Die erste Ausbaustufe der Applikation soll lediglich von Professoren und Mitarbeitern der Hochschule für Angewandte Wissenschaften verwendet werden. Aus diesem Grund wurden bis jetzt nur Anforderungen von Professoren, Mitarbeitern und sich daraus ableitende Anforderungen für andere Stakeholder aufgenommen.

Die Anforderungen werden in zwei unterschiedliche Arten unterteilt. Am Anfang werden die funktionalen Anforderungen beschrieben, im Anschluss die nichtfunktionalen Anforderungen.

2.2.1 Funktionale Anforderungen

Dieser Abschnitt beschreibt die funktionalen Anforderungen, die an die neue zu entwickelnde Applikation gestellt werden. Eine funktionale Anforderung sollte immer einen direkten Nutzen für die Software haben. Sie sollte die Funktionen eines kleinen Teil des Systems in natürlicher Sprache beschreiben, damit sowohl Anwender als auch Entwickler oder Manager verstehen, was diese Anforderung dem System für einen Nutzen bringt [13]. Die in Abschnitt 2.1 beschriebenen Personengruppen werden für die Gestaltung der Anforderungen verwendet.

A1: Als Jürgen Bruns möchte ich Themen strukturieren können, weil Themenideen aus E-Mails oder Diskussionen im Chaos verloren gehen.

Das Erfassen der Themen muss möglichst zügig gehen, bei hunderten von E-Mails kann leicht eine Idee oder ein Thema verloren gehen. Da Professoren nicht so viel Zeit haben, sollte es möglichst schnell gehen, sodass man eine Idee auch mal zwischen zwei Terminen, wenn die Zeit knapp ist, hinzufügen kann.

A2: Als Jürgen Bruns möchte ich Projekte übersichtlich und strukturiert speichern können, da ich sonst, wenn es viele Projekte auf einmal sind (und noch mehr Projekte dazu kommen), den Überblick über die wichtigen Dinge verliere.

Da Professoren oft in mehreren Projekten gleichzeitig involviert sind, muss es möglich sein auf einen Blick alle Details eines Projekts zu sehen. Außerdem sollte man neue Projekte möglichst zügig hinzufügen können, sodass man auch während oder zwischen zwei Besprechungen ein Projekt anlegen kann. Aktuelle Projekte, die häufig bearbeitet werden, sollen direkt sichtbar sein.

A3: Als Harald Bremer möchte ich zu meinen Projekten oder Themen andere Professoren, Mitarbeiter oder Studenten hinzufügen können.

Ein Projekt wird nie von nur einem Mitarbeiter oder Professor durchgeführt. Meist ist es ein Team von Personen, die an einem Projekt arbeiten. Diese sollten ebenfalls alle auf Informationen zu dem Projekt zugreifen können, damit niemand von Büro zu Büro laufen muss, um jedem einen Fortschritt oder eine neue Idee mitteilen zu müssen. Außerdem spart es Zeit und Informationen gehen nicht verloren, wenn sie beispielsweise per E-Mail geschickt werden.

A4: Als Jürgen Bruns möchte ich alles was mit Abschlussarbeiten zu tun hat speichern, veröffentlichen und zu Themen und Projekten zuordnen können, da ich sehr viele Studenten habe, die gern bei mir ihre Abschlussarbeit schreiben wollen.

Der Professor unterscheidet in drei verschiedene Arten von Abschlussarbeiten. Die erste Art ist eine Idee oder ein Vorschlag für eine Abschlussarbeit. Als zweites gibt es Abschlussarbeiten, die aktuell laufen und als letztes gibt es fertige Abschlussarbeiten. Diese sollten möglichst unterscheidbar gespeichert werden, sodass bei der Übersicht nicht viel Zeit gebraucht wird diese zu unterscheiden. Die Abschlussarbeiten sollten auch möglichst leicht einem oder mehreren Themen oder Projekten zugeordnet werden. Das hat den Vorteil, dass man genau sehen kann, welche Arbeiten grade in dem Thema oder Projekt geschrieben werden oder wurden, oder welche Ideen es für Arbeiten gibt. Außerdem sollte es eine Möglichkeit geben die Abschlussarbeiten auf der Webseite des Professors darzustellen.

A5: Als Harald Bremer möchte ich Termine erfassen können, um sie nicht zu vergessen. Außerdem möchte ich die Termine, Projekte oder Themen zuordnen können, damit ich weiß, wann ich den nächsten Termin für das Projekt oder Thema habe. Außerdem möchte ich einen Kalender haben, in dem ich alle Termine auf einen Blick sehen kann.

Das Erfassen von Terminen soll zügig gehen und auch von einem Handy aus möglich sein, damit man auch auf dem Flur oder beim Kaffee einen Termin eintragen kann. Damit man weiß, welcher Termine zu welchem Projekt oder Thema gehört, sollte es möglich sein, diese zuzuordnen, da man bei zehn Terminen an einem Tag sonst leicht den Überblick darüber verliert, welcher Termin für welches Projekt ist. Damit alle Termine auf einen Blick zu sehen sind, soll es möglichst auf der Startseite einen zusammengefassten Kalender mit allen Terminen geben.

A6: Als Jürgen Bruns möchte ich Konferenzen suchen und erfassen können, die mich interessieren oder zu meinem Projekt oder Thema passen. Außerdem möchte ich die Konferenzen zu Projekten oder Themen zuordnen können, damit ich weiß zu welchem Thema oder welchem Projekt die Konferenz passt.

Das Suchen von Konferenzen erstreckt sich öfters über mehrere Webseiten, was zeitlichen Aufwand bedeutet, wenn auf jeder Seite gesucht werden muss. Der Professor hätte gerne

eine zusammengefasste Suche, damit er sich Zeit spart und nicht auf jeder Webseite auf teilweise die gleichen Konferenzen trifft. Außerdem sollen Konferenzen zu Themen oder Projekten zugeordnet werden können, damit auch andere Nutzer aus einem Thema oder Projekt sehen können, welche Konferenz für ein Thema oder Projekt interessant wäre.

A7: Als Harald Bremer möchte ich Verlinkungen zu Publikationen erstellen können, die ich interessant finde. Außerdem möchte ich die Publikationen Themen oder Projekten zuordnen, damit auch meine Kollegen diese sehen und lesen können.

Publikationen zu verlinken hilft den Mitarbeitern und Professoren dabei diese wieder zu finden. Bei mehreren Projekten und Themen gleichzeitig kommt meist eine große Menge an Publikationen zusammen. Damit der Mitarbeiter nicht den Überblick verliert, sollte es möglich sein, Konferenzen zu Speichern und mit Projekten oder Themen zu verlinken.

A8: Als Jürgen Bruns möchte ich Links speichern können, damit ich einfacher auf sie zugreifen kann.

Durch die Arbeit mit vielen unterschiedlichen Personen verliert man z.B. leicht den Überblick welcher Gruppenchat, von welchem Anbieter, für welches Projekt oder Thema genutzt wird. Deshalb sollte es möglich sein einen Link mit einer Beschreibung zu speichern und diesen zu einem Projekt oder Thema zuordnen zu können.

A9: Als Harald Bremer möchte ich meine Kontakte speichern und diese zu einem Projekt oder Thema zuordnen können, damit ich weiß, welche Kontakte welches Projekt oder Thema bearbeiten.

Professoren oder Mitarbeiter haben durch viele Projekte meist auch viele Kontakte. Um diese zu ordnen und auch noch nach ein paar Jahren zu wissen, was für ein Projekt man mit wem gemacht hat, sollte es möglich sein, Kontakte zu speichern und mit Projekten oder auch Themen zu verknüpfen.

A10: Als Jürgen Bruns möchte ich Themen oder Projekte verlinken können, sodass ich weiß, welches Projekt zu welchem Thema gehört, oder welches Thema übergeordnet zu einem anderen steht.

Durch das Arbeiten in mehreren Themengebieten oder Projekten kommt es manchmal dazu, dass man nicht mehr genau weiß, welche Projekte zu welchen Themen gehören.

Dadurch sollte es möglich sein, Themen und Projekte mit anderen zu verlinken, sodass es sichtbar wird, wie Themen und Projekte miteinander in Beziehung stehen. Außerdem sollte es eine Gesamtübersicht über alle Zusammenhänge zwischen Themen und Projekten geben.

A11: Als Jürgen Bruns möchte ich, dass ich möglichst wenige Felder redundant schreiben muss, ich würde sie lieber auswählen können.

Viele verschiedene Objekte passen meist zu mehreren Projekten oder Themen. Damit nicht jedes Mal ein neues Objekt angelegt werden muss, sollte es möglich sein aus einer Liste von schon angelegten Objekten auszuwählen. Das würde viel Zeit sparen und auch zwischen oder während eines Meetings funktionieren.

2.2.2 Nichtfunktionale Anforderungen

In diesem Abschnitt geht es um die Erfassung der nichtfunktionalen Anforderungen. Eine nichtfunktionale Anforderung hat für den Benutzer der Software auf den ersten Blick keinen direkten Nutzen, vielmehr geht es bei diesen Anforderungen um den 'versteckten' Nutzen. Nichtfunktionale Anforderungen beziehen sich meist auf die Eigenschaften des gesamten Systems, wie beispielsweise die Zuverlässigkeit, Antwortzeiten oder auch die Verfügbarkeit [13].

N1 Als Jürgen Bruns möchte ich nicht länger als drei Sekunden warten müssen bis Ergebnisse erscheinen.

Der Professor will nicht zu lange warten müssen, um beispielsweise ein Projekt mit sehr vielen Daten laden zu können. Es soll darauf geachtet werden, dass diese Zeit unter fünf Sekunden bleibt.

N2 Als Harald Bremer will ich, dass die Anwendung nicht öfter als eine Stunde im Monat nicht erreichbar ist.

Es soll darauf geachtet werden, dass die Anwendung möglichst in 99,5% der Fälle erreichbar ist, damit wie in Anforderung A5 erwähnt, man auch beim Kaffee immer einen neuen Termin eintragen kann.

2.2.3 Zusammenfassung

Aus den in Abschnitt 2.2.1 und 2.2.2 erläuterten Anforderungen an die Applikation ergeben sich die in Abbildung 2.1 aufgezeigten Zusammenhänge zwischen den Anforderungen.

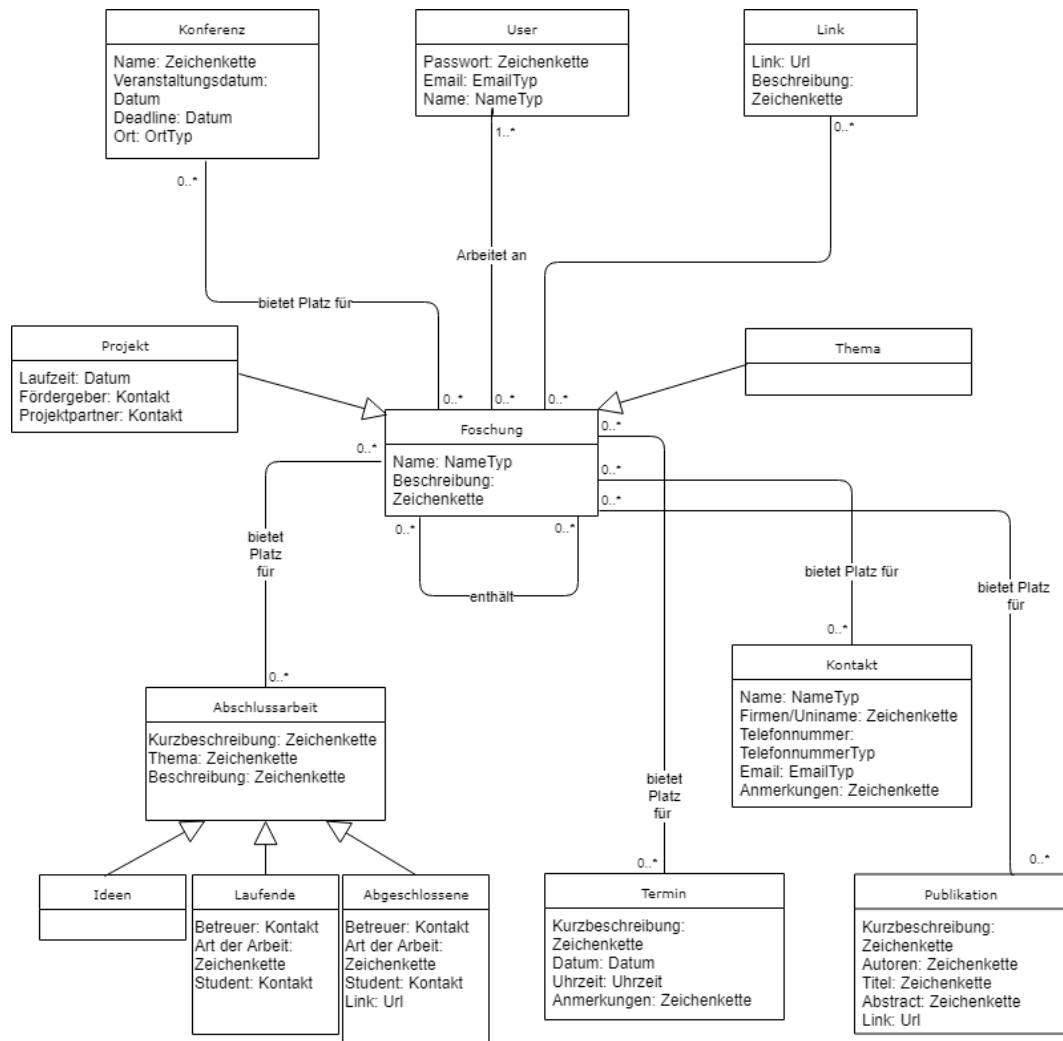


Abbildung 2.1: Fachliches Datenmodell

Man kann erkennen, dass Themen und Projekte sehr ähnliche Anforderungen und Zusammenhänge haben, was dazu veranlasst hat, diese unter der Entität 'Forschung' zusammenzufassen. Aus Anforderung A4 geht hervor, dass Abschlussarbeiten in drei ver-

schiedene Arten aufgeteilt werden, aus diesem Grund wurden auch diese zu der Entität Abschlussarbeit zusammengefasst.

2.3 Technologieentscheidungen

In diesem Abschnitt geht es um die grundlegenden Entscheidungen, die für diese Arbeit getroffen werden mussten.

2.3.1 Auswahl der Werkzeuge und der Persistenzlösung

Im Folgenden wird die Auswahl der Werkzeuge sowie der Persistenzlösung für die Applikation erläutert.

Auswahl der Werkzeuge

Nachdem die Anforderungen aufgenommen wurden, kann nun festgelegt werden, mit welchen Technologien die Applikation realisiert werden soll. Für ein Forschungsmanagement Tool stehen viele Möglichkeiten zur Verfügung.

Es wurde entschieden eine Webanwendung zu entwickeln, da diese sowohl auf mobilen Endgeräten als auch im normalen Webgebrauch nutzbar ist und so allen Nutzer der Zielgruppen später auch von Nutzen sein kann. Bei der Entscheidung, welches Framework genutzt werden soll, fiel die Wahl auf Angular. Angular ist eine von Google unterstütztes Web Applikation Framework [3]. Es dient zum Entwickeln von Client-Basierten Applikationen, geschrieben in HTML, CSS und TypeScript [1]. Eine Alternative zu Angular wäre React. React ist eine von Facebook unterstützte JavaScript Bibliothek zum entwickeln von Benutzeroberflächen [12]. Unterschied zwischen diesen Programmen liegen im Data-Binding, DOM (Document Object Model) und den zu benutzenden Sprachen. Angular nutzt Two-way Data Binding, React lediglich One-way, was für Angular spricht. Für React hingegen spricht, dass es ein virtuellen DOM benutzt, Angular nicht. Ein weiterer Unterschied liegt darin, welche Programmiersprachen genutzt werden. Wie oben schon erwähnt wird in Angular mit HTML, CSS und vor allem TypeScript gearbeitet. React hingegen nutzt JavaScript [2]. Ausschlaggebend für die Entscheidung Angular zu nutzen war jedoch, dass Angular ein Framework ist und somit einen Rahmen vor gibt, was bei React nicht der Fall ist.

Auswahl der Persistenzlösung

Neben den Werkzeugen zur Erstellung der Applikation muss festgestellt werden, was für Daten anfallen und wie diese zu speichern sind. Dadurch, dass später eine Vielzahl von Personen die Applikation benutzen sollen und jede Person eigene Daten speichern und abrufen möchte, wird davon ausgegangen, dass die zum Speichern vorgesehenen Daten zumindest teilweise personenbezogen sind. In diesem Prototypen werden Daten in nutzerbezogene Daten und in allgemeine Daten unterteilt. Nutzerbezogene Daten sind beispielsweise UserId, Username, Passwort, Email. Aber auch die anfallenden Daten in Projekten oder Themen sind einem Nutzer zugeordnet. Die allgemeinen Daten sind Daten, die für alle Nutzer einsehbar sind. Für die Speicherung der Daten gibt es verschiedenste Möglichkeiten. Es wurde entschieden eine dokumentenbasierte Datenbank zu nutzen. Die Auswahl an Dokument Datenbanken ist groß. Die wohl bekanntesten Dokument Datenbanken sind CouchDB, MongoDB, sowie Google Firebase, um nur ein paar zu nennen. Firebase hat die Vorteile, dass es eine Plattform ist und nicht nur eine Datenbank. Firebase bietet neben einem Software Development Kit (SDK) und der Infrastruktur zum Entwickeln von Webanwendungen auch einen Backend Authentifizierungsservice [7]. Firebase Authentication bietet die Möglichkeit, den kompletten Authentifizierungsvorgang sowie das Session handling zu übernehmen. Dadurch, dass Firebase sowohl Authentifizierung als auch eine Datenbank zusammen als REST API anbietet, ist die Entscheidung auf Firebase als Persistenzlösung gefallen. Nachdem die grundlegende Entscheidung gefallen ist, musste nun noch entschieden werden, welche Datenbank von Firestore genutzt werden soll. Firebase bietet zwei unterschiedliche Datenbanken an. Zum einen die Firebase Realtime Database, zum anderen den Firebase Cloud Firestore. Der Unterschied liegt darin, dass die Realtime Datenbank Daten als einen großen JSON Baum speichert, was es einfach macht einfache Daten zu speichern. Firestore hingegen ist in Collections und Dokumenten organisiert. Die Daten werden in Dokumenten gespeichert, diese sind ähnlich aufgebaut wie JSON-Dateien. Vorteil von Firestore ist es, dass sich komplexere Daten durch die Speicherstruktur leichter skalieren lassen [6]. Resultierend aus den Vorteilen von Firestore wurde entschieden, Firestore der Realtime Datenbank vorzuziehen.

2.3.2 Marktanalyse

Nachdem die Anforderungen erfasst wurden, wurde geprüft, ob vergleichbare Applikationen für das Forschungsmanagement von Professoren oder Projektgruppen existieren.

Ein vergleichbares System ist das der Hochschul Informations System eG (HIS). Das System der HIS bezieht sich allerdings auf eine komplette Forschungseinrichtung, wie eine Universität oder eine Hochschule. Sie stellen Verwaltungssysteme, wie HISinOne oder Hochul-ERP [5], zur Verfügung, die für alle Stakeholder einer Einrichtung gedacht sind. Der große Unterschied zu der zu entwickelnden Applikation liegt darin, dass diese Systeme nicht auf die Forschung eines einzelnen Professors oder einer kleinen Gruppe ausgelegt sind, sondern auf die Lehre und Verwaltung der gesamten Einrichtung. Die Verwaltungssysteme organisieren primär Kurse, Vorlesungen, Noten oder die allgemeine Verwaltung der Einrichtung. Bei einigen Systemen ist es auch möglich, Unterlagen zu verteilen. Ein weiteres System ist Moodle. Moodle ist, im Gegensatz zu den Produkten von der HIS, eine Open Source Software. Sie dient dazu, Kurse zu entwickeln, durchzuführen und zu verwalten [14]. Moodle bietet eine Vielzahl von Funktionalitäten an, darunter eine 'Alles-in-einem-Kalender' und eine Datenverwaltung Funktion an [11]. Im Vergleich zu der entwickelnden Applikation ist Moodle allerdings eine Online Lernplattform und Managementsystem [11]. Moodle ist primär für die Lehre entwickelt und weniger für die Forschung. Recht nah an die Erfüllung der Anforderungen kommt Asana. Asana ist eine Plattform zur Verwaltung von Teams, Aufgaben, Ziele und Projekte. Es bietet ebenfalls einen Kalender und eine Diskussionsforum, sowie einen Platz zum Brainstorming. Allerdings gibt es keinen Platz zum Verwalten von Listen wie z.B. von Abschlussarbeiten. Noch dazu sind viele Features nur mit der Premium Version nutzbar. Der größte Nachteil liegt allerdings darin, dass keine Themengebiete gespeichert werden können und vor allem, dass es keinerlei Möglichkeiten gibt, Verknüpfungen unter Projekten oder Themen herzustellen.

3 Analyse des Ist-Zustands

In den Gesprächen mit den Professoren wurde gefragt, welche Programme oder Techniken genutzt werden, um den Überblick über einzelne Komponenten der Forschung nicht zu verlieren. In diesem Kapitel geht es darum, die aktuellen Programme und Techniken zu analysieren und zu prüfen, wie diese in der neuen Applikation genutzt oder eingebunden werden können.

Zur Verwaltung von Abschlussarbeitslisten, Literaturverzeichnissen, Papern, Konferenzen und vielem mehr, gibt es eine Vielzahl von Programmen, die genutzt werden können und auch werden. Eines haben alle Programme gemeinsam. Sie sind alle nur für einen Teilbereich der gesamten Verwaltung bestimmt.

3.1 Listenverwaltung

Aus der Befragung (siehe Anhang A.1 Frage zwei) der Professoren ging hervor, dass zum Verwalten von Listen mit Abschlussarbeiten, Themen, Projekten, Konferenzen usw. meist Excel verwendet wird. Eine einfache Lösung für Listen ohne Verknüpfung zu anderen Listen. Wenn Listen allerdings eine Verknüpfung mit anderen Listen haben, wird dies meist sehr unübersichtlich, umständlich und sehr zeitaufwendig in der Wartung. Im Vergleich zu Excel sollte in der neuen Applikation daher darauf geachtet werden, dass die Übersichtlichkeit sowie Wartbarkeit von Listen möglichst einfach gehalten wird und dass Verknüpfungen automatisch erstellt werden.

3.2 Programme zur Bewahrung der Übersicht

Um die Übersicht über Projekte, Themen, Brainstorming, etc. nicht zu verlieren, ergab sich aus den Gesprächen, dass die Professoren meist eine Mindmap nutzen. Diese gibt sehr schnell einen Überblick über den aktuellen Stand der Dinge oder fasst Dinge sehr gut

zusammen. Diese Art zur Schaffung von Übersichtlichkeit hat sich bewährt, da sie schnell durchzuführen ist. Soweit es möglich ist, sollte man solche Übersicht in die Applikation integrieren, allerdings nur bis zu einem bestimmten Punkt. Die Realisierung der Abhängigkeiten beispielsweise zwischen Projekten und Themen sollte selbst integriert werden. Dies wäre zum Beispiel über eine Graphdatenbank möglich, als Datenspeicher und einer Realisierung der Daten in der Webanwendung. Angular unterstützt solche Realisierungen. Mindmapping zum Brainstormen, etc. sollten über ein externes Tool generiert und bearbeitet werden. Tools wie Mindejet sind dafür ausgelegt. Im Anschluss kann dann ein Link auf diese Tools hinterlegt werden, was den Speicherbedarf der Applikation klein hält.

3.3 Literaturverzeichnisse

Literaturverzeichnisse werden von den Professoren oft in Mendeley geführt. Mendeley ist ein Programm zur Verwaltung und Organisation von Literatursammlungen. Es ist möglich sich ein Forschungsnetzwerk aufzubauen, was dem Austausch von Literatur dient. Um nicht die Übersichtlichkeit des neuen Systems, durch zu viele Daten, einzuschränken, wurde entschieden die Literaturverzeichnisse nicht selbst aufzubauen, sondern lediglich über einen Link zu Mendeley, oder anderen vergleichbaren Programmen, zu verknüpfen. Die Verzeichnisse werden dann über diese Programme geteilt, sodass es keinen weiteren Aufwand in der neuen Applikation bedarf.

3.4 Kommunikationsmittel

Beim Thema Kommunikationsmittel gibt es viele verschiedenen Wege. Die Professoren nutzen bis zu sieben verschiedene Programme zum Kommunizieren. Dies zu vereinheitlichen ist nahezu unmöglich. Deshalb wurde sich hier darauf verständigt, lediglich Links zu den einzelnen Kommunikationsplattform zu speichern, was auch wieder der Übersicht dient und verhindert, dass die Applikation sich in die ohnehin schon lange Reihe der Kommunikationsplattformen einreihet. Hierbei gilt es zu unterscheiden in Links zu einzelnen Personen, wie beispielsweise der SkypeID, oder Links zu Gruppen, wie beispielsweise zu einem Gruppenchat bei Slack. Beides sollte möglich sein.

3.5 Terminverwaltung

Termine werden aktuell primär in einem oder mehreren Google Kalendern abgespeichert. Um die Übersichtlichkeit zu wahren und projekt- oder themeninterne Termine im jeweiligen Kontext abspeichern zu können, wird ein Kalender für jedes Projekt bzw. Thema angestrebt. Hierbei soll ein Google Kalender importiert bzw. exportiert werden können.

3.6 Abschlussarbeiten

Aus den Interviews mit den Professoren stellte sich heraus, dass Abschlussarbeiten in mehre Kategorien aufgeteilt werden müssen. Abschlussarbeiten sind aufgeteilt in Ideen oder Vorschläge, laufende und fertige Abschlussarbeiten.

Ideen oder Vorschläge für Abschlussarbeiten werden meist in Excel Listen verwaltet und auf einer oder mehreren Webseiten veröffentlicht. Dabei kann die Übersicht schnell leiden, indem die Listen nicht mit den Webseiten übereinstimmen. Besser wäre es, wenn sich die Liste der Vorschläge direkt mit der Webseite synchronisiert, was den Arbeitsaufwand minimieren würde.

Informationen über laufende und fertige Abschlussarbeiten, wie welcher Student, was für eine Art der Arbeit usw., werden ähnlich wie bei den Vorschlägen auch primär in Excel Listen verwaltet. Ein neuer Aspekt ist die Speicherung des aktuellen Stands der Arbeit. Meist werden Entwürfe zwischen Professoren und Studenten per Mail verschickt. Dies kann dazu führen, dass nicht jede Partei auf dem neusten Stand der Arbeit ist. Um dies zu ändern, sollte in der neuen Applikation eine Liste gespeichert werden, wo der letzte Stand der Arbeit als PDF gespeichert ist. Alternativ kann auch ein Link zur Arbeit gespeichert werden, bei dem man diese einsehen kann.

3.7 Konferenzen

Um Konferenzen zu finden, nutzen die Professoren Seiten wie WikiCFP [15]. Auf WikiCFP werden in allen möglichen Kategorien Workshops und Konferenzen verzeichnet. Diese besitzen immer eine Bewerbungsdeadline. Vor dieser Deadline muss ein Paper oder Abstract eingereicht werden, damit man an der Konferenz teilnehmen kann. Damit man sehen kann, welche Konferenzen für ein jeweiliges Projekt oder Thema interessant sind, sollten Konferenzen im Kontext des Projekts oder Themas gespeichert werden können.

Außerdem sollte es eine Warnung oder eine Erinnerung geben, bevor die Deadlines verpasst werden. Aktuell werden Konferenzen eher weniger von den Professoren gespeichert. Deshalb sollte es möglichst wenig Aufwand bereiten, die Konferenzen im Kontext zu speichern.

3.8 Publikationen

Die Suche von Publikationen geschieht meist über Mendeley [10] oder Google Scholar [8]. Da die Suche an sich schon durch mehrere, eben erwähnte, Seiten abgedeckt ist, wird sich darauf konzentriert, gefundene Publikationen in irgendeiner Art zu speichern. Meist speichern die Professoren nur die Links zu Publikationen in einer Liste oder sie nutzen Mendeley oder ähnliches. Die Informationen werden jedoch meist nicht in einem Kontext gespeichert. Die neue Anwendung sollte dies ändern und Publikationen, die beispielsweise in ein Projekt Kontext hinein passen, speichern. Um keine Probleme mit der Speicherung oder der späteren 'Veröffentlichung' in der Applikation zu bekommen, sollte lediglich ein Link mit Beschreibung gespeichert werden.

4 Umsetzung

In diesem Kapitel wird die Umsetzung der von den Professoren gewünschten Applikation erläutert. Im Folgenden wird die Architektur erläutert, am Ende des Kapitels geht es um den Betrieb der Webseite. Abbildung 4.1 zeigt die sich aus Abbildung 2.1 ergebenden Komponenten und ihre Schnittstellen.

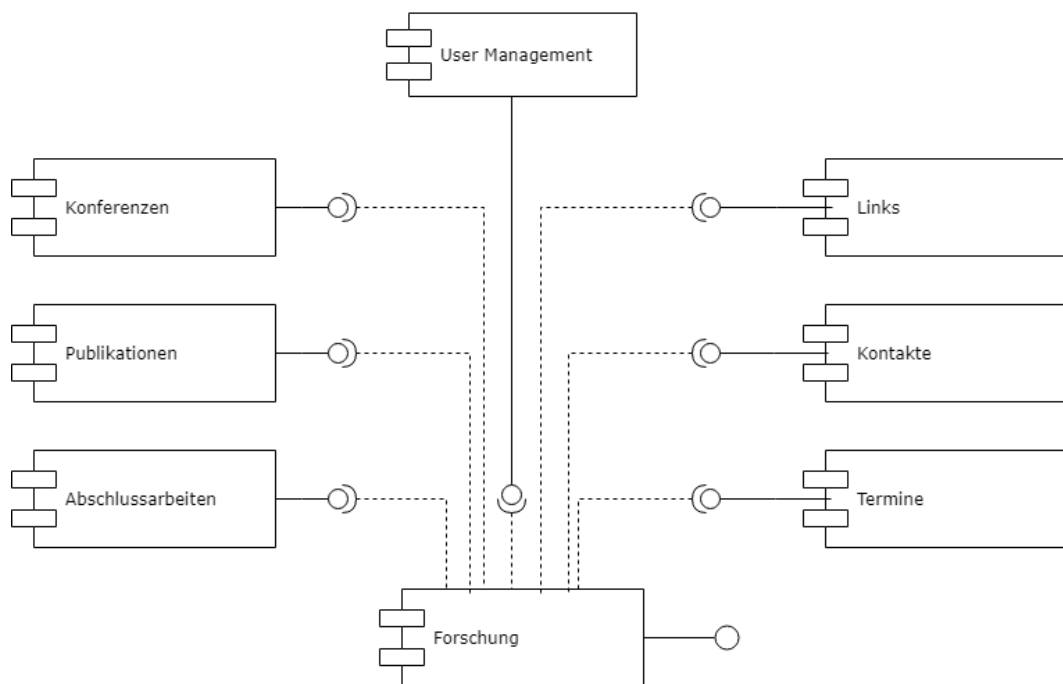


Abbildung 4.1: Komponentendiagramm

4.1 Aufbau der Applikation

Dieser Abschnitt beschäftigt sich mit dem Aufbau der Applikation und mit deren Besonderheiten. Die Applikation besteht aus mehreren Ansichten. Die wichtigsten Ansichten in dieser Arbeit sind die Ansichten auf Projekte, Themen und den Öffentlichen Teil. Im Folgenden werden die Gemeinsamkeiten, Unterschiede und Abhängigkeiten sowie der Aufbau der Ansichten beschrieben.

4.1.1 Themen und Projekten

In diesem Abschnitt geht es um den Aufbau, die Gemeinsamkeiten, Unterschiede und Abhängigkeiten zwischen Themen und Projekten.

Zusammenhang zwischen Themen und Projekten

Wie aus Anforderung A1, A2 und A3 hervor geht, wird Forschung meist in Forschungsthemen und Forschungsprojekten aufgeteilt¹. Projekte und Themen sind dabei oft miteinander verknüpft. Wie in Abbildung 4.2 (Siehe auch A.1, erste Skizze während eines Interviews mit einem Professor) zu sehen ist, können Themen verwandt zu anderen Themen sein. Äquivalent zu den Themen können auch Projekte verwandt zu anderen Projekten sein. Noch dazu können Projekte aber auch untergeordnet zu Themen sein.

¹Aus Gründen der besseren Lesbarkeit wird im folgenden nur noch von Themen und Projekten ausgegangen.

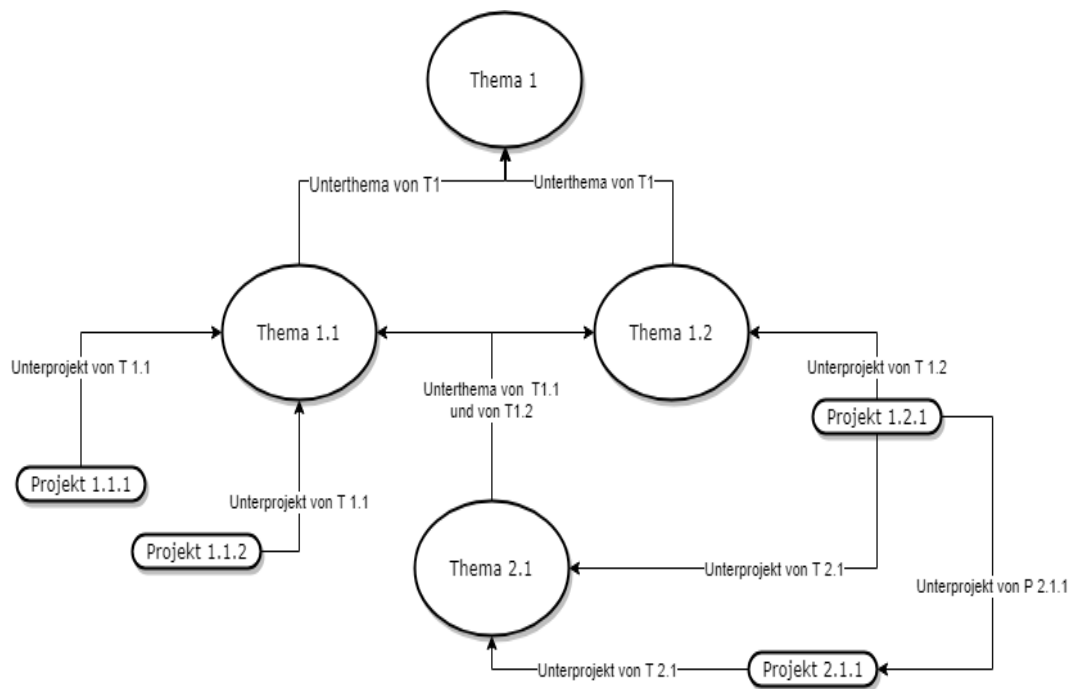


Abbildung 4.2: Verschachtelung von Themen und Projekten

Aufbau, Gemeinsamkeiten und Unterschiede

Themen und Projekte sind in der Applikation jeweils eigene Komponenten. Wie aus Abbildung 2.1 hervorgeht, soll in der Projektansicht angezeigt werden, was für Abschlussarbeiten gerade geschrieben werden, welche im Projektkontext schon geschrieben wurden und was für Vorschläge es gibt, worüber man noch schreiben kann. Außerdem werden die am Projekt beteiligten Mitarbeiter, Termine, Kontakte, Links, Publikationen, Konferenzen, Überprojekte, Unterprojekte und Projekt Zugehörigkeit angezeigt. Viele der gerade genannten Punkte sind auch in der Themenansicht wieder zu finden. Lediglich Über- und Unterprojekte sind nicht zu finden. Sie werden ersetzt durch Über- und Unterthemen. Ebenfalls gibt es keine Projekt Zugehörigkeit. Dies listet alle Themen auf, welche einen Bezug zu dem Projekt haben. Bei der Themenansicht werden im Gegensatz zur Projektansicht alle Projekte angezeigt, die zu einem Thema zugeordnet werden. Diese wurden durch Über- und Unterthemen und Projekte ersetzt.

Alle eben erwähnten Punkte, wie beispielsweise Publikationen, Termine oder Kontakte, werden in der Applikation als einzelne, aufklappbare Reiter angezeigt. Die Informationen sind dann tabellarisch dargestellt. Die Komponenten geben dabei nur ein Grundgerüst

vor, in das mit Hilfe einer Konfigurationsdatei die oben genannten Reiter mit den Tabellen automatisch generiert werden.

Der Aufbau der Seiten von Projekten und Themen ist bis auf einzelne Details, sowohl in der Ansicht als auch in den Komponenten, identisch. Beide Komponenten haben dieselben Beziehungen zu den restlichen Komponenten der Applikation. Deshalb wurden die Komponenten 'Themen' und 'Projekte' in Abbildung 4.1 auch unter der Komponente 'Forschung' zusammengefasst. Die Unterschiede zwischen den Komponenten als auch in der Ansicht liegen in den jeweiligen Details. Projekte haben eine Beschreibung, Fördergeber, Projektpartner und eine Laufzeit. Themen hingegen lediglich eine Beschreibung.

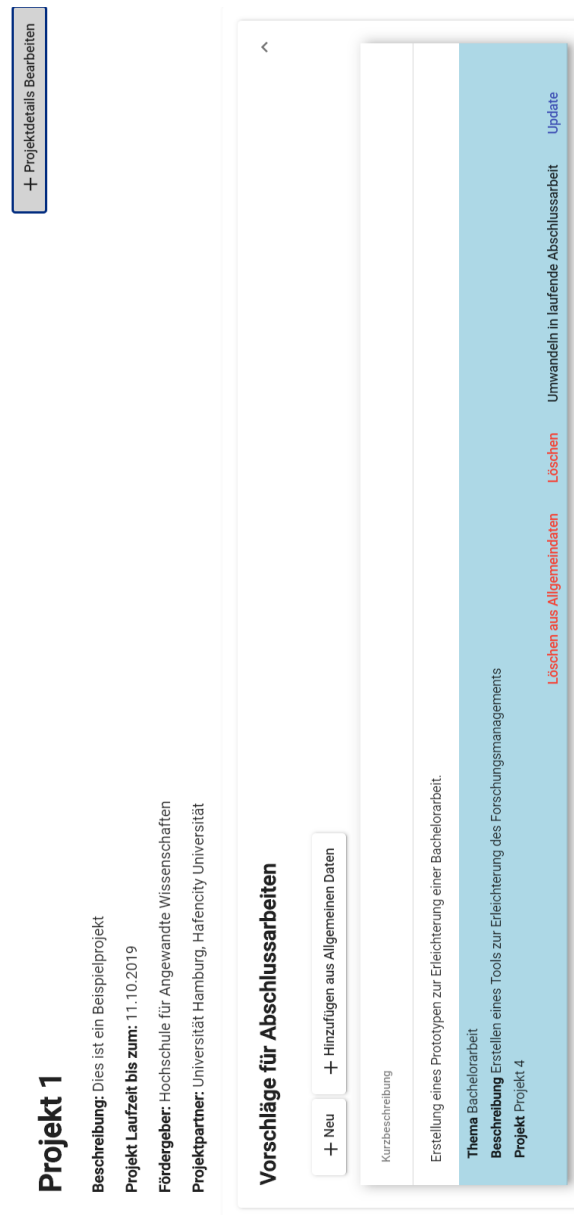


Abbildung 4.3: Screenshot von der Projektseite mit geöffnetem Reiter 'Vorschläge für Abschlussarbeiten'

4.1.2 Die öffentlichen Ansichten

Die Daten in der öffentlichen Ansicht sind für alle registrierten Nutzer der Applikation sichtbar. Anders als bei der Projekt- oder Themenansicht ist die öffentliche Ansicht auf

mehrere Seiten verteilt. Es gibt Seiten für Vorschläge für Abschlussarbeiten, für laufende und für fertige Abschlussarbeiten. Außerdem gibt es Seiten für Links, Termine, Kontakte, Publikationen und Konferenzen. Jede Seite besitzt nur den Inhalt für einen Punkt. Somit bekommt man beispielsweise einen Überblick über alle öffentlichen Kontakte, die gespeichert wurden. Die Daten, die in der öffentlichen Ansicht dargestellt werden, können einen Bezug zu einem Thema oder Projekt haben, müssen es jedoch nicht. Es wurde entschieden, dass Themen oder Projekte nicht öffentlich sind. In Abbildung A.4 werden die Allgemeinen Vorschläge für Abschlussarbeiten dargestellt.

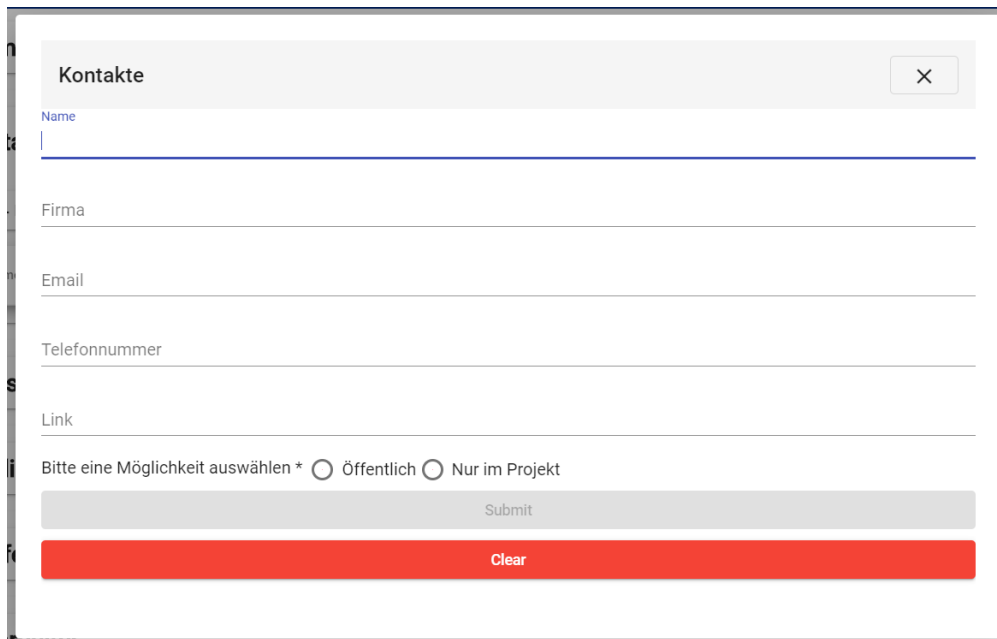
4.1.3 Aufbau aus Sicht des Benutzers

Die oben erwähnten drei Ansichten, Öffentlich, Projekte und Themen, bilden die grundlegenden Bestandteile des von den Benutzern einzusehenden Teils der Applikation. Um die Applikation nutzen zu können, wird ein Login vorausgesetzt. Vor einem möglichen Login muss ein neuer Benutzer sich registrieren und verifizieren durch eine Bestätigungsmail. Ist der Register- und Loginvorgang abgeschlossen wird man direkt auf die sogenannte Homeseite, also die Startseite der Anwendung geleitet. Diese wird aktuell für die Erstellung von neuen Projekten und Themen genutzt. Sobald man zur Homeseite geführt wird, wird auch die Navigationsleiste auf der linken Seite sichtbar. Wie auf Abbildung A.2 zu sehen ist, ist die Navigationsleiste dauerhaft geöffnet, sie wird nur geschlossen, wenn die Webanwendung auf ein bestimmtes Maß verkleinert wird. Auf mobilen Endgeräten ist die Navigationsleiste immer eingeklappt und kann bei Bedarf geöffnet werden.

Von der Homeseite aus kann man über die Navigation auf die drei Hauptteile der Anwendung gelangen. Abbildung A.3 zeigt, dass die Überpunkte aufklappbar sind und die nutzerbezogenen Projekte oder Themen darunter angezeigt werden.

4.1.4 Hinzufügen, Updaten, Verschieben

Das Hinzufügen und Updaten eines Datensatzes ist mit einem Dialogfenster möglich.



The screenshot shows a dialog window titled 'Kontakte' with a close button (X) in the top right corner. The form contains the following fields and elements:

- Name:** A text input field.
- Firma:** A text input field.
- Email:** A text input field.
- Telefonnummer:** A text input field.
- Link:** A text input field.
- Bitte eine Möglichkeit auswählen *:** Two radio buttons: Öffentlich and Nur im Projekt.
- Submit:** A grey button.
- Clear:** A red button.

Abbildung 4.4: Screenshot des Dialogfensters beim Erstellen eines neuen Kontakts in einem Projekt oder Thema.

Wenn man einen Datensatz in einem Thema oder Projekt hinzufügen oder updaten möchte, kann man diesen Datensatz öffentlich machen. Wenn man ihn öffentlich macht, wird er nach der Speicherung im Projekt und in der öffentlichen Ansicht sichtbar. Wird der Datensatz nur für das Projekt freigegeben, können nur die am Projekt beteiligten Nutzer den Datensatz im jeweiligen Projekt sehen.

4.2 Struktur der Datenbank

In diesem Abschnitt geht es um den Aufbau der Persistenzlösung. Wie in Absatz 2.3.1 bereits erläutert, wird zum Speichern der Daten Googles Cloud Firestore genutzt. Die Daten werden in Firestore in verschiedenen Collections gespeichert.

Jede Entität aus Abbildung 2.1 wird in einer eigenen Collection gespeichert. Themen und Projekte, sowie jede Art von Abschlussarbeiten haben eine eigene Collection. Die Collections 'Projekte' und 'Themen' haben wiederum Subcollections, die die zugeordneten Abschlussarbeiten, Termine, Kontakte, usw. enthalten.

Neben Projekten und Themen hat auch die Collection 'Nutzer' weitere Subcollections.

Nach dem Registrieren und Verifizieren des Nutzers wird dieser zu der Collection 'Nutzer' hinzugefügt. Sobald der Nutzer sein erstes Projekt oder Thema erstellt, wird eine Subcollection für den jeweiligen User erstellt, wo die Daten wie ProjektId oder UserId in einzelnen Dokumenten gespeichert werden.

4.3 Betrieb der Webseite

In diesem Abschnitt geht es um den Betrieb der Applikation. Neben der Entwicklung der Applikation und dem Speichern der Daten ist auch zu überlegen, wie die Webseite betrieben werden soll. Um eine Webseite betreiben zu können, wird eine Domain und ein Server benötigt. Da die Applikation vorerst nur für die HAW genutzt werden soll, macht es Sinn auf die dortige Infrastruktur zurückzugreifen. Die Informatik Compute Cloud (ICC) ist eine von der HAW zur Verfügung gestellte Container Cloud Umgebung, welche mit Gitlab und Kubernetes betrieben und von den Mitarbeitern sowie Studierenden des Departments Informatik genutzt werden kann. Für das Hosten der Applikation, unter der Subdomain "pat.informatik.haw-hamburg.de" wird ein Nginx Gateway benötigt, der von der Hochschule in der ICC zur Verfügung gestellt wird.

5 Realisierung

In diesem Kapitel wird beschrieben welche Anforderungen aus Kapitel 2 erfüllt werden konnten und welche noch nicht realisierbar waren. Darüber hinaus wird auf die wichtigsten Stellen der Implementierung eingegangen.

5.1 Erfüllte und nicht erfüllte funktionale Anforderungen

In diesem Abschnitt geht es darum, welche funktionalen Anforderungen aus 2.2.1 erfüllt werden konnten und welche nicht. Im Folgenden werden die Antworten aufgelistet und die aktuelle Situation besprochen.

A1: Als Jürgen Bruns möchte ich Themen strukturieren können, weil Themenideen aus E-Mails oder Diskussionen im Chaos verloren gehen.

Diese Anforderung ist erfüllt worden, Themen können über ein Template hinzugefügt werden. Dazu bedarf es lediglich einen Themennamen, einer Beschreibung und, falls mehrere Personen beteiligt sind, die Email Adressen der weiteren Nutzer.

A2: Als Jürgen Bruns möchte ich Projekte übersichtlich und strukturiert speichern können, da ich sonst, wenn es viele Projekte auf einmal sind (und noch mehr Projekte dazu kommen), den Überblick über die wichtigen Dinge verliere.

Auch diese Anforderung wurde erfüllt. Wie auch bei A1 gibt es ein Template zum hinzufügen neuer Projekte. Diese werden dann in einer Liste in der Navigationsleiste (siehe A.3) übersichtlich gespeichert.

A3: Als Harald Bremer möchte ich zu meinen Projekten oder Themen andere Professoren, Mitarbeiter oder Studenten hinzufügen können.

Auch diese Anforderung ist komplett erfüllt worden. Sobald sich eine Person registriert und verifiziert hat, kann diese zu Projekten oder Themen hinzugefügt werden.

A4: Als Jürgen Bruns möchte ich alles was mit Abschlussarbeiten zu tun hat speichern, veröffentlichen und zu Themen und Projekten zuordnen können, da ich sehr viele Studenten habe, die gern bei mir ihre Abschlussarbeit schreiben wollen.

Abschlussarbeiten in unterschiedliche Arten aufzuteilen wurde realisiert, indem jede Art eine eigene Tabelle bekommen hat und man Objekte zwischen den einzelnen Arten verschieben kann. So kann man nun eine Themenidee zu einer laufenden Abschlussarbeit und eine laufende in eine fertig Abschlussarbeit umwandeln. Auch können Abschlussarbeiten zu Themen oder Projekten zugeordnet werden.

Die Anforderung konnte nicht komplett umgesetzt werden, eine Darstellung auf der Webseite des Professors konnte noch nicht umgesetzt werden.

A5: Als Harald Bremer möchte ich Termine erfassen können, um sie nicht zu vergessen. Außerdem möchte ich die Termine, Projekte oder Themen zuordnen können, damit ich weiß, wann ich den nächsten Termin für das Projekt oder Thema habe. Außerdem möchte ich einen Kalender haben, in dem ich alle Termine auf einen Blick sehen kann.

Sowohl einen Termin erstellen als auch zu einem Thema oder Projekt zuordnen ist möglich. Nicht erfüllt werden konnte eine kalendarische Übersicht über alle Termine.

A6: Als Jürgen Bruns möchte ich Konferenzen suchen und erfassen können, die mich interessieren oder zu meinem Projekt oder Thema passen. Außerdem möchte ich die Konferenzen Projekten oder Themen zuordnen können, damit ich weiß zu welchem Thema die Konferenz passt.

Eine Suchfunktion für Konferenzen konnte noch nicht umgesetzt werden, allerdings ist es möglich eine Konferenz zu erstellen und diese Projekten und Themen zuzuordnen.

A7: Als Harald Bremer möchte ich Verlinkungen zu Publikationen erstellen können, die ich interessant finde. Außerdem möchte ich die Publikationen

Themen oder Projekten teilen damit auch meine Kollegen diese sehen und lesen können.

Diese Anforderung konnte komplett erfüllt werden. Es ist möglich Verlinkungen zu Publikationen mit dem oder den Autoren, dem Abstract und dem Titel zu speichern und mit Projekten oder Themen zu verknüpfen.

A8 Als Jürgen Bruns möchte ich Links speichern können, damit ich einfacher auf sie zugreifen kann.

Auch diese Anforderung konnte umgesetzt werden. Es ist möglich Links mit einer Beschreibung zu Speichern und Projekten oder Themen zuzuordnen.

A9: Als Harald Bremer möchte ich meine Kontakte speichern und diese zu einem Projekt oder Thema zuordnen können, damit ich weiß welche Kontakte welches Projekt oder Thema bearbeiten.

Auch Anforderung A9 konnte umgesetzt werden. Es ist möglich Kontakte zu speichern und mit Projekten oder Themen zu verlinken.

A10: Als Jürgen Bruns möchte ich Themen oder Projekte verlinken können, sodass ich weiß welches Projekt zu welchem Thema gehört, oder welches Thema übergeordnet zu einem anderen steht.

Themen und Projekte untereinander zu verlinken ist in der Applikation möglich. Eine Gesamtübersicht der Zusammenhänge zwischen Themen und Projekten wie in 4.1.1 erklärt, konnte noch nicht umgesetzt werden.

A11: Als Jürgen Bruns möchte ich, dass ich möglichst wenige Felder redundant schreiben muss, ich würde sie lieber auswählen können.

Die Anforderung wurde realisiert. In der Applikation ist es möglich Daten, die öffentlich zugänglich sind, auszuwählen.

5.2 Erfüllte und nicht erfüllte nichtfunktionale Anforderungen

In diesem Abschnitt geht es darum, welche nichtfunktionalen Anforderungen erfüllt werden konnten und welche nicht.

N1 Als Jürgen Bruns möchte ich nicht länger als drei Sekunden warten müssen bis Ergebnisse erscheinen.

Es wurde durch zwei unterschiedliche Benutzer getestet, wie lange die Antwortzeiten waren. Diese waren alle unter einer Sekunde, was im Rahmen der geforderten Zeit liegt. Der Test wurde durchgeführt indem die laufende Anwendung mit den Developer Tools von Google Chrome, Firefox und Microsoft Edge, untersucht wurde und die angezeigten Antwortzeiten aller Anwendungsseiten immer unter einer Sekunde lagen.

N2 Als Harald Bremer will ich, dass die Anwendung nicht öfter als eine Stunde im Monat nicht erreichbar ist.

Diese Anforderung zu testen erwies sich als schwierig. Sie konnte nicht erfüllt werden. Ein Ansatz zum erfassen der Ausfallzeiten der Anwendung war es, den Slack Channel der ICC [9], über einen bestimmten Zeitraum, beispielsweise drei Monate, auszuwerten. Man würde prüfen wann ein Ausfall gemeldet wurde und wie lange es gedauert hat, bis das Problem behoben wurde. Diese Zeit könnte man dann monatlich zusammenrechnen. Da wahrscheinlich aber nicht alle Ausfallzeiten im Channel gemeldet werden, wäre dies nur ein Richtwert.

6 Erweiterungsöglichkeiten

Es gibt zahlreiche Möglichkeiten die Applikation noch zu erweitern oder zu verbessern. In diesem Kapitel werden, diese Möglichkeiten erläutert. Diese weiteren Wünsche an die Applikation wurden teilweise in den Gesprächen mit den Professoren erfasst. Die restlichen Möglichkeiten stammen aus den nicht erfüllten, oder nur zum Teil erfüllten Anforderungen sowie aus den sich im Laufe des Projekts ergebenden Ideen oder Verbesserungen.

6.1 Verbesserungsmöglichkeiten

Um die Anwendung zu erweitern, haben sich am Ende der Bearbeitung mehrere Verbesserungsmöglichkeiten ergeben.

Zum einen stellte sich heraus, dass, sobald die Anwendung weiter entwickelt werden sollte, ein Backend-Service sehr nützlich sein dürfte. Am Anfang der Bearbeitung erschien dies noch nicht notwendig, da durch den SDK und die zu Verfügung gestellten REST-API von Firebase alle Anforderungen, im Zusammenspiel mit Angular, realisierbar erschienen. Eine weitere Verbesserung betrifft die Datenbank. Aktuell werden viele Daten redundant gespeichert. Dies ist beispielsweise der Fall, wenn eine Abschlussarbeit erstellt und diese dann sowohl für ein Projekt oder Thema als auch öffentlich freigegeben wird. In solch einem Fall werden alle Information der Abschlussarbeit in mehreren Collections hinterlegt. Besser wäre es allerdings, wenn die Abschlussarbeit nur in einer Collection hinterlegt und dann nur noch auf diese Daten referenziert wird.

6.2 Nicht erfüllte Anforderungen

Hier geht es um die nicht oder nur zum Teil erfüllten Anforderungen aus der Anforderungsanalyse. Zu den unerfüllten Anforderungen gehören, wie aus Anforderung A5 hervorgeht, dass Termine auf der Startseite in einem Kalender angezeigt werden. Dazu

müsste man einen Kalender implementieren und alle Termine darin aufrufen und passend anzeigen lassen.

Aus Anforderung A4 geht hervor, dass Themen, laufende und fertige Abschlussarbeiten auf Webseiten gepusht werden können. Für diese Anforderung würde ein der in Abschnitt 6.1 erwähnte Backend-Service benötigt werden. Der Backend-Service könnte dann beispielsweise eine Schnittstelle zur Verfügung stellen, die die Webseiten dann aufrufen könnten. Eine andere Alternative wäre, dass die Webseiten eine Schnittstelle zur Verfügung stellen, auf die der Backend-Service dann zugreifen könnte.

Wie in Anforderung A10 erwähnt, soll der Gesamtzusammenhang zwischen allen Projekten und Themen dargestellt werden. Dies könnte über eine Art Mindmap, ähnlich wie in Abbildung 4.2 realisiert werden. Eine weitere nicht komplett erfüllte Anforderung ist die Anforderung A6. Hier ist noch eine Suchfunktion für Konferenzen und Paper zu ergänzen. Diese Suchfunktion soll beispielsweise aus den Daten von Seiten wie WikiCFP [15] die wichtigsten Informationen bereit stellen. Dies würde voraussetzen, dass die Informationen von diesen Seiten beschafft und dann in einer Datenbank gespeichert werden müssen, um dann ausgelesen und präsentiert werden zu können. Eine Möglichkeit die Informationen zu beschaffen wäre es bestehende Schnittstellen der Webseiten zu nutzen.

6.3 Ideen und Wünsche der Professoren

Hier geht es um die Wünsche der Professoren, die während der Befragungen noch aufgetaucht sind und noch nicht in den Anforderungen erwähnt wurden.

Die erste Idee war es, alle Kontakte in Kategorien einteilen zu können. Ein Beispiel dafür war, Kontakte aus unterschiedlichen Wirtschaftsbereichen unterscheidbar abspeichern zu können. Kontakte in Kategorien einzuteilen ist mit Firebase durchaus möglich. Jeder Kontakt würde dann ein oder mehrere Felder mit der Kategorie bekommen. In der dann neu zu schaffenden Kontaktübersicht müsste dann jeweils abgefragt werden, welche Kontakte in welche Übersicht gehört. Diese Datenbankabfrage ist in Angular allerdings gut umzusetzen.

Ein weiterer Wunsch war es, terminbezogene Daten automatisch zu löschen. Beispiel dafür wären Konferenzen, deren Deadlines abgelaufen sind. Diese sollen nach dem Ablaufdatum automatisch gelöscht werden. Das Löschen sollte über eine Funktion, die beispielsweise täglich prüft, ob ein Datum noch vor dem aktuellen Datum liegt, realisierbar sein. Ist das Datum älter als das Aktuelle, soll der Datensatz gelöscht werden.

Eine große Erleichterung für Professoren wäre es auch, dass jeder Professor mit seinen Studenten alle möglichen Daten teilen kann. Beispielsweise wäre ein gemeinsamer Terminplan, wo Professoren offene Termine für Studenten einstellen können und Studenten sich diese blockieren könnten. Dies würde viel Zeit bei der Findung von Terminen sparen.

6.4 Weitere Erweiterungsmöglichkeiten

In diesem Abschnitt geht es um Ideen oder Erweiterungen, die während der Durchführung des Projekts aufgekommen sind.

Eine weitere wichtige Möglichkeit die Anwendung auf der Seite des Quellcodes zu erweitern sind Tests. Testen ist wichtig. Allerdings ist testen bei Webanwendungen schwierig. Mit Tools wie Selenium ist testen aber möglich. Testen eines neuen Backend-Services, wie in Abschnitt 6.1 beschrieben, wäre im Vergleich zum testen von Webanwendungen einfacher.

7 Fazit

Eine Applikation zur Unterstützung von Professoren gibt es, in der Form wie es hier programmiert wurde, nicht auf dem Markt. Die Applikation vereinfacht das Management für Professoren, da durch die Übersichtlichkeit, die durch die Applikation geschaffen wurde, die Zugehörigkeiten von Informationen klar gegliedert sind. Der öffentliche Teil macht das Ganze noch benutzerfreundlicher, da man in Projekten oder Themen, die in den öffentlichen Daten gespeicherten Informationen auswählen kann und nicht alles separat pro Projekt oder Thema erneut eingeben muss.

Beim Erstellen der Applikation mit Angular gab es Anfangs kleinere Probleme beim Aufbau. Im Laufe des Projektes stellte sich heraus, dass viele Komponenten überflüssig waren, da fast jede Komponente den selben Aufbau hatte und sich nur in den Daten unterschieden hat. Durch das Komprimieren des Codes wird auch die Wartung vereinfacht, da jetzt nur noch eine Komponente gewartet werden muss, zu Beginn des Projekts waren es zehn.

Im Laufe des Projekts wurde auch die Datenstruktur im Google Firestore immer komplexer. Beim Arbeiten mit Firestore stellte sich heraus, dass viel Zeit in die Strukturierung von Daten investiert werden muss. Sollte das Projekt also weiter geführt werden, sollte man dies auch nochmals prüfen, da im Moment noch viele Daten redundant gespeichert werden.

Literaturverzeichnis

- [1] *The Architecture of Angular*. – URL <https://angular.io/guide/architecture>. – Eingesehen am 26.03.2019
- [2] *Angular im Vergleich mit React*. – URL <https://mlsdev.com/blog/angular-vs-react>. – Eingesehen am 02.07.2019
- [3] *Google Open Source Beschreibung von Angular*. – URL <https://opensource.google.com/projects/angular>. – Eingesehen am 01.07.2019
- [4] BERGSMANN, Johannes: *Requirements Engineering für die agile Softwareentwicklung*. dpunkt.verlag GmbH, 2018. – ISBN 9783864904851
- [5] *Hochschul Informations System eG - Produkte*. – URL <https://www.his.de/produkte.html>. – Eingesehen am 29.03.2019
- [6] *Unterschiede zwischen den von Google Firebase bereitgestellten NoSQL Datenbanken*. – URL <https://firebase.google.com/docs/database/rtdb-vs-firestore>. – Eingesehen am 29.03.2019
- [7] *Einführung in den Firebase Authentifizierungs Service*. – URL <https://firebase.google.com/docs/auth>. – Eingesehen am 02.07.2019
- [8] *Dokumentation von Google Scholar*. – URL <https://scholar.google.com/intl/de/scholar/about.html>. – Eingesehen am 02.07.2019
- [9] *Link zu Slack Channel der ICC*. – URL <https://haw-hh-inf.slack.com/>. – Eingesehen am 05.07.2019
- [10] *Elsevier Solution Mendeley*. – URL <https://www.elsevier.com/de-de/solutions/mendeley>. – Eingesehen am 25.03.2019
- [11] *Funktionen von Moodle*. – URL <https://docs.moodle.org/37/de/Funktionalitäten>. – Eingesehen am 02.07.2019

- [12] *Webseite von React.* – URL <https://reactjs.org/>. – Eingesehen am 01.07.2019
- [13] SOMMERVILLE, Ian: *Software Engineering.* Pearson Deutschland GmbH, 2018. – ISBN 9783868943443
- [14] *Dokumentation von Moodle. Grundlegende was ist Moodle.* – URL https://docs.moodle.org/37/de/Was_ist_Moodle. – Eingesehen am 01.07.2019
- [15] *WikiCFP, A Wiki for Calls for Papers.* – URL <http://www.wikicfp.com/cfp/>. – Eingesehen am 02.07.2019

A Anhang

A.1 Aufzeichnung Interviews mit den Professoren

Interview Fragen zum Thema Forschungsmanagement

Was erwarten sie von einem System zum Forschungsmanagement? Welche großen Bereiche sehen sie da? Beispiel Abschlussarbeiten.

- Forschungsprojekte und Forschungsthemen → verknüpft mit bspw. Abschlussarbeiten
- Konferenzen (am besten eine Suchfunktion von mehreren Webseiten u.a. WikiCFP, HAW eigene), Klassifizierung von Konferenzen
- Suchfunktion für Förderangebote
- Publikationen → vielleicht verlinken
- Paper siehe Publikation. Deadlines sollen angezeigt werden.
- Kontakte: Aufteilung in Forschungsbereiche, Ansprechpartner von Firmen, Kontakt hat teilgenommen an dem und dem Projekt, Unternehmensdatenbank
- Literaturverzeichnisse → teilbar und mit Projekten/Themen verlinken
- Termine → Kalender; Deadline für Forschungsanträge usw.
- Brainstorming
- Übersicht welche Themen und Projekte mit welchen zusammenhängen oder Verknüpft sind. Ob sie Unter- oder Überprojekt oder Thema sind.
- Themenhistorie von Personen → damit man weiß, was derjenige schon gemacht hat damit man ihn eventuell mal was danach fragen kann. Für neue Projekte, falls er/sie auch mit in das projekt will. Am besten angezeigt in Kategorien

Welche Aufgaben erledigen sie Breits mit Software? Welche Software nutzen sie dafür?

- Excel: Abschlussarbeiten verwalten als Themen, laufende und fertige Abschlussarbeiten wobei die Dateien in der Owncloud oder auf dem PC liegen.
- Mendeley, Zotero, Bibtex: Literaturverzeichnis
- Google Scholar: Suchen von Publikationen, Papern usw.
- Email, Whatsapp, Mattermost, Skype, Slack, Emil, Wire: Kommunikation
- Google Kalender: Termine verwalten
- GitLab, HAW Owncloud, GitHub, Google Drive: Speicherorte für Datei, Bilder, Code, etc.
- Call4Paper, WikiCFP: Konferenzen

Gibt es dort Daten, die integriert/migriert werden sollten?

- Sehr unterschiedliche Meinungen
- Kalender integrieren
- Lieber Verlinkungen z.B. zu Gruppen Chats
- Keine Paper veröffentliche/integrieren → eventuell nicht Rechtssicher → besser wäre Verlinkung

Wie könnte eine Einbindung der verwalteten Informationen in eine Webseite aussehen?

- Unterschiedliche Meinungen.
- 1. Sobald sich was ändert soll es sich auch auf der Webseite ändern.
- 2. Sobald sich was ändert soll gefragt werden ob es auf der Webseite geändert werden soll

- 3. Es muss eine Aufteilung geben welche Webseite sich ändern soll wenn es auf 2 oder 3 Webseiten steht.

Welche Informationen möchten sie mit der Forschungsgruppe, mit anderen Kollegen an der HAW oder außerhalb, mit Studis, mit Doktoranden, Firmenkontakten teilen?

Studies/Doktoranden:

- Sollen neuste Versionen ihrer Abschlussarbeiten hochladen können.
- Profs wollen Termine für Sprechzeiten erstellen, für die sich die Studie dann eintragen können und diese so für sich blocken
- Termine → Kolloquien, Termine für Fortschritt in der BA

Firmenkontakte bzw. externe Kontakte:

- Anträge für Forschung → Linkspeicherort
- Deadlines für Forschung
- Abschlussarbeiten sowohl Ideen als auch laufende und fertige → könnte für Firmen interessant sein

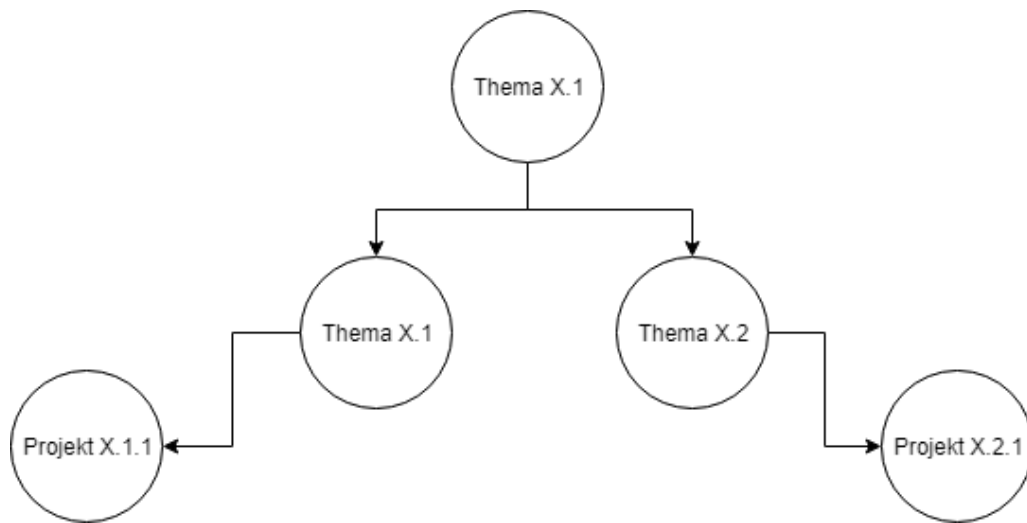


Abbildung A.1: Erster schneller Entwurf der Verknüpfungen zwischen Themen und Projekten während eines Interviews mit einem Professor

A.2 Screenshots der Applikation

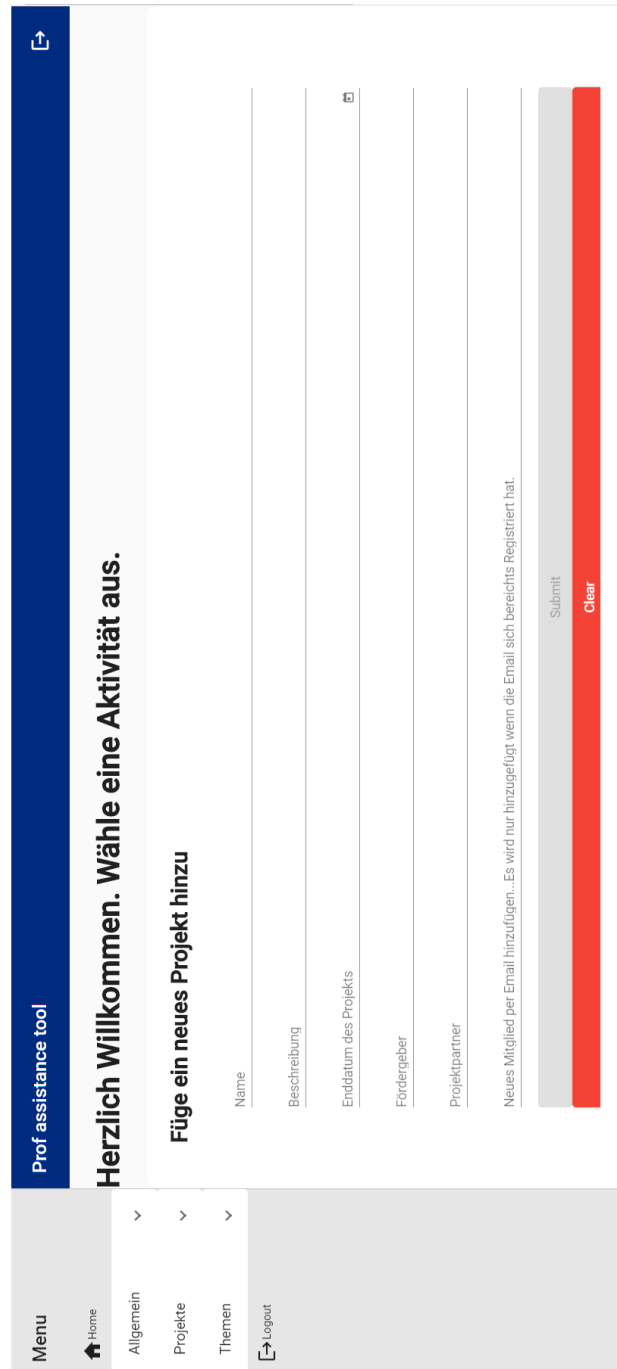


Abbildung A.2: Screenshot der Homepage und des Sidenavs des Programms

Allgemein ^

Vorschläge für
Abschlussarbeiten
Laufende
Abschlussarbeiten
Fertige
Abschlussarbeiten

Links

Termine

Kontakte

Publikationen

Konferenzen

Projekte ^

Projekt 1

Projekt 2

Themen ^

Thema 2

Thema 1

Abbildung A.3: Screenshot vom offenen Sidenav mit bestehenden Projekten, Themen

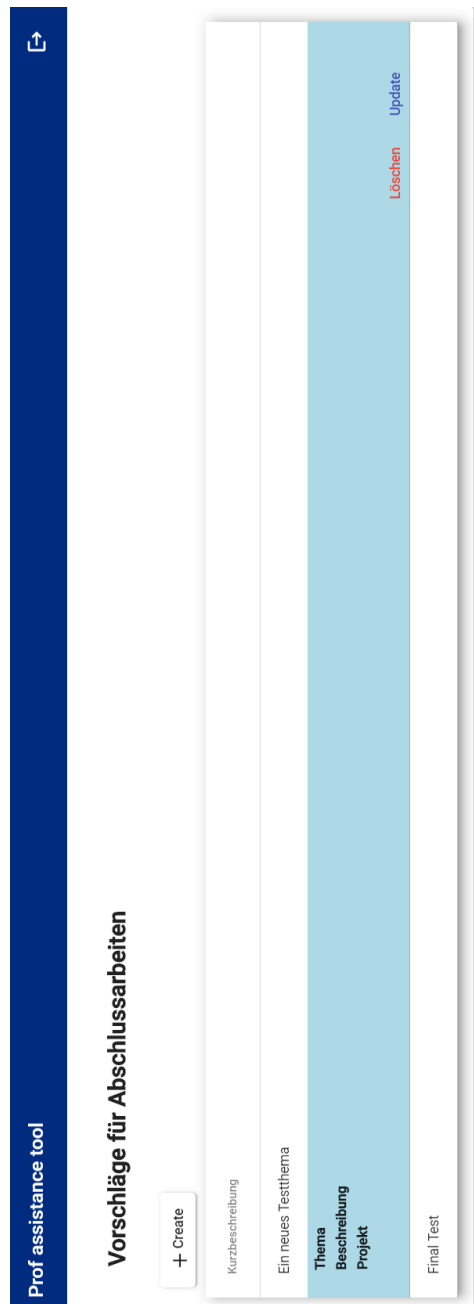


Abbildung A.4: Screenshot der öffentlichen Seite von Ideen für Abschlussarbeiten

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Entwurf und prototypische Umsetzung eines Tools zum Forschungsmanagement von Professoren

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort Datum Unterschrift im Original