

OST ResearchNavigator: Projekt-Tracking-Software für die Ostschweizer Fachhochschule

Autoren

Simon Hefti & Livio Mauchle

Bachelorarbeit

OST – Ostschweizer Fachhochschule
Campus Rapperswil-Jona

Betreuung

Laurent Metzger
Fabio Daniel Marti
Clemens Meier

13.06.2025

Abstract

Die Ostschweizer Fachhochschule (OST) führt zahlreiche anwendungsorientierte Forschungsprojekte in verschiedenen Bereichen durch. Derzeit werden diese über fachspezifische Kanäle veröffentlicht, was ein gesamtheitliches Monitoring der Forschung umständlich macht. Um dieser Problematik entgegenzuwirken, wurde im Rahmen einer vorangegangenen Studienarbeit ein erster Prototyp einer webbasierten Plattform – der OST ResearchNavigator – entwickelt. Dieser ermöglicht es, Forschungsprojekte zentral zu erfassen, einheitlich zu verwalten und effizient zu durchsuchen. Ziel der vorliegenden Bachelorarbeit ist die Weiterentwicklung des bestehenden Prototyps mit einer Benutzer-Authentifizierung und einem Admin-Bereich zur Stammdatenverwaltung zu einem voll funktionstauglichen und einsatzfähigen Produkt.

Das bestehende Datenmodell wurde ergänzt, um den neuen Anforderungen zu entsprechen. Der vorhandene Tech-Stack, bestehend aus einem Nuxt.js-Frontend, einer ASP.NET Core API, einer PostgreSQL Datenbank sowie einem MinIO Filesystem, wurde übernommen. Für neue Features wurden zusätzliche Technologien integriert, nämlich GitLab für die Authentifizierung, die FluentEmail-Library für den E-Mail-Versand sowie LLMHub und OpenAI zur Generierung von Zusammenfassungen aus PDF-Dateien. Die Plattform LLMHub wird OST-intern gehostet, was die Datensicherheit steigert. Erkenntnisse aus der vorherigen Arbeit sowie regelmässige Code Reviews führten zu einer umfassenden Neustrukturierung des Backend-Codes. Durch mehrere Benutzertests mit potenziellen Anwendern der Software wurde Feedback gesammelt und laufend integriert. So wurde die an das OST Corporate Design angelehnte Applikation intuitiv, benutzerfreundlich und gezielt auf die Bedürfnisse des Kunden abgestimmt. Alle Komponenten sind mit Docker containerisiert, Einstellungen lassen sich in einer simplen Konfigurationsdatei vornehmen und sind übersichtlich dokumentiert.

Alle definierten Anforderungen konnten vollständig umgesetzt werden. Das entstandene System zeichnet sich durch eine sichere Benutzerauthentifizierung, eine benutzerfreundliche Oberfläche zur Verwaltung von Stammdaten und Benutzerkonten sowie erweiterte Funktionen zur Bearbeitung und Archivierung von Projekten aus. Die Suchfunktion wurde ergänzt und bietet nun eine leistungsstarke Volltextsuche mit Relevanzbewertung und die Möglichkeit, Inhalte aus PDF-Dokumenten zu durchsuchen. Zusätzlich wurden ein erweitertes Statistik-Dashboard, ein strukturierter Texteditor mit Vorschaufunktion, E-Mail-Benachrichtigungen, ein PDF-Export, sowie KI-basierte Unterstützung bei der Erstellung von Projektbeschreibungen entwickelt, was die Benutzerfreundlichkeit und Effizienz des Gesamtsystems weiter steigert. Die nicht-funktionalen Anforderungen Bedienbarkeit, Performance und Wartbarkeit wurden auf die neuen Funktionalitäten ausgeweitet und mit geeigneten Tools verifiziert. Die umfassenden Optimierungen des Codes erleichtern die zukünftige Weiterentwicklung.

Die weiterentwickelte Software erweitert den bestehenden Prototypen zu einem umfassenden und intuitiv nutzbaren System. Die Applikation stiess auf positives Feedback von Kunden und potenziellen Anwendern. Sie erfüllt alle Anforderungen für einen produktiven Einsatz und ist somit bereit für die Inbetriebnahme durch ein Institut der OST.

Management Summary

Die Ostschweizer Fachhochschule (OST) führt zahlreiche anwendungsorientierte Forschungsprojekte in verschiedenen Bereichen durch. Die Veröffentlichung dieser Projekte erfolgt derzeit über fachspezifische Kanäle, was ein gesamtheitliches Monitoring der Forschung umständlich macht. Im Rahmen einer vorangegangenen Studienarbeit wurde ein Minimum Viable Product (MVP) einer Anwendung entwickelt, um die Erfassung sowie das Monitoring von Projekten an der OST gezielt zu verbessern. Der erfolgreich umgesetzte Prototyp ermöglicht es, Forschungsprojekte zentral zu erfassen, einheitlich zu verwalten und effizient zu durchsuchen. Zudem bietet er grafische Auswertungen zu aktuellen und abgeschlossenen Forschungsprojekten an der OST.

Für den produktiven Einsatz fehlen jedoch noch zentrale Funktionen wie eine Authentifizierung, die Pflege von Stammdaten sowie die Verwaltung bereits veröffentlichter Projekte. Ziel dieser Arbeit ist es daher, alle notwendigen Kernfunktionen zu implementieren, das System um umfassende Zusatzfunktionen zu erweitern und den bestehenden Prototyp damit zu einer produktionsreifen Gesamtlösung auszubauen.

Die Umsetzung erfolgte nach dem Rational Unified Process (RUP), wobei die einzelnen Phasen in Sprints nach dem Scrum-Framework unterteilt wurden. Dieses Vorgehen ermöglichte eine iterative und flexible Entwicklung. Die Anforderungen an das finale Produkt wurden gemeinsam mit dem Auftraggeber frühzeitig ausgearbeitet. Erarbeitete Ergebnisse wurden periodisch präsentiert, um neue Ideen und Vorschläge flexibel in die Anwendung einzubauen. Die Qualität der Arbeit wurde fortlaufend durch Usability-Tests, Code-Reviews, Unit-Tests sowie Performance-/Load-Tests geprüft und verbessert.

Der bestehende Tech-Stack, bestehend aus einem Nuxt.js-Frontend, einer ASP.NET Core API, einer PostgreSQL-Datenbank und einer Dateiablage mittels MinIO, wurde aus der vorausgegangenen Studienarbeit übernommen. Dieser wurde um eine GitLab-basierte Authentifizierung, einen E-Mail-Dienst sowie die Integration von KI-Funktionalitäten (LLMHub / OpenAI) erweitert. Darüber hinaus wurde im technischen Bereich gezielt auf die Wartbarkeit geachtet, indem technische Schulden aus dem Vorgängerprojekt abgebaut und Verbesserungen an der Architektur der Anwendung implementiert wurden.

Das entwickelte Produkt ermöglicht es Benutzern, Forschungsprojekte nicht mehr nur wie bisher nach Titel, Autor oder Abstract zu durchsuchen, sondern nach sämtlichen verfügbaren Projektmetadaten sowie auch nach Texten, die sich in angehängten PDF-Dokumenten befinden. Die Suchfunktion lässt sich flexibel mit den bestehenden Filtern kombinieren. Zur besseren Übersicht werden dabei alle aktiven Filter in der Filterleiste als Tags angezeigt (siehe Abbildung 1).

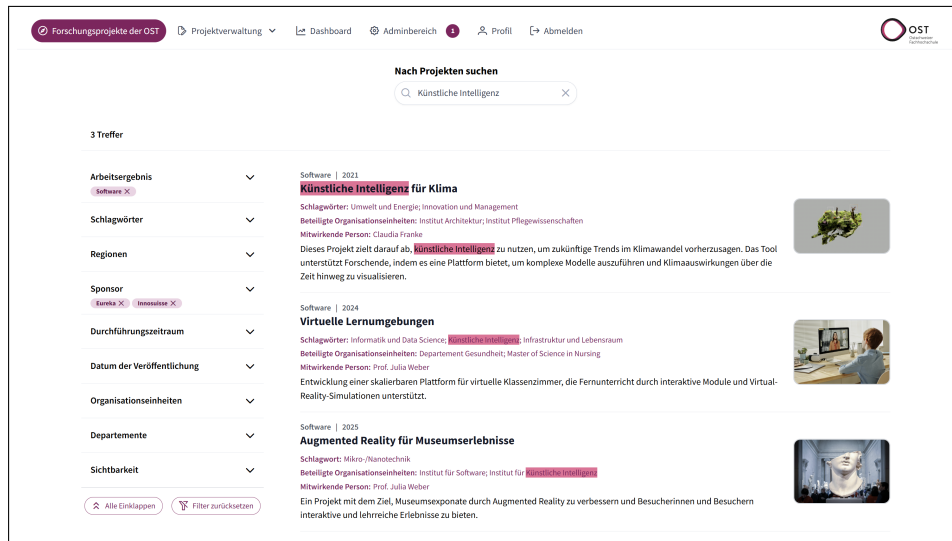


Abbildung 1: Startseite der Anwendung

Die Übersicht der Projektstatistiken (siehe Abbildung 2) wurde um zusätzliche Grafiken erweitert. Neu besteht zudem die Möglichkeit, Lesezeichen anzulegen, um gesetzte Filtereinstellungen zu speichern.

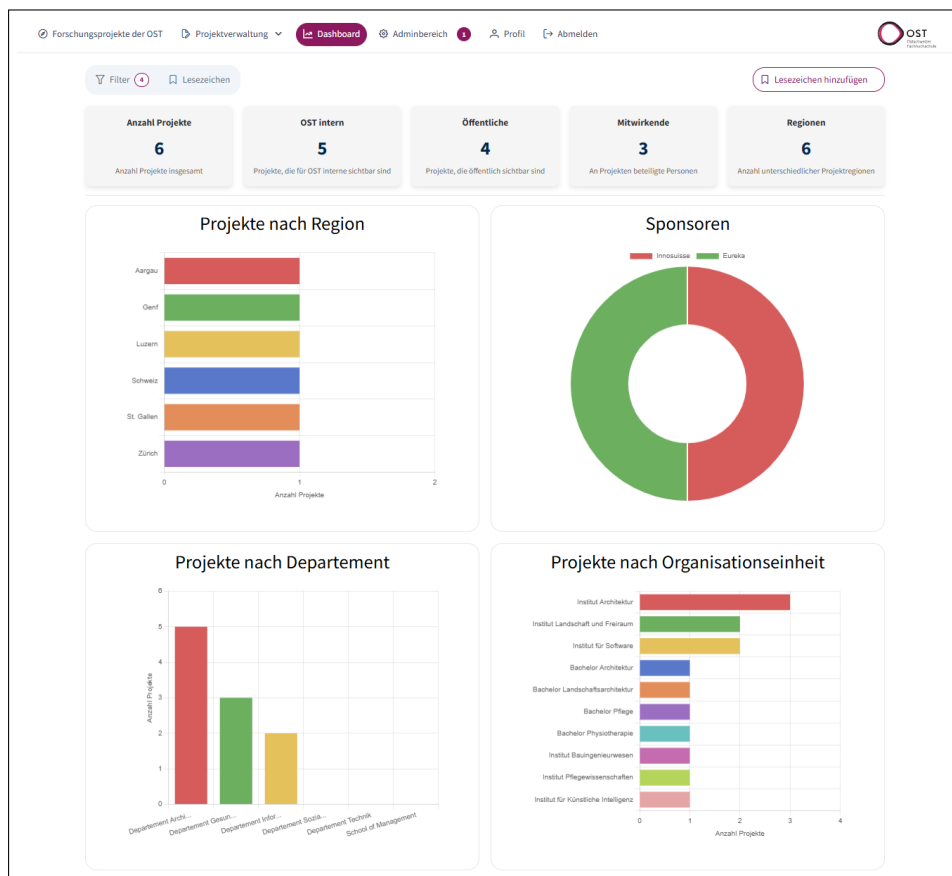


Abbildung 2: Dashboard zur Anzeige von grafischen Statistiken

Anwender können ein Benutzerkonto erstellen und sich über GitLab authentifizieren. Die Zugriffsrechte werden durch mehrere vordefinierte Rollen klar geregelt. Diese Rollen sowie weitere Profildaten lassen sich über eine Administrationsoberfläche vom autorisierten Personal der OST verwalten. Neue Benutzerkonten sowie Änderungen an bestehenden Accounts müssen aus Sicherheitsgründen von einem Administrator geprüft und freigegeben werden. Anhand einer integrierten Stammdatenverwaltung (siehe Abbildung 3) lassen sich zudem Änderungen an den Metadaten auch im laufenden Betrieb von berechtigten Personen vornehmen.

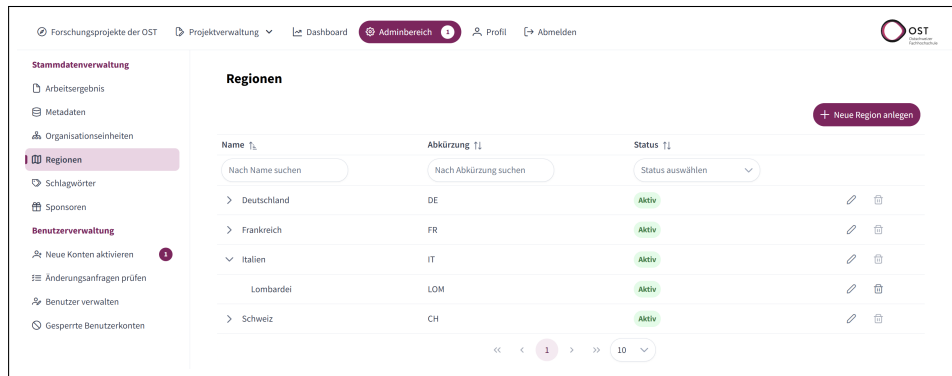


Abbildung 3: Stammdatenverwaltung - Seite zur Verwaltung von Regionen

Die Erfassung von Projekten wird durch den Einsatz Künstlicher Intelligenz unterstützt, die Projektbeschreibungen anhand hochgeladener PDF-Dateien generiert. Ein neu integrierter Texteditor ermöglicht zudem die Formatierung dieser Beschreibungen (siehe Abbildung 4). Während der Bearbeitung kann das Projekt jederzeit in einer Vorschau angezeigt werden.

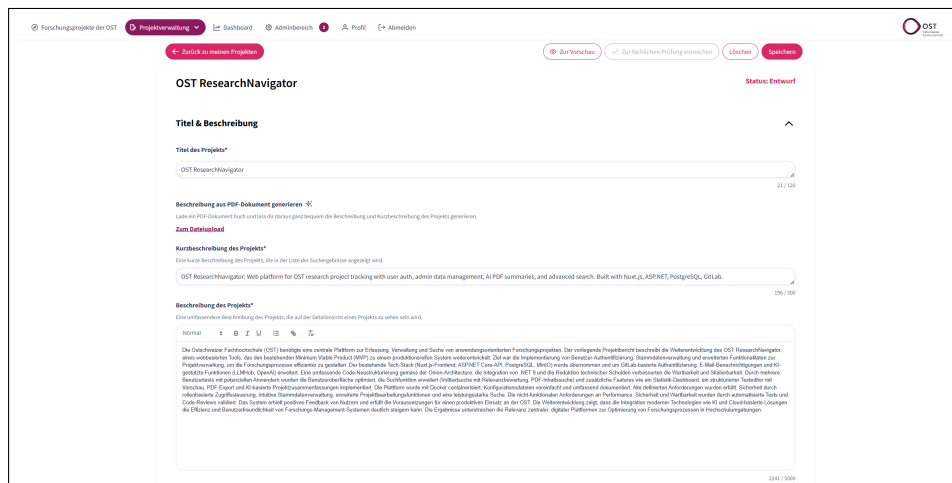


Abbildung 4: Seite zur Projektbearbeitung

Weitere umgesetzte Funktionen umfassen die Verwaltung bestehender Projekte, den Export von PDF-Dokumenten aus den Projektbeschreibungen sowie den Versand von Benachrichtigungs-E-Mails zur Unterstützung der Projektfreigabe und Benutzerverwaltung.

Durch diese Funktionen wie auch durch technische und optische Verbesserungen der Anwendung konnte der Prototyp zu einem einsatzfähigen Produkt weiterentwickelt werden. Das gesamte System ist containerisiert und kann künftig von einem Institut der OST übernommen, betrieben und erweitert werden.

Inhaltsverzeichnis

Abbildungsverzeichnis	VIII
Tabellenverzeichnis	X
1 Einleitung	1
1.1 Ausgangslage	1
1.2 Problemstellung	1
1.3 Aufgabenstellung	2
1.4 Organisatorische Rahmenbedingungen	2
1.5 Abgrenzung	3
2 Analyse & Anforderungen	4
2.1 Funktionale Anforderungen	5
2.1.1 Akteure	5
2.1.2 Use Case Diagramm	6
2.1.3 Use Cases	6
2.2 Nicht-Funktionale Anforderungen	18
2.3 Nicht Anforderungen	22
2.3.1 Integration weiterer Single-Sign-On-Provider	22
2.3.2 Push-Benachrichtigungen via Desktop-Notifications	22
2.4 Domain Model	22
3 Lösungskonzept	24
3.1 Analyse und Evaluation	24
3.1.1 Auswahl der KI-Modelle	24
3.1.2 Authentifizierung	25
3.1.3 Abläufe bei der Benutzerverwaltung	26
3.1.4 Suchoptimierung und Rankingalgorithmus	27
3.1.5 Stammdatenverwaltung	28
3.2 Architektur	29
3.2.1 Kontext & Abgrenzung	29
3.2.2 Lösungsstrategie	31
3.2.3 Bausteinsicht Level 1: Containers	31
3.2.4 Bausteinsicht Level 2: Components	32
3.2.5 Bausteinsicht Level 3: Code	34
3.2.6 Laufzeitsicht	39

3.2.7	Deployment-Sicht	40
3.2.8	Querschnittliche Konzepte	41
3.2.9	Architekturentscheidungen	42
3.3	Technische Schulden	44
3.4	Design	45
3.4.1	Anforderungen und Allgemeines Konzept	45
3.4.2	Accessibility	45
3.4.3	Mockups	46
3.4.4	Verbesserung der Projektsuche	49
3.4.5	Designentscheide	52
4	Umsetzung	53
4.1	Suchoptimierung	53
4.1.1	Datenbank Trigger	53
4.1.2	Suche in PDF-Dokumenten	54
4.2	Authentifizierung	55
4.2.1	Setup	55
4.2.2	GitlabService	55
4.2.3	Provider-Wechsel	57
4.3	KI-Zusammenfassungen	57
4.4	Setup	57
4.4.1	LLMHubClient	58
4.4.2	OpenAIClient	60
4.4.3	FileService	60
4.5	Texteditor	61
4.5.1	Content Sanitizing (Backend)	61
4.5.2	Content Sanitizing (Frontend)	62
4.6	E-Mail-Versand	62
5	Qualitätsmassnahmen	64
5.1	Code Qualität	64
5.2	Versionskontrolle	65
5.2.1	Monorepository Ansatz	65
5.2.2	Workflow	65
5.2.3	Pipeline	66
5.3	Code Reviews	66
5.4	Testing	66
5.4.1	Unit Tests	66
5.4.2	Usability Tests	67
5.4.3	System Tests	68
5.4.4	Load Tests	68
5.5	Dokumentation des Codes	68
6	Projektplan	69
6.1	Vorgehen im Projekt	69
6.2	Zeitplan	69

6.2.1	Meilensteine	71
6.3	Risikomanagement	71
6.3.1	Initiale Risikoauswertung vom 24.02.2025 - Ende der Inception Phase	71
6.3.2	Risikoauswertung vom 09.03.2025 - Ende der Elaboration Phase	72
6.3.3	Risikoauswertung vom 20.04.2025 - Mitte der Construction Phase	72
6.3.4	Risikoauswertung vom 01.06.2025 - Mitte der Transition Phase	73
6.3.5	Identifizierte Risiken	74
7	Ergebnisse	76
7.1	Übersicht der Ergebnisse	76
7.1.1	Statistiken zum Code	81
7.1.2	Zielerreichung	81
8	Ergebnisdiskussion	83
8.1	Fazit	83
8.2	Ausblick	84
8.2.1	Erweiterungen	84
8.2.2	Verbesserungsmöglichkeiten	84
8.3	Weiterentwicklung	84
	Literaturverzeichnis	86
9	Anhang	91
9.1	Tests und Qualität	91
9.2	Ergänzende Inhalte	154

Abbildungsverzeichnis

1	Startseite der Anwendung	III
2	Dashboard zur Anzeige von grafischen Statistiken	III
3	Stammdatenverwaltung - Seite zur Verwaltung von Regionen	IV
4	Seite zur Projektbearbeitung	IV
2.1	Use Case Diagramm	6
2.2	Domain Model	23
3.1	Ablaufdiagramm Benutzerverwaltung	26
3.2	Kontext Diagramm	30
3.3	Container Diagramm	32
3.4	Component Diagramm	33
3.5	Onion Architektur	34
3.6	Entity Relationship Diagram	38
3.7	OIDC Authentication Flow zwischen Client, ASP.NET Core Web API und GitLab	39
3.8	C4 Deployment Diagramm	40
3.9	Mockup der Stammdatenverwaltung	47
3.10	Mockup der Benutzerverwaltung	48
3.11	Filter vor den Verbesserungen	49
3.12	Filter nach den Verbesserungen	49
3.13	Vergleich der Filterleisten vor und nach den Usability-Verbesserungen	49
3.14	Datumsfilter vor Verbesserungen	50
3.15	Datumsfilter nach Verbesserungen	50
3.16	Vergleich der Datumsfilter vor und nach den Verbesserungen	50
3.17	Seiter der Projektsuche vor den Verbesserungen	50
3.18	Seite der Projektsuche nach den Verbesserungen	51
5.1	Testabdeckung der Service-Klassen im Backend	66
5.2	Testabdeckung der Unit Tests im Frontend	67
6.1	Projektplan	70
7.1	Screenshot Startseite	76
7.2	Zusammenstellung der vier neuen Statistiken	77
7.3	Screenshot Lesezeichenleiste	77
7.4	Screenshot Stammdatenverwaltung: Seite „Regionen“	78
7.5	Screenshot Benutzerverwaltung: Seite „Benutzer verwalten“	78

7.6	Screenshot der Profilseite „Profil bearbeiten“	79
7.7	Screenshot des Texteditors	80
7.8	Screenshot archivieren und Neufassung erstellen	80
7.9	Screenshot eines Benachrichtigungs-Mails	80
9.1	Identity Cookie und Content Headers für alle Requests	118
9.2	CSV Data Set Config mit IDs für die Delete-Tests	118
9.3	Request für die Erstellung eines ProjectTypes	119
9.4	Request für das Löschen eines ProjectTypes mit dynamisch eingefügter ID in der URL	119
9.5	Request für die Bearbeitung eines ProjectTypes mit ID und dynamisch eingefügter Version (aus CSV) im Body.	120
9.6	Request für die Bearbeitung eines verwendeten SearchTags mit ID, ParentID und dynamisch eingefügter Version (aus CSV) im Body.	120
9.7	Summary Report für den gesamten Testplan (50 Threads). Die maximale Response-Time aller Stammdatenoperationen liegt mit 0.1s weit unter dem Grenzwert (2s). Die Error Rate von 48% entstand durch Concurrency-Probleme beim ersten Durchlauf, wodurch die Version des Datensatzes nicht mehr in der richtigen Reihenfolge beim Server ankam. Dieses Verhalten ist gewollt.	121
9.8	Aggregate Report für den gesamten Testplan (50 Threads). Die maximale Response-Time liegt auch hier mit 1.57s unter dem Grenzwert (2s).	121
9.9	Anzahl Codezeilen im Frontend	154
9.10	Anzahl Codezeilen der Unit Tests im Frontend	154
9.11	Anzahl Codezeilen im Backend	154
9.12	Anzahl Codezeilen der Unit Tests im Backend	155

Tabellenverzeichnis

2.1	Farbcodes der Anforderungen	4
2.2	Beschreibung der Akteure	5
2.3	UC01 - Authentifizierung	7
2.4	UC02 - Stammdatenverwaltung	8
2.5	UC03 - veröffentlichtes Projekt verwalten	9
2.6	UC04 - KI-Abstract-Generator	10
2.7	UC05 - Benutzerverwaltung	11
2.8	UC06 - Benachrichtigungs-E-Mail versenden	12
2.9	UC07 - Suchoptimierung	13
2.10	UC08 - Dashboard-Erweiterung	14
2.11	UC09 - Editor-Erweiterung	15
2.12	UC10 – PDF generieren	16
2.13	UC11 - PDF Textsuche	17
2.14	UC12 - Projekte aus SAP laden	17
2.15	NFR-1: Response Time Projektbearbeitung	19
2.16	NFR-2: Affordance Anmeldung	19
2.17	NFR-3: Authorization	20
2.18	NFR-4: Test Coverage	20
2.19	NFR-5: Konfiguration	21
2.20	NFR-6: Backups	21
3.1	Priorisierung der Metadaten für den Rankingalgorithmus	27
3.2	Grundfunktionen der Stammdatenverwaltung	28
3.3	Zentrale Technologien aus der Vorgängerarbeit	29
3.4	Lösungsstrategie	31
3.5	Erklärung der Ordnerstruktur	35
3.6	Erklärung der Ordnerstruktur (SA)	35
3.7	Erklärung der Ordnerstruktur (SA)	36
3.8	Technische Schulden	44
3.9	Accessibility Massnahmen	46
3.10	Designentscheide	52
5.1	Massnahmen zur Verbesserung der Code Qualität Teil 1/2	64
5.2	Massnahmen zur Verbesserung der Code Qualität Teil 2/2	65
6.1	Meilensteine	71

6.2	Risikomatrix 24.02.2025	71
6.3	Risikomatrix 09.03.2025	72
6.4	Risikomatrix 20.04.2025	72
6.5	Risikomatrix 20.04.2025	73
6.6	Risiko - Umfang	74
6.7	Risiko - Bugs	74
6.8	Risiko - Usability	74
6.9	Risiko - Security	75
6.10	Risiko - KI-Integration	75
6.11	Risiko - Nuxt Performance	75
7.1	Auflistung der Anzahl Codezeilen	81
9.1	Testfälle Authentifizierung	92
9.2	Testfälle Stammdatenverwaltung	93
9.3	Testfälle Stammdatenverwaltung Teil 2	94
9.4	Testfälle Projektverwaltung	95
9.5	Testfälle Projektsuche	96
9.6	Testfälle Benutzerverwaltung	97
9.7	Testfälle KI-Beschreibungen und Editor	98
9.8	Testfälle Dashboard Erweiterung	99
9.9	Verbesserungen durch Usability Tests Teil 1	100
9.10	Verbesserungen durch Usability Tests Teil 2	101
9.11	Code Review - Feedback und Massnahmen Teil 1/4	102
9.12	Code Review - Feedback und Massnahmen Teil 2/4	103
9.13	Code Review - Feedback und Massnahmen Teil 3/4	104
9.14	Code Review - Feedback und Massnahmen Teil 4/4	105
9.15	Code Review - nicht oder nur teilweise behobene Mängel	106

Codeverzeichnis

4.1	Ausschnitt Trigger	54
4.2	Registrierung der OIDC Authentication Services	55
4.3	GitlabService, welcher im OnTokenValidated-Event aufgerufen wird.	56
4.4	Registrierung des konfigurierten LLM-Clients	57
4.5	SummarizeAsync-Implementation im LLMHub-Client	58
4.6	GetResponseFromPrompt-Methode im LLMHub-Client	59
4.7	GetFirstAvailableModel-Methode im LLMHub-Client	59
4.8	SummarizeAsync-Implementation im OpenAI-Client	60
4.9	Aufruf von SummarizeAsync in der FileService-Methode GetSummary	60
4.10	Konfiguration des HtmlSanitizer	61
4.11	SendUserChangeRequestNotification-Implementierung im EmailSender	62
4.12	EnqueueEmail-Methode im EmailSender	63

1 Einleitung

Diese Bachelorarbeit knüpft an den im Rahmen einer **Studienarbeit (SA)** [1] als **Minimum Viable Product (MVP)** entwickelten OST ResearchNavigator an und führt dessen Prototypenstatus zu einem einsatzreifen System weiter. In diesem Kapitel werden zunächst die Ausgangssituation erläutert und anschliessend die Zielsetzung der vorliegenden Folgearbeit aufgezeigt.

1.1 Ausgangslage

Die **Ostschweizer Fachhochschule (OST)** führt anwendungsorientierte Forschungsprojekte in verschiedenen Fachbereichen durch. Die gewonnenen Ergebnisse werden erfolgreich über fachspezifische Kanäle veröffentlicht. Im Rahmen einer vorangegangenen **SA** wurde das **MVP** des „OST ResearchNavigator“ entwickelt. Die entwickelte Webapplikation ermöglicht eine zentrale Erfassung, Verwaltung und Suche von Forschungsprojekten.

Das **MVP** deckt Kernfunktionen wie die Projekterfassung, den Publikations-Workflow, standardisierte Filter- und Suchmöglichkeiten sowie ein Dashboard mit Statistiken bereits ab. Funktionalitäten wie ein Authentifizierungsmechanismus oder die Verwaltung von Stammdaten sind im **MVP** hingegen noch nicht enthalten, da sie bewusst nicht Teil des initialen Projektumfangs waren.

Die vorliegende **Bachelorarbeit (BA)** knüpft unmittelbar an diesen Prototyp an und verfolgt das Ziel, ihn zu einem einsatzreifen Produkt auszubauen.

1.2 Problemstellung

Für eine produktive Einführung der Anwendung gilt es folgende Hürden zu überwinden, deren Umsetzung im **MVP** aufgrund des definierten Projektumfangs zunächst ausgeklammert wurde:

- **Sicherheit und Governance:** Aktuell verfügt die Anwendung über keinerlei Authentifizierung. Somit kann keine rollenbasierte Zugriffskontrolle umgesetzt werden. Sensible Projektdaten sind dadurch unzureichend geschützt.
- **Stammdatenpflege:** Die manuelle Pflege von Stammdaten (Organisationseinheiten, Sponsoren usw.) ist nur für technisch versierte Mitarbeiter zumutbar und lediglich direkt über die Datenbank möglich.
- **Verwalten von bestehenden Projekten:** Es fehlt die Möglichkeit, bereits veröffentlichte Projekte nachträglich zu bearbeiten oder zu archivieren.
- **Eingeschränkte Suchfunktionen:** Die Volltextsuche beschränkt sich auf die Autoren, den Titel und die Beschreibung eines Projekts. Zudem existiert kein Ranking der Suchergebnisse.

- **Verzögerter Veröffentlichungsworkflow:** Ohne Benachrichtigungen über anstehende Aufgaben (z. B. „Projekt prüfen“) verlangsamt sich der gesamte Freigabeprozess.
- **Begrenzte Auswertungsmöglichkeiten:** Es fehlen erweiterte Dashboards und eine Funktion zum Speichern individueller Filtereinstellungen, die eine effizientere Analyse von Forschungstrends ermöglichen würden.

Über die oben genannten Punkte hinaus können zusätzliche Funktionen wie etwa die automatische Generierung von Projektbeschreibungen per KI, eine inhaltsbasierte Suche in zugehörigen PDF-Dokumenten, ein PDF-Export der Projektbeschreibungen sowie ein moderner, funktionsreicher Texteditor die Benutzerfreundlichkeit weiter steigern.

1.3 Aufgabenstellung

Diese Bachelorarbeit baut auf dem Vorgängerprojekt „OST ResearchNavigator“ auf und hat zum Ziel, den bestehenden Prototypen zu einem vollständig einsetzbaren Tool für die OST zu entwickeln. Die Arbeit sollte gemäss Kundenwünschen und mit Ausrichtung auf die zukünftige Inbetriebnahme besonders auf die Punkte Wartbarkeit, Verständlichkeit und Funktionalität umgesetzt werden. Spezifische Wünsche und Anforderungen sind laufend mit dem Kunden zu besprechen und demnach sind die spezifischen Projektziele aktuell zu halten. Die Schwerpunkte dieser Bachelorarbeit bestehen aus folgenden Punkten:

- Authentifizierung mit GitLab
- Bearbeitbare Stammdatenverwaltung (Suchfilter)
- Forschungs-Projektverwaltung
- Mail Client
- KI-gestützte Projektzusammenfassung

Weitere Punkte, die gewünscht, aber nicht zwingend notwendig sind, umfassen folgende Themen:

- Optimierung der Suchfunktion
- Erweiterte Dashboard-Möglichkeiten
- Inhaltserfassung über Editor

1.4 Organisatorische Rahmenbedingungen

Das Produkt wird im Rahmen einer Bachelorarbeit weiterentwickelt. Pro Teammitglied werden dafür 12 Credits vergeben, das entspricht einem Arbeitsaufwand von rund 360 Stunden. Bei zwei Studierenden ergibt sich so ein Gesamtaufwand von 720 Stunden. Verteilt auf den verfügbaren Zeitraum von 17 Wochen bedeutet das ein Pensum von etwa 21 Stunden pro Woche und Person.

1.5 Abgrenzung

Parallel zu dieser Bachelorarbeit gibt es das Projekt «Publikationsdatenbank», das unabhängig von diesem Projekt läuft. Die Abgrenzung der Projekte wurde bereits im Rahmen der Studienarbeit wie folgt definiert:

Publikationsdatenbank:

Erstellung einer einheitlichen und übersichtlichen Plattform, die speziell für die Veröffentlichung an der OST konzipiert ist. Externe können somit Projekte, Artikel und andere Veröffentlichungen der OST an einem zentralen Ort suchen und einsehen.

OST ResearchNavigator:

Eine Applikation zum Tracking von Projekten an der OST mit besonderem Fokus auf die Sammlung relevanter Metadaten. Diese Applikation dient primär als interne Projektübersicht und -erfassung, soll aber auch ein öffentliches Dashboard bieten.

2 Analyse & Anforderungen

Dieses Kapitel analysiert den Systemkontext und beschreibt die funktionalen sowie auch die nichtfunktionalen Anforderungen des ResearchNavigators. Die Anforderungen werden wie auch schon in der Studienarbeit in die folgenden drei Stufen unterteilt und in der Dokumentation jeweils mit der abgebildeten Farbe gekennzeichnet.

Must Have Features
Could Have Features
Out of Projectscope Features

Tabelle 2.1: Farbcodes der Anforderungen

Must Have Features	Anforderungen, die am Ende des Projekts erfüllt sein müssen.
Could Have Features	Anforderungen, die umgesetzt werden, wenn die Zeit es erlaubt.
Out of Projectscope Features	Anforderungen, die ausserhalb des definierten Projektumfangs liegen und nicht im Rahmen dieser Arbeit bearbeitet werden.

2.1 Funktionale Anforderungen

Die funktionalen Anforderungen an das System werden in diesem Abschnitt anhand von Use Cases erfasst und in einer Abbildung veranschaulicht. Die hier vorgestellten Use Cases, Diagramme und Vorgehensweise basieren auf dem Vorgängerprojekt und wurden im Hinblick auf die neuen Anforderungen angepasst und erweitert.

2.1.1 Akteure

Im Folgenden werden die 5 Hauptakteure der Applikation beschrieben. Diese wurden im Rahmen der Studienarbeit definiert und werden hier vollständigshalber nochmals erläutert. Hervorzuheben ist, dass ein Benutzer der Anwendung diese aus der Perspektive verschiedener Akteure nutzen kann. So kann z.B. jemand gleichzeitig als Prüfer tätig sein und die eingegebenen Projektdaten auf fachliche Korrektheit überprüfen, gleichzeitig aber auch als Mitarbeiter die Möglichkeit haben, eigene Projekte zu erfassen. Die Akteure dienen primär der Beschreibung der Anwendungsfälle (Use-Cases) und sind nicht mit den Rollen im Sinne der Autorisierung zu verwechseln, auf die in späteren Kapiteln noch eingegangen wird.

Akteure	Beschreibung
Besucher	Bezeichnet einen Benutzer, der nicht angemeldet ist. Es kann sich dabei um eine externe Person oder einen Mitarbeiter der OST handeln. Nicht autorisierte Benutzer haben Zugang zu allen öffentlichen Projekten.
Mitarbeiter	Der Mitarbeiter bezeichnet Personen, die Forschungsprojekte veröffentlichen dürfen.
Leiter	Der Leiter ist ein mit besonderen Befugnissen ausgestatteter Mitarbeiter einer Organisationseinheit. Er kann alle Projekte seiner Organisationseinheit verwalten.
Prüfer	Der Akteur Prüfer ist eine Person, die am Prüfungsprozess eines Projekts beteiligt ist. Er überprüft und vervollständigt die Angaben zum Projekt.
Systemadministrator	Der Systemadministrator kann den Benutzern der Applikation ihre jeweiligen Berechtigungen zuweisen und ändern sowie Filterkriterien definieren und verwalten.

Tabelle 2.2: Beschreibung der Akteure

2.1.2 Use Case Diagramm

Das Use Case Diagramm gibt einen Überblick über alle definierten Anwendungsfälle, die im nächsten Abschnitt näher erläutert werden. Um die Übersichtlichkeit zu gewähren, sind im Use Case Diagramm nicht alle Beziehungen vollständig dargestellt. So haben z.B. theoretisch alle Akteure eine Beziehung zu den Use Cases, in denen der Besucher der Hauptakteur ist. Ebenso hätten abgesehen vom Besucher alle Akteure eine Verbindung zum Anwendungsfall Authentifizierung.

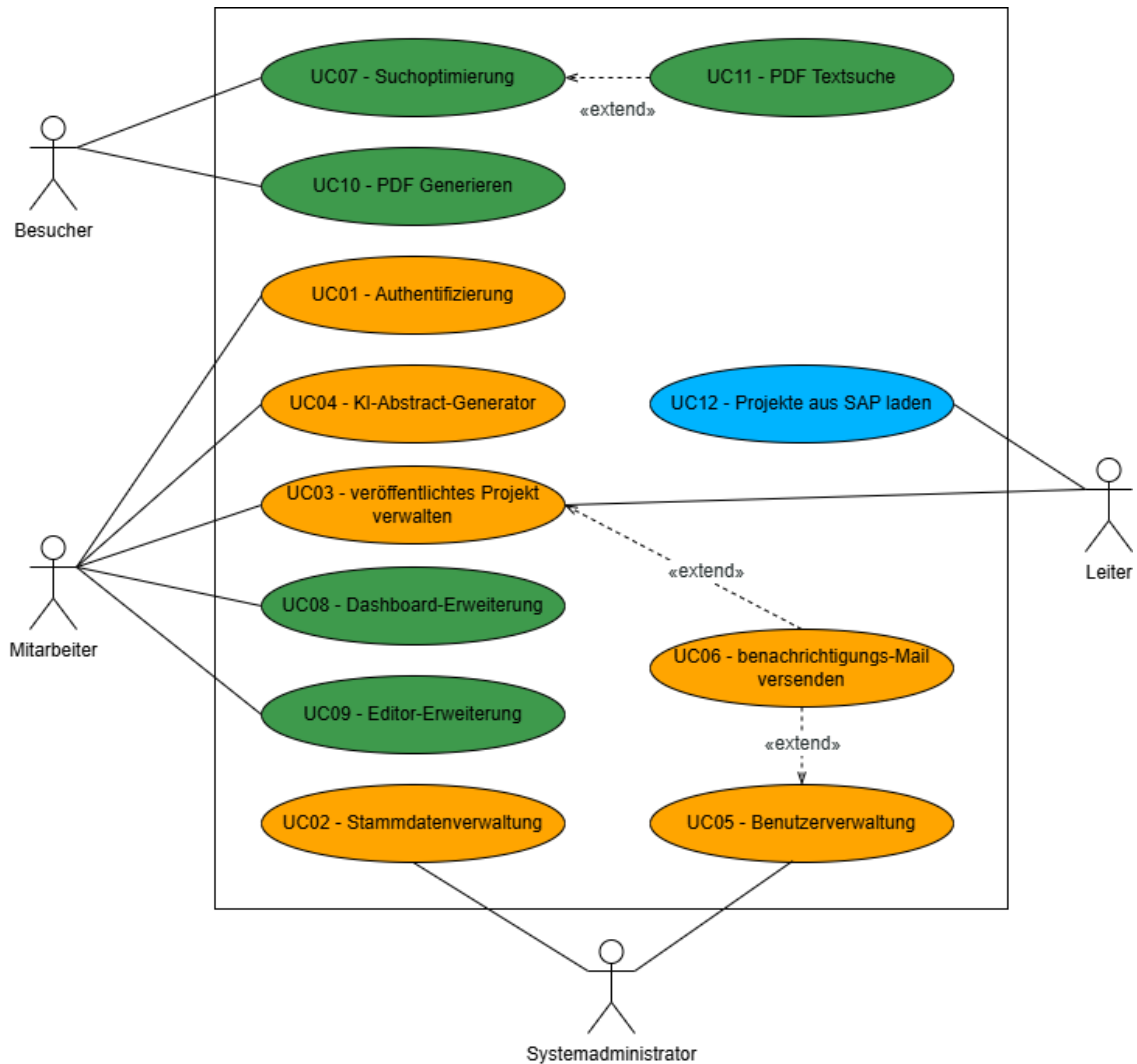


Abbildung 2.1: Use Case Diagramm

2.1.3 Use Cases

Die Use Cases, welche die funktionalen Anforderungen an das System bilden bauen auf dem Vorgängerprojekt auf und werden, wie bereits dort, nach dem Modell von Craig Larman erfasst [2]. Die von Larman empfohlene Vorlage im **Fully Dressed Format**, auf deren Grundlage die einzelnen Use Cases ausformuliert werden, ist jedoch aus Gründen der Übersichtlichkeit und Redundanz um einige Felder gekürzt und leicht abgeändert. Zudem werden Use Cases der Kategorie Out of Projectscope nur anhand ihres jeweiligen Haupterfolgsszenarios beschrieben.

UC01 - Authentifizierung

Must Have

Ein Benutzer kann sich mit seinem GitLab-Konto im System anmelden.

Dieser Use Case wurde im Vorgängerprojekt erfasst und im Rahmen dieser Arbeit detailliert spezifiziert und ausgearbeitet.

Primärer Akteur	Mitarbeiter
Weitere Akteure	Leiter, Prüfer, Administrator
Haupterfolgsszenario	1. Der Benutzer navigiert zur Login-Seite. 2. Auf der Login-Seite werden ein Informationstext sowie eine Weiterleitung zum Login mit GitLab angezeigt. 3. Der Benutzer wird automatisch verifiziert und auf die Startseite weitergeleitet. Im Browser wird die Benutzersitzung in Form eines Cookies persistiert und diejenigen Seiten angezeigt, für welche der Benutzer berechtigt ist.
Alternatives Szenario	Der Benutzer hat noch kein Benutzerkonto. 1. Der Mitarbeiter navigiert zur Login-Seite. 2. Auf der Login-Seite werden ein Informationstext sowie eine Weiterleitung zum externen Authentifizierungsprovider angezeigt. 3. Der Mitarbeiter authentifiziert sich mit seinem GitLab-Benutzerkonto. 4. Der Mitarbeiter nach der erfolgreichen Authentifizierung auf die Profilseite der Applikation weitergeleitet. 5. Der Mitarbeiter erfasst die benötigten Informationen zur Erstellung seines Benutzerkontos. Diese beinhalten unter anderem seinen Vor- und Nachnamen, seine gewünschte Rolle sowie seine Organisationseinheit. 6. Der Mitarbeiter wird darüber benachrichtigt, dass seine Informationen von einem Administrator geprüft werden und er bis dahin noch keine Rechte zum Anlegen von Projekten hat. 7. Dem Mitarbeiter wird der aktuelle Zustand der Administrator-Prüfung im Profil angezeigt.
Detailspezifikationen	• GitLab wird als Authentifizierungsprovider verwendet. • Beim Erstellen eines neuen Benutzerkontos muss ein Mitarbeiter alle erforderlichen Informationen angeben. Diese werden anschliessend durch einen Administrator geprüft (2.7). Erst nach einer erfolgreichen Prüfung kann der Benutzer die Berechtigungen seiner gewählten Rolle nutzen.
Häufigkeit	Hoch

Tabelle 2.3: UC01 - Authentifizierung

UC02 - Stammdatenverwaltung

Must Have

Ein Systemadministrator kann in der Stammdatenverwaltung neue Stammdatensätze erstellen sowie bestehende Einträge bearbeiten und archivieren.

Primärer Akteur	Systemadministrator
Haupterfolgsszenario	1. Der Administrator navigiert zum Admin-Panel. 2. Das System zeigt im Bereich der Stammdaten eine Übersicht aller bearbeitbaren Stammdatensätze. 3. Der Administrator wählt einen bestimmten Stammdatentyp aus, um die zugehörigen Einträge einzusehen. 4. Der Administrator fügt bei Bedarf neue Einträge hinzu oder nimmt Änderungen an bestehenden Einträgen vor. 5. Die vorgenommenen Änderungen werden im System gespeichert.
Alternatives Szenario	Wenn ein zweiter Administrator einen Eintrag speichert, unmittelbar bevor der erste seine Änderungen abgeschlossen hat, wird beim Speichern durch den ersten Administrator eine Fehlermeldung angezeigt. Die Änderungen des zweiten werden nicht überschrieben.
Detailspezifikationen	In der Stammdatenverwaltung können folgende Stammdatensätze erfasst, bearbeitet und archiviert werden: • Arbeitsergebnisse • Schlagwörter • Regionen • Sponsoren • Organisationseinheiten • selbstdefinierte Metadaten
Häufigkeit	Mittel

Tabelle 2.4: UC02 - Stammdatenverwaltung

UC03 - veröffentlichtes Projekt verwalten

Must Have

Ein Leiter kann Projekte, die geprüft und veröffentlicht wurden, verwalten.

Dieser Use Case wurde im Vorgängerprojekt erfasst und ausgearbeitet.

Primärer Akteur	Leiter
Weitere Akteure	Mitarbeiter
Haupterfolgsszenario	1. Der Leiter navigiert zur Ansicht aller Forschungsprojekte seiner Organisationseinheit. 2. Das System zeigt alle Projekte einer Organisationseinheit an. Jedes Projekt hat einen Button, mit dem man in die Bearbeitungsansicht gelangt. 3. Der Leiter klickt auf den Button, um ein Forschungsprojekt zu bearbeiten. 4. Das System zeigt die Bearbeitungsansicht des Projekts mit der Möglichkeit, projektspezifische Informationen zu ändern und zu speichern.
Alternatives Szenario	Handelt es sich beim Akteur um einen Mitarbeiter, so kann dieser nur seine eigenen Projekte verwalten.
Detailspezifikationen	• Projekt kann archiviert werden. • Sichtbarkeit des Projekts sowie Informationen über das Projekt können bearbeitet werden. • Projekte werden nach einer Bearbeitung, je nach Sichtbarkeitsstufe, an die entsprechenden Prüfstellen übergeben.
Häufigkeit	Selten

Tabelle 2.5: UC03 - veröffentlichtes Projekt verwalten

UC04 - KI-Abstract-Generator

Must Have

Ein Mitarbeiter kann automatisch ein Abstract sowie eine Kurzzusammenfassung anhand eines PDF-Dokuments generieren lassen.

Dieser Use Case wurde im Vorgängerprojekt erfasst und im Rahmen dieser Arbeit detailliert spezifiziert und ausgearbeitet.

Primärer Akteur	Mitarbeiter
Weitere Akteure	Leiter
Haupterfolgsszenario	1. Der Mitarbeiter navigiert zur Seite der Projekterfassung. 2. Er navigiert zum Upload-Dialog für Dokumente und lädt dort ein PDF-Dokument hoch. 3. Das System zeigt dem Mitarbeiter eine Schaltfläche an, mit der die Beschreibungen generiert werden können. 4. Der Mitarbeiter betätigt die genannte Schaltfläche. 5. Das System generiert automatisch ein Abstract sowie eine Kurzzusammenfassung und zeigt diese an. 6. Der Benutzer überprüft die generierten Texte und kann diese bei Bedarf anpassen.
Alternatives Szenario	• Wird ein Dokument hochgeladen, aus dem keine Zusammenfassung generiert werden kann (z.B. ein Bild), so wird dem Benutzer die entsprechende Option gar nicht erst angezeigt.
Detailspezifikationen	• Es kann ein Modell eingesetzt werden, das sicherstellt, dass keine internen OST-Daten an Dritte gelangen – beispielsweise ein OST-internes Large Language Model.
Häufigkeit	Mittel

Tabelle 2.6: UC04 - KI-Abstract-Generator

UC05 - Benutzerverwaltung

Must Have

Ein Systemadministrator kann neu registrierte Benutzer überprüfen und deren Benutzerkonten anlegen sowie bestehende Konten und deren Berechtigungen bearbeiten und löschen.

Dieser Use Case wurde im Vorgängerprojekt erfasst und im Rahmen dieser Arbeit detailliert spezifiziert und ausgearbeitet.

Primärer Akteur	Systemadministrator
Haupterfolgsszenario	1. Der Administrator navigiert zur Übersichtsseite der Benutzerverwaltung. 2. Das System zeigt eine Übersicht mit Optionen zur Verwaltung der Berechtigungen bestehender Benutzer sowie der neu registrierten Benutzer, die auf eine Prüfung warten. 3. Der Administrator passt Benutzerrechte an und bestätigt neu registrierte Benutzerkonten.
Alternatives Szenario	• Falls der Administrator fehlerhafte Registrierungsdaten vorfindet, erhält er die Möglichkeit, diese Daten zu korrigieren.
Detailspezifikationen	• Die Berechtigungen sind rollenbasiert vergeben. Es existieren vordefinierte Rollen mit unterschiedlichen Rechten. • Der Administrator kann Rollen einzelnen Benutzern zuweisen oder entziehen. • Die Benutzeroberfläche zeigt dem Administrator klar unterscheidbar an, ob ein Benutzer bereits freigeschaltet wurde oder sich noch im Prüfstatus befindet.
Häufigkeit	Mittel

Tabelle 2.7: UC05 - Benutzerverwaltung

UC06 - Benachrichtigungs-E-Mail versenden

Must Have

Ein Benutzer kann per E-Mail über wichtige Änderungen benachrichtigt werden.

Dieser Use Case wurde im Vorgängerprojekt erfasst und im Rahmen dieser Arbeit detailliert spezifiziert und ausgearbeitet.

Primärer Akteur	Mitarbeiter
Weitere Akteure	Leiter, Prüfer, Systemadministrator
Haupterfolgsszenario	Für diesen Use Case sind diverse Szenarien denkbar, die in den Detailspezifikationen aufgelistet werden. 1. Der Mitarbeiter leitet ein Projekt an den fachlich zuständigen Prüfer weiter. 2. Das System sendet automatisch eine Benachrichtigungs-E-Mail an den Prüfer, um ihn auf das zur Prüfung anstehende Projekt hinzuweisen. 3. Der Prüfer erhält die E-Mail und kann umgehend mit der Überprüfung beginnen.
Detailspezifikationen	E-Mails werden in folgenden Szenarien versendet: • Bei der Weiterleitung eines Projekts zur fachlichen und kommunikativen Prüfung: E-Mail an den fachlich zuständigen Prüfer beziehungsweise an das Postfach der Kommunikationsprüfer. • Wenn ein Projekt im Prüfprozess zurückgewiesen wird: E-Mail an die Person, die das Projekt eingereicht hat. • Wenn ein Projekt veröffentlicht wird: Meldung an Ersteller des Projekts. • Bei der Anlage eines neuen Benutzerkontos: E-Mail an das Postfach der Systemadministratoren (oder an alle Systemadministratoren). • Wenn ein angelegtes Konto aktiviert wird, oder eine Änderungsanfrage am Benutzerprofil angenommen wird: Meldung an entsprechenden Benutzer. • Wenn ein Benutzerkonto gesperrt wird: Meldung an gesperrten Benutzer.
Häufigkeit	Mittel

Tabelle 2.8: UC06 - Benachrichtigungs-E-Mail versenden

UC07 - Suchoptimierung

Could Have

Ein Benutzer kann anhand einer Volltextsuche Projekte anhand der wichtigsten Metadaten finden.

Primärer Akteur	Besucher
Weitere Akteure	Mitarbeiter, Leiter, Prüfer
Haupterfolgsszenario	1. Der Benutzer gibt einen oder mehrere Suchbegriffe in das Suchfeld ein (z.B. „KI“, „Nachhaltigkeit“). 2. Das System führt eine Volltextsuche über Titel, Beschreibung, Abstract, Tags und weiteren Metadaten durch. 3. Eine Ergebnisliste wird angezeigt, sortiert nach Relevanz (Ranking). 4. Der Benutzer sieht pro Projekt die wichtigsten Metadaten (Titel, Kurzzusammenfassung, Sponsor ...). Ist der Suchbegriff in diesen Metadaten vorhanden, wird er zusätzlich farbig hervorgehoben. 5. Der Benutzer wählt ein Projekt aus, um die Detailseite aufzurufen.
Alternatives Szenario	• Falls der Benutzer ein Wort nicht vollständig zu Ende schreibt, sollen Metadaten, die dieses Wort enthalten, dennoch gefunden werden.
Detailspezifikationen	• Die Volltextsuche nutzt PostgreSQLs 'tsvector' und 'tsquery' mit deutschem Konfigurationsprofil und Stemming. • Ergebnisse werden durch eine sinnvolle Gewichtung der Metadaten geordnet. • Trefferstellen im Text werden im Auszug hervorgehoben.
Häufigkeit	Sehr Hoch

Tabelle 2.9: UC07 - Suchoptimierung

UC08 - Dashboard-Erweiterung

Could Have

Ein Leiter kann sich anhand von erweiterten grafischen Auswertungen über Forschungsprojekte an der OST informieren, in diesen filtern und gespeicherte Filter für einen schnellen Zugriff nutzen.

Primärer Akteur	Leiter
Weitere Akteure	Mitarbeiter, Leiter, Prüfer
Haupterfolgsszenario	1. Der Leiter öffnet das Dashboard. 2. Das System präsentiert ihm eine Übersicht mit verschiedenen grafischen Auswertungen und Kennzahlen, die den Umfang der Studienarbeit (Departement, Sponsoren, zeitlicher Verlauf und Themenbereiche) erweitern. 3. Der Leiter wendet Filter an, um die für ihn relevanten Informationen darzustellen. 4. Das System aktualisiert die Visualisierungen entsprechend der gesetzten Filterkriterien. 5. Der Leiter speichert die Filtereinstellungen, um bei späteren Zugriffen schnell die gewünschten Daten abrufen zu können.
Alternatives Szenario	• Falls durch die gesetzten Filter keine Ergebnisse angezeigt werden, informiert das System den Leiter.
Detailspezifikationen	• Es soll eine Auswertung zu den Regionen erstellt werden. • Jeder angemeldete Benutzer verfügt über eine Lesezeichenleiste, in der er seine bevorzugten Filterkriterien speichern kann.
Häufigkeit	Mittel

Tabelle 2.10: UC08 - Dashboard-Erweiterung

UC09 - Editor-Erweiterung

Could Have

Ein Benutzer kann mithilfe eines erweiterten Texteditors das Layout der Beschreibung während der Texterfassung individuell gestalten. Zudem erhält er eine Vorschau des erfassten Projekts.

Primärer Akteur	Mitarbeiter
Weitere Akteure	Leiter
Haupterfolgsszenario	1. Der Mitarbeiter navigiert zur Bearbeitungsseite eines von ihm erstellten Forschungsprojekts. 2. Das System stellt ein Formular bereit, in das der Mitarbeiter die relevanten Informationen und Dokumente zu seinem Projekt eintragen kann. 3. Der Mitarbeiter gibt Inhalte zum Projekt in einen erweiterten Editor ein, der es ihm ermöglicht, die Formatierung individuell anzupassen. 4. Das System speichert die Eingaben und stellt sicher, dass die Darstellung der Inhalte konsistent bleibt. 5. In einer Vorschau kann der Benutzer das erfasste Projekt einsehen.
Detailspezifikationen	• Der Editor unterstützt verschiedene Formatierungsoptionen, darunter Überschriften, Listen und Hervorhebungen.
Häufigkeit	Mittel

Tabelle 2.11: UC09 - Editor-Erweiterung

UC10 – PDF generieren

Could Have

Ein Besucher kann in der Projekt-Detailansicht ein PDF mit Titel und Abstract erzeugen lassen und herunterladen.

Primärer Akteur	Besucher
Weitere Akteure	Mitarbeiter, Leiter, Prüfer
Haupterfolgsszenario	1. Der Besucher öffnet die Detailansicht eines veröffentlichten Projekts. 2. Er klickt auf „PDF generieren“ 3. Das System erstellt ein PDF-Dokument, das den Projekttitel sowie den Abstract enthält. 4. Der Besucher lädt die Datei herunter und kann sie lokal speichern oder weiterleiten.
Alternatives Szenario	• Die PDF-Generierung schlägt fehl. Das System informiert den Besucher mit einer Fehlermeldung.
Detailspezifikationen	• PDF-Format: A4 • Titel als Überschrift, Abstract als Fliesstext. • Mit dem Editor formatierte Inhalte (HTML) werden im PDF korrekt dargestellt. • Funktion steht auch unauthentifzierten Besuchern zur Verfügung.
Häufigkeit	Selten

Tabelle 2.12: UC10 – PDF generieren

UC11 - PDF Textsuche

Could Have

Projekte können anhand von Inhalten gefunden werden, die in den zum jeweiligen Projekt zugehörigen PDF-Dokumenten enthalten sind.

Dieser Use Case wurde im Vorgängerprojekt erfasst und im Rahmen dieser Arbeit detailliert spezifiziert und ausgearbeitet.

Primärer Akteur	Besucher
Weitere Akteure	Mitarbeiter, Leiter, Prüfer
Haupterfolgsszenario	1. Der Besucher navigiert zur Seite der Projektsuche. 2. In dem dafür vorgesehenen Suchfeld gibt er einen Suchbegriff ein. 3. Falls der gesuchte Text in einem der PDF-Dokumente eines Projekts enthalten ist, wird das entsprechende Projekt in den Suchergebnissen aufgelistet.
Alternatives Szenario	• Kein Treffer: Das System zeigt eine „Keine Ergebnisse gefunden“-Meldung
Detailspezifikationen	• Volltext-Indexierung über PostgreSQL mit tsvector. • Aktualisierung des Suchindex beim Hochladen, Löschen oder Austausch eines PDFs.
Häufigkeit	Mittel

Tabelle 2.13: UC11 - PDF Textsuche

UC12 - Projekte aus SAP laden

Out of Project Scope

In SAP erfasste Projekte werden automatisch in die Datenbank der Applikation übernommen. Ein Mitarbeiter des Projekts kann den Eintrag anschliessend vervollständigen.

Dieser Use Case wurde im Vorgängerprojekt erfasst und ausgearbeitet.

Primärer Akteur	Mitarbeiter
Weitere Akteure	Leiter
Haupterfolgsszenario	1. Der Mitarbeiter schliesst ein Projekt in SAP ab. 2. Das System übernimmt dieses Projekt automatisch in die Datenbank der Anwendung. 3. Der Mitarbeiter ergänzt die zur Veröffentlichung erforderlichen Angaben.

Tabelle 2.14: UC12 - Projekte aus SAP laden

2.2 Nicht-Funktionale Anforderungen

Die **Non-functional Requirements (NFR)** sind wie in der Vorgängerarbeit gestaltet. Sie richten sich nach der Norm ISO/IEC 25010, welches die folgenden acht Qualitätscharakteristiken enthält:

- Functional Suitability
- Performance Efficiency
- Compatibility
- Operability/Usability
- Reliability
- Security
- Maintainability
- Transferability

Neben den hier aufgelisteten NFR müssen auch diejenigen aus der vorhergehenden Studienarbeit weiterhin erfüllt sein.

Zusätzlich sind neue Anforderungen für sechs der acht Charakteristiken in Tabellenform definiert.

Das Format wurde ebenfalls aus der Studienarbeit übernommen. Hier zur Vollständigkeit nochmals die komplette Beschreibung:

Die **NFR** erhalten eine ID, bestehend aus dem Prefix "NFR-" und einer Nummerierung. Für die Priorisierung werden die drei Level "Niedrig", "Medium", und "Hoch" verwendet.

Die Anforderungen müssen bis zum angegebenen Fälligkeitsdatum mit den beschriebenen Verifikationstools überprüft werden. Für das Erfüllungslevel werden die Zustände "Nicht erfüllt", "Teilweise erfüllt" sowie "Vollständig erfüllt" eingetragen und mit einer kurzen Beschreibung oder gegebenenfalls mit einem Verweis auf zugehörige Anhänge ergänzt.

ID	NFR-1
Titel	Response Time Stammdatenverwaltung
Anforderung	Performance Efficiency
Beschreibung	Die Response Time für CRUD Operationen auf den Stammdaten im Administrator-Tool ist <2s.
Verifikationstools	Apache JMeter
Priorität	Medium
Definiert am	28.02.2025
Fälligkeitsdatum	07.04.2025
Verifikationsdatum	02.04.2025 / 30.04.2025
Erfüllungslevel	Vollständig erfüllt
Verifikationsdetails	02.04.2025: Mit Apache JMeter wurden wie in Unterabschnitt 5.4.4 beschrieben Tests für alle Stammdatenoperationen ausgeführt. Jeder Durchlauf blieb unter dem Grenzwert von 2s. Screenshots dazu befinden sich im Anhang (Unterabschnitt 9.1.4). 30.04.2025: Die Benutzerverwaltung wurde manuell getestet. Auch hier wurde jede Operation in unter 2s ausgeführt Anmerkung: Mit Einführung der E-Mails wurde die Response Time deutlich erhöht. Dies konnte jedoch durch die Background Queue behoben werden.

Tabelle 2.15: NFR-1: Response Time Projektbearbeitung

ID	NFR-2
Titel	Affordance Anmeldung
Anforderung	Usability
Beschreibung	Vom abgemeldeten Zustand bis zum abgeschlossenen Login inklusive Weiterleitung auf Start- oder Profilseite sind nicht mehr als 2 Buttonklicks nötig. Klicks auf der Seite des OAuth-Providers sind davon ausgenommen. Im Falle einer Neuregistrierung kann das Benutzerkonto auf der Profilseite mit weniger als 5 Klicks vervollständigt werden.
Verifikationstools	Usability Testing mit mindestens 3 potenziellen Benutzern
Priorität	Hoch
Definiert am	28.02.2025
Fälligkeitsdatum	01.04.2025
Verifikationsdatum	28.03.2025
Erfüllungslevel	Vollständig erfüllt
Verifikationsdetails	Für die Anmeldung ist maximal ein Klick auf den Menu-Button sowie ein weiterer auf den Login-Button notwendig. Auf dem Profil muss die Organisationseinheit zwingend ausgewählt werden, alle weiteren Angaben sind optional. Mit einem weiteren Klick wird die Anfrage abgeschlossen und an die Administratoren übergeben. Nach der Bestätigung wird eine Benachrichtigung an die E-Mail-Adresse des Benutzers versendet. Die Affordance des Logins sowie der gesamten Applikation wurde ausserdem in mehreren Usability-Tests mit verschiedenen Benutzern überprüft (Unterabschnitt 5.4.2).

Tabelle 2.16: NFR-2: Affordance Anmeldung

ID	NFR-3
Titel	Authentication
Anforderung	Security
Beschreibung	Das Projekterfassungs- und das Admin-Tool können nur Authentifizierung durch den OAuth-Provider GitLab sowie einer Verifizierung der Benutzerangaben durch einen Administrator verwendet werden. Die Berechtigungen werden mit jeder Backend-Request geprüft.
Verifikationstools	Swagger UI
Priorität	Hoch
Definiert am	28.02.2025
Fälligkeitsdatum	01.04.2025
Verifikationsdatum	30.04.2025
Erfüllungslevel	Vollständig erfüllt
Verifikationsdetails	Alle Endpunkte der API sind nur mit den entsprechenden Rollen verwendbar, welche bei der Anmeldung mit GitLab in einem sicheren Cookie übergeben werden. Jede Änderung der Benutzerrollen bzw. Organisationseinheit muss erst von einem Administrator geprüft werden.

Tabelle 2.17: NFR-3: Authorization

ID	NFR-4
Titel	Test Coverage
Anforderung	Maintainability
Beschreibung	Die Unit Tests weisen eine Branch Coverage von >80% für alle Service-Methoden im Backend auf.
Verifikationstools	XUnit
Priorität	Medium
Definiert am	28.02.2025
Fälligkeitsdatum	30.05.2025
Verifikationsdatum	08.06.2025
Erfüllungslevel	Vollständig erfüllt
Verifikationsdetails	Im Services-Projekt liegt die Branch Coverage bei 88.6% (siehe Coverage Metrics auf GitLab und Unterabschnitt 5.4.1). Damit ist der Zielwert erreicht. Die individuelle Coverage der Services liegt zwischen 83.3% und 100%.

Tabelle 2.18: NFR-4: Test Coverage

ID	NFR-5
Titel	Konfiguration
Anforderung	Transferability
Beschreibung	Zur Vereinfachung der Konfiguration können alle zum Start mit Docker notwendigen Konfigurationen im gleichen env-File vorgenommen werden. Für dieses env-File gibt es eine vollständige Vorlage sowie eine Beschreibung zu allen Variablen.
Verifikationstools	Docker Compose
Priorität	Medium
Definiert am	28.02.2025
Fälligkeitsdatum	30.05.2025
Verifikationsdatum	23.05.2025
Erfüllungslevel	Vollständig erfüllt
Verifikationsdetails	Alle Variablen können im .env-File im Root-Folder spezifiziert werden. Als Beispiel ist ein .env.example vorhanden. Zudem sind alle Variablen im README beschrieben. Docker Compose lädt alle Variablen und gibt diese im korrekten Format an die jeweiligen Services weiter.

Tabelle 2.19: NFR-5: Konfiguration

ID	NFR-6
Titel	Backups
Anforderung	Reliability
Beschreibung	Mindestens alle 24h wird ein Backup der gesamten Datenbank angelegt, welches alle aktuell gespeicherten Projekt-Metadaten sowie alle Benutzerdaten enthält.
Verifikationstools	Docker Desktop
Priorität	Niedrig
Definiert am	28.02.2025
Fälligkeitsdatum	23.05.2025
Verifikationsdatum	01.06.2025
Erfüllungslevel	Vollständig erfüllt.
Verifikationsdetails	Auf dem Server unter dem Pfad /var/lib/docker/volumes/publikationsapp-fuer-forschungsprojekte_pgbackup/, welcher das Volume für den Docker-Container pgbackup enthält, wird täglich um 02:00 Uhr ein gzip-Archiv der gesamten PostgreSQL-Datenbank angelegt. Die Backups können wie im README beschrieben aufgelistet und wiederhergestellt werden.

Tabelle 2.20: NFR-6: Backups

2.3 Nicht Anforderungen

Im Rahmen dieser Arbeit wurden auch Anforderungen identifiziert, die bewusst nicht umgesetzt werden. Diese sogenannten Nicht-Anforderungen betreffen Funktionalitäten oder Eigenschaften, die grundsätzlich denkbar wären, vom Auftraggeber jedoch nicht gefordert wurden oder im Kontext der Zielsetzung als nicht sinnvoll erscheinen. Gründe hierfür können ein unverhältnismässiger Entwicklungsaufwand im Verhältnis zum Nutzen oder potenzielle Einschränkungen für die Nutzerinnen und Nutzer sein.

Solche Anforderungen wurden bereits im Rahmen des Vorgängerprojekts identifiziert und dokumentiert. Die dort aufgeführten Nicht-Anforderungen behalten weiterhin ihre Gültigkeit und werden im Folgenden um zusätzliche Aspekte ergänzt, die im Verlauf dieser Arbeit identifiziert wurden.

2.3.1 Integration weiterer Single-Sign-On-Provider

Eine GitLab-SSO-Integration deckt alle internen Nutzerinnen und Nutzer ab. Die Einbindung zusätzlicher Single-Sign-On-Provider (z.B. Edu-ID, Google oder Microsoft) ist grundsätzlich denkbar, wird im Rahmen dieser Arbeit jedoch nicht umgesetzt. Der erwartete Nutzen steht nicht im Verhältnis zum Entwicklungs- und Wartungsaufwand, zumal im aktuellen Nutzungskontext kein expliziter Bedarf besteht. Eine spätere Erweiterung um weitere Provider bleibt bei entsprechendem Bedarf möglich.

2.3.2 Push-Benachrichtigungen via Desktop-Notifications

E-Mail-Benachrichtigungen decken die relevanten Ereignisse im System zuverlässig ab. Die Implementierung einer zusätzlichen Push-Infrastruktur für Desktop-Benachrichtigungen würde die Systemkomplexität erhöhen, ohne einen signifikanten Mehrwert zu bieten. Aus Sicht des Workflows erweisen sich E-Mails zudem als praktikabler: Nutzerinnen und Nutzer können Benachrichtigungen in ihrem eigenen Tempo bearbeiten und beispielsweise ein Projekt dann prüfen, wenn es zeitlich passt. Daher wird auf die Integration von Desktop-Notifications im Rahmen dieser Arbeit verzichtet.

2.4 Domain Model

Die Domänenanforderungen inklusive Modell wurden aus der Studienarbeit übernommen und an die neuen Anforderungen angepasst, welche in einem Kundenmeeting in Woche 2 definiert und in zwei weiteren in den Wochen 7 und 12 verfeinert wurden.

Ein Benutzer kann neu eine Benutzeranfrage erstellen. Eine solche Anfrage kann sowohl einen neuen Account als auch die Änderung eines bestehenden Accounts beinhalten. Ein Benutzer mit der Rolle Administrator kann diese Benutzeranfragen bearbeiten.

Die Struktur der Organisationseinheiten wurde gemäss dem Kundenwunsch im letzten Meeting der Vorgängerarbeit überarbeitet. Diese entsprechen nun einer Baumstruktur mit beliebig vielen Stufen und werden deshalb mit einer rekursiven Beziehung dargestellt. Die Departemente sind ein spezieller Fall, nach welchem sich einzeln filtern lässt.

Sponsoren können neu einen Nachfolger und dementsprechend auch einen Vorgänger haben. Auch das resultiert in einer rekursiven Beziehung. Der Sponsor bei einem Projekt ist nun optional.

Die Arbeitsergebnisse sind nun ebenfalls in einer eigenen Entität dargestellt, aus welcher eine Filterauswahl getroffen werden kann.

In der Suchanfrage wurden alle Textsuch-Attribute zu einem Attribut Suchtext zusammengefasst, da neu immer in allen Feldern gesucht werden kann. Zudem wurden die Anzahl Suchergebnisse pro Seite und die aktuell ausgewählte Seite zur Suchanfrage hinzugefügt. Angemeldete Benutzer können nun ausserdem im Dashboard ihre Suchanfrage als Lesezeichen abspeichern, um diese zu einem späteren Zeitpunkt schnell wieder abrufen zu können.

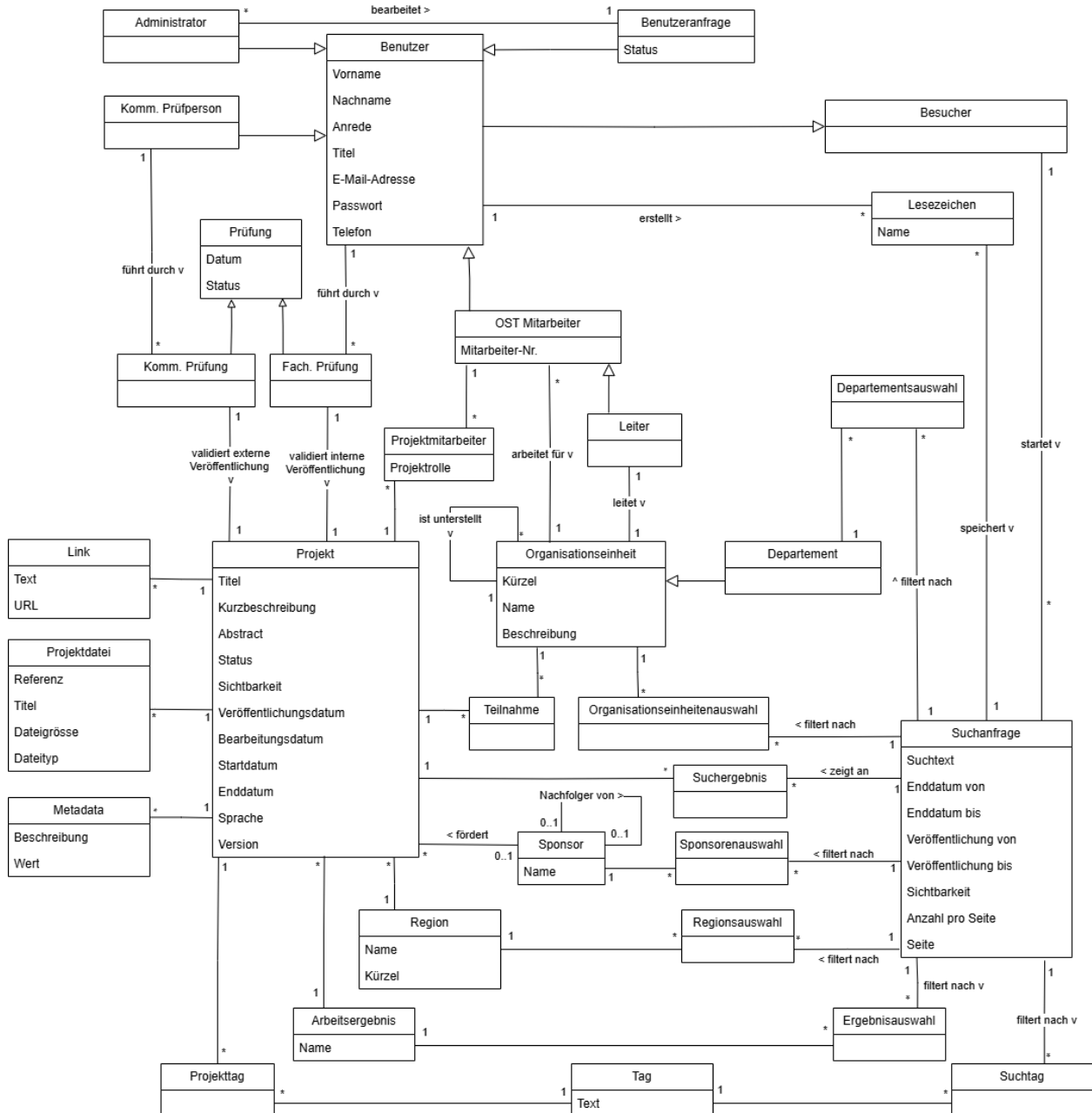


Abbildung 2.2: Domain Model

3 Lösungskonzept

3.1 Analyse und Evaluation

3.1.1 Auswahl der KI-Modelle

Die Beschreibung sowie die Kurzbeschreibung sollen mithilfe von **Künstliche Intelligenz (KI)** aus dem Inhalt einer hochgeladenen Datei generiert werden können. Für die Analyse und Generierung von Texten eignet sich ein **Large Language Model (LLM)**. Es gibt bereits viele Anbieter solcher Modelle mit einer öffentlichen, jedoch meist kostenpflichtigen **Application Programming Interface (API)**. Die Datensicherheit bei externen Anbietern ist ebenfalls ein Kritikpunkt, da der Umgang mit eingegebenen Daten oft intransparent erscheint. Beispielsweise sollten Projektdaten der OST nicht zu lange oder bestenfalls gar nicht auf externen Systemen gespeichert werden. Falls mit den Daten neue Modelle trainiert werden, könnte dies bei ungenügend abgesicherten Modellen durch geschickt angefertigte Anfragen anderer Benutzer zu Datenlecks führen [3].

Bei der wohl verbreitetsten Plattform OpenAI werden API-Anfragen für 30 Tage gespeichert und nicht für das Training verwendet. Auf Anfrage ist eine

Zero Data Retention Policy möglich, bei welcher die Inhalte nicht zwischengespeichert werden [4]. Die Kosten pro Anfrage und die Qualität der generierten Texte variieren sehr stark zwischen den verschiedenen angebotenen Modellen [5].

Das OST-Institut **Institut für Software (IFS)** entwickelt und hostet eine eigene **Plattform mit dem Namen LLMHub**, welche ebenfalls über eine **API** verfügt. Die Plattform bietet verschiedene Modelle an, welche jeweils auf Open-Source-Lösungen basieren [6]. Bezüglich Datensicherheit ist diese Option optimal, da alle Daten innerhalb der OST bleiben. Da die Modelle ständig weiterentwickelt werden, steigert sich auch die Qualität der Zusammenfassungen stetig. Die Performance ist laut ersten Tests ebenfalls ausreichend.

Eine weitere Option ist ein lokal gehostetes Modell, welches auf dem gleichen Server in einem eigenen Container läuft. Der Vorteil dabei ist die vollständige Unabhängigkeit von externen Systemen. Im Rahmen der Analyse wurde die Lösung Ollama [7], welche eine grosse Auswahl an Modellen bietet, in Form eines Docker Containers getestet. Auch Ollama wird über eine API angesprochen. Dabei fiel jedoch schnell auf, dass die Lösung nicht praxistauglich ist. Die Performance ist aufgrund der fehlenden GPU nicht ausreichend, weshalb Antworten der API erst nach über einer Minute oder aufgrund einer Zeitüberschreitung gar nicht eintreffen. Ausserdem ist alleine das Docker Image schon mehr als 1.7GB und jedes Modell ebenfalls noch mehrere Gigabyte gross. Somit ist diese Lösung auszuschliessen, da der zur Verfügung stehende Server weder die Rechenleistung noch den Speicherplatz dafür bietet.

Im Anbetracht der genannten Faktoren Performance, Datensicherheit, Kosten und Qualität wird die OST-interne Lösung LLMHub favorisiert. Da diese jedoch zwischenzeitlich ausfallen kann oder möglicherweise irgendwann abgeschaltet wird, sollte die Einbindung weiterer Anbieter möglichst einfach sein. Als Proof-of-Concept sollen deshalb Lösungen mit OpenAI und LLMHub implementiert werden. Zwischen den Lösungen soll mit einer einfachen Konfiguration gewechselt werden können.

Um die Zusammenfassung betreffend Qualität und Konsistenz zu optimieren, wird gemeinsam mit dem Inhalt der Datei ein Prompt an das LLM gesendet, welcher die Anforderungen an die Zusammenfassung genau beschreibt [8]. Er enthält sowohl inhaltliche Vorgaben als auch klare Anforderungen in Form der minimalen und maximalen Länge sowie der Sprache, in welcher die Zusammenfassung verfasst werden soll. Für die Kurzbeschreibung (Startseite) und die umfassende Beschreibung (Projektdetails) werden unterschiedliche Prompts benötigt.

Des Weiteren soll das scharfe S des deutschen Alphabets in jeder Zusammenfassung durch das Doppel-s der schweizerischen Grammatik ersetzt werden. Dies kann jedoch problemlos in der Applikationslogik umgesetzt werden.

3.1.2 Authentifizierung

Im vorhandenen System können neue Benutzer eine beliebige E-Mail-Adresse eingeben und Rolle sowie Institut auswählen, ohne sich zu authentifizieren. Für den produktiven Einsatz ist eine Authentifizierung zwingend nötig. Neben einem selbst entwickelten Benutzersystem stehen mit Microsoft, Switch edu-ID und GitLab mehrere externe OpenID-Connect Provider zur Verfügung, welche bereits von vielen oder allen OST-Angehörigen verwendet werden. Die Vorteile einer bestehenden Lösung liegen im reduzierten Entwicklungsaufwand, der automatischen Abfüllung von Benutzerinformationen wie dem Namen sowie in der auf dem Know-How von Experten beruhenden und ausführlich getesteten Sicherheit [9] [10].

Für die Anbindung an Microsoft oder Switch muss der Kontakt zu externen Experten mit unklarer Erreichbarkeit gesucht werden, was nicht abzuschätzende Verzögerung zur Folge hätte. Dieses Risiko war angesichts des engen Zeitplans nicht akzeptabel. Somit fiel die Entscheidung auf das System GitLab, welches alle Anforderungen erfüllt und eine einfache Anbindung bietet, die bereits in anderen Projekten der OST zur Verwendung kam. Alle Angehörigen der OST können sich auf der Plattform mit **Single Sign-On** einfach und schnell anmelden und diesen Account dann für den ResearchNavigator verwenden. Ist ein Benutzer bereits bei GitLab angemeldet, wird dieser sofort auf die Ursprungsseite (hier der ResearchNavigator) weitergeleitet.

Neben der plattformspezifischen Benutzer-ID liefert GitLab auch den vollständigen Namen des Benutzers, welcher dann in die Benutzertabelle der angebundenen Applikation abgefüllt werden kann. Weitere Informationen wie Anrede oder Titel sind nicht enthalten, werden aber auch nicht zwingend benötigt.

In gewissen Szenarien, beispielsweise bei Kollaborationen mit anderen Organisationen oder im Falle einer Überprüfung durch externe Experten, benötigen auch nicht der OST angehörige Personen ein Benutzerkonto. Dies ist bei GitLab bereits vorgesehen. Ein Administrator der OST-GitLab-Instanz kann dazu wie in der **GitLab-Anleitung** beschrieben einen neuen Benutzer anlegen.

3.1.3 Abläufe bei der Benutzerverwaltung

Eine benutzerfreundliche und sichere Benutzerverwaltung bildet eine zentrale Grundlage dieser Arbeit. Abbildung 3.1 zeigt den gesamten Prozess mit seinen Zuständen und Übergängen. Wesentlich ist hierbei, dass alle Änderungen an einem Benutzerkonto erst dann wirksam werden, wenn sie von einem Administrator bestätigt wurden. Dadurch bleibt die vollständige Kontrolle über Rechte und Privilegien stets der OST. Selbst öffentlich einsehbare Nutzerdaten, wie etwa Vor- und Nachname im Rahmen einer Projektveröffentlichung, können so nicht ohne Weiteres verändert werden.

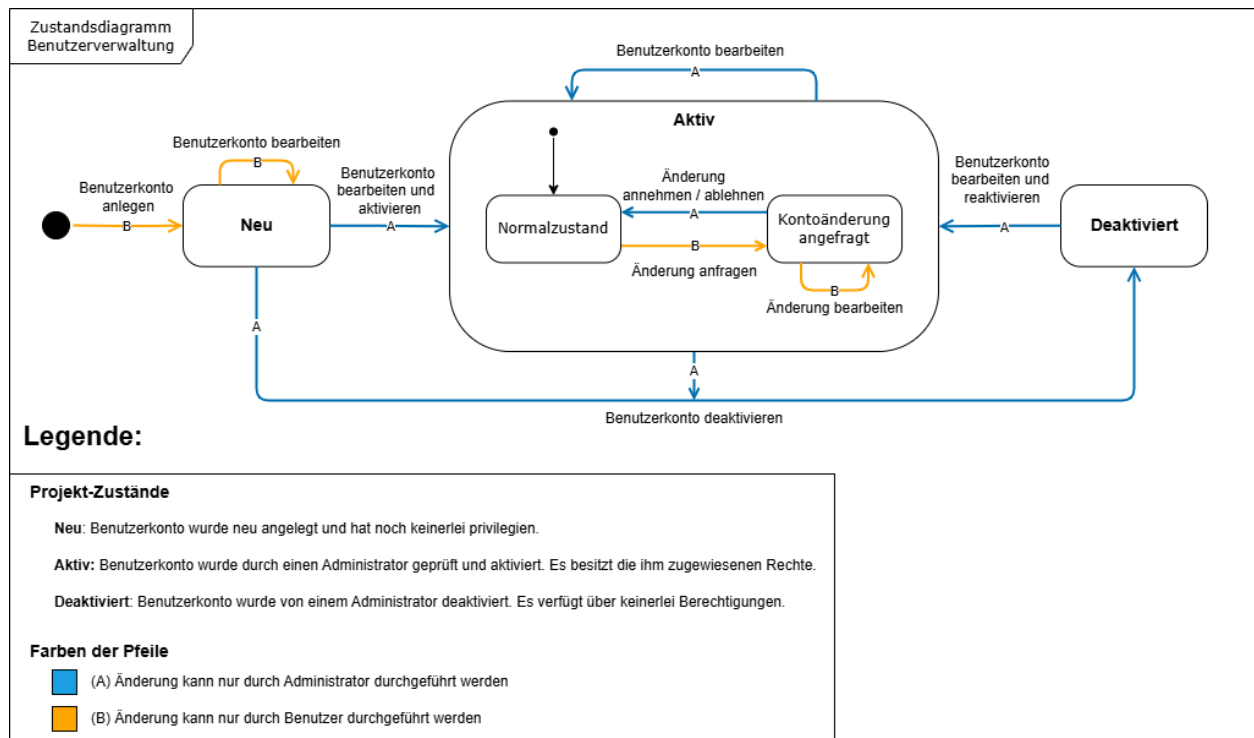


Abbildung 3.1: Ablaufdiagramm Benutzerverwaltung

Wie im Diagramm dargestellt, kann ein neuer Benutzer zunächst eigenständig ein Konto anlegen und seine persönlichen Informationen hinterlegen. Durch diese Selbstverwaltung werden Administratoren entlastet, da sie jeweils nur über die Korrektheit der Daten entscheiden und diese anschliessend bestätigen oder ablehnen müssen. Sollte ein Benutzer bei der Eingabe Fehler gemacht haben, haben die Administratoren jederzeit die Möglichkeit, die Kontodaten manuell zu korrigieren.

Für aktive Konten sieht der Prozess vor, dass Nutzer Änderungswünsche über eine gesonderte Anfrage einreichen können. Auch hier liegt die finale Entscheidung über Annahme oder Ablehnung bei den Administratoren. Zusätzlich sind Administratoren berechtigt, Benutzerkonten jederzeit zu bearbeiten oder, mit Angabe einer Begründung, zu deaktivieren. Ein deaktiviertes Konto kann später wieder reaktiviert werden, beispielsweise wenn ein Mitarbeiter die OST verlässt und zu einem späteren Zeitpunkt zurückkehrt. Dieses Vorgehen sorgt dafür, dass alle Benutzerdaten und Rechte zuverlässig verwaltet werden können.

3.1.4 Suchoptimierung und Rankingalgorithmus

In der bisherigen Version des ResearchNavigators beschränkt sich die Suchfunktionalität auf einfache Abfragen nach Autor, Titel und Abstract. Ziel der Weiterentwicklung ist es, eine umfassendere Suchlösung zu realisieren, mit der Projekte anhand aller relevanten Metadaten gefunden werden können. Im Idealfall soll die Suche sämtliche in der Datenbank vorhandenen Metadaten, von Schlagwörtern und beteiligten Personen bis hin zu Sponsoren und Organisationseinheiten berücksichtigen, um eine möglichst vollständige Ergebnisliste bereitzustellen.

Um die Treffer sinnvoll zu ordnen, wird ein Rankingalgorithmus auf Basis des PostgreSQL tsvector-Datentyps [11] eingesetzt, der bereits im Vorgängerprojekt für die Volltextsuche verwendet wurde. Gefundene Projekte werden dabei in vier Prioritätsstufen (A–D) eingeteilt, wobei A die höchste Relevanz und D die geringste Relevanz darstellt. Gleichzeitig wird auch der Textinhalt hochgeladener PDF-Dokumente durchsucht, jedoch ohne Priorisierung: PDF-Treffer werden erst nach allen Treffern in priorisierten Feldern angezeigt. Die Zuordnung der Metadatenfelder zu den Prioritätsstufen orientiert sich an ihrer erwarteten Aussagekraft.

Metadatenfeld	Prioritätsstufe
Titel	A
Namen der beteiligten Personen	A
Schlagwörter	B
Arbeitsergebnis	C
Sponsor	C
Region	C
Organisationseinheit	C
Durch Administrator konfigurierbare Metadaten	C
Abstract	D
Kurzbeschreibung	D
Volltextinhalt von PDF-Dokumenten	Ohne Priorität

Tabelle 3.1: Priorisierung der Metadaten für den Rankingalgorithmus

Titel und beteiligte Personen (Priorität A)

Der Titel eines Projekts ist in der Regel die prägnanteste und direkteste Zusammenfassung seines Inhalts. Ebenso spielen die beteiligten Personen eine zentrale Rolle: Bei der Suche nach bestimmten Autoren sollen entsprechende Projekte in der Ergebnisliste möglichst weit oben erscheinen.

Schlagwörter (Priorität B)

Schlagwörter geben eine thematische Einordnung der Projekte und ermöglichen es, relevante Inhalte gezielt zu identifizieren. Für den Auftraggeber sind sie von hoher Relevanz. Da sie jedoch weniger spezifisch sind als Titel oder Personennamen, werden sie im Ranking direkt hinter diesen gewichtet.

Arbeitsergebnis, Sponsor, Region, Organisationseinheit und konfigurierbare Metadaten (Priorität C)

Diese Felder liefern wichtige Kontextinformationen, die für bestimmte Suchszenarien relevant sind. Sie werden daher direkt hinter den Metadaten A und B eingeordnet.

Abstract und Kurzbeschreibung (Priorität D)

Freitextfelder wie Abstract und Kurzbeschreibung enthalten ausführliche Informationen, sind jedoch weniger strukturiert. Da sie Phrasen oder Begriffe beinhalten können, die nicht dem Hauptfokus eines Projekts entsprechen, werden sie nur niedrig gewichtet, um irrelevante Treffer in der Ergebnisliste zu minimieren.

Volltextinhalt von PDF-Dokumenten (ohne Priorität)

Die Volltextsuche ergänzt das Ranking durch zusätzliche Treffer aus hochgeladenen PDF-Dokumenten. Diese werden jedoch erst nach den priorisierten Metadatenfeldern berücksichtigt, da der Umfang von PDF-Inhalten tendenziell grösser ist und dadurch die Wahrscheinlichkeit irrelevanter Treffer steigt.

3.1.5 Stammdatenverwaltung

Die Pflege der Stammdaten soll künftig zentral durch Administratoren erfolgen. Zu den verwaltbaren Daten gehören Arbeitsergebnisse, Organisationseinheiten, Regionen, Schlagwörter und Sponsoren. Darüber hinaus soll es möglich sein, neue eigene Metadatentypen anzulegen. Die Tabelle 3.2 zeigt die grundlegenden Funktionen der Stammdatenverwaltung und deren Auswirkungen auf verschiedene Bereiche der Anwendung.

Funktion	Suchfilter	Projekterstellung	Projektdetail
Erfassen	hinzugefügt	hinzugefügt	(kann verwendet werden)
Bearbeiten	umbenannt	umbenannt	umbenannt
Löschen	entfernt	entfernt	(löschar, sofern hier nicht verwendet)
Archivieren	nicht mehr sichtbar	nicht mehr sichtbar	weiterhin sichtbar

Tabelle 3.2: Grundfunktionen der Stammdatenverwaltung

Wie in der Tabelle ersichtlich, können Stammdaten hinzugefügt und anschliessend in der Projekterstellung sowie bei der Projektsuche in den Filtern verwendet werden. Sobald ein Eintrag in einem Projekt verwendet wird, ist ein direktes Löschen nicht mehr möglich. Stattdessen soll der Eintrag archiviert werden. Archivierte Stammdaten sind in den Suchfiltern und der Projekterstellung nicht mehr sichtbar, bleiben jedoch in bestehenden Projektdetails erhalten, um die Nachvollziehbarkeit und Datenvollständigkeit sicherzustellen.

Wird ein bestehender Eintrag bearbeitet, etwa durch eine Umbenennung, wird diese Änderung systemweit übernommen. In der Projekterstellung, den Filteroptionen sowie in allen Projekten, in denen der Eintrag bereits verwendet wurde.

3.2 Architektur

In diesem Abschnitt wird die Architektur des Systems nach Arc42 beschrieben [12]. Einzelne Abschnitte beinhalten aus dem C4-Modell bekannte Diagramme [13]. Die Diagramme wurden aus der Vorgängerarbeit übernommen und gemäss den Architekturanpassungen erweitert

3.2.1 Kontext & Abgrenzung

Neben den technologischen Vorgaben ist die wichtigste Einschränkung auch in diesem Projekt der zeitliche Rahmen. Darum wird auf den bisher verwendeten Technologien aufgebaut, da eine Änderung dieser einen sehr grossen Mehraufwand zur Folge hätte. Die Technologien sind ausserdem immer noch passend und ein Wechsel hätte kaum einen Nutzen. Hier eine Auflistung der wichtigsten technologischen Entscheidungen aus der Vorgängerarbeit:

Backend	Als Schnittstelle dient eine Level 2 RESTful HTTP API, entwickelt mit dem Web-Framework ASP.NET Core und der Sprache C# unter Anwendung einer Adaption des Onion Architecture Pattern.
Datenbank	Als Datenbank wird eine relationale PostgreSQL-Datenbank verwendet, welche mit EF Core Migrations beim Startup der Applikation automatisch generiert und befüllt wird. Für den Zugriff auf die Datenbank wird das Repository Pattern angewendet.
Frontend	Das Frontend wurde mit dem Framework Nuxt.js entwickelt, welches auf Vue.js basiert und die Möglichkeit bietet, die Renderingstrategie speziell anzupassen. Damit ist es besser für Performance-Optimierungen und SEO geeignet. Ausserdem wurde das Framework PrimeVue verwendet, welches viele konfigurierbare UI-Komponenten enthält.
File Storage	Für das Speichern von Projektdateien wird das MinIO Object Storage System verwendet. Es bietet eine S3-kompatible API und ein Package für .NET Core. Im FileRepository wird über einen MinioClient auf das Object Storage System zugegriffen, welches in einem eigenen Docker Container läuft.

Tabelle 3.3: Zentrale Technologien aus der Vorgängerarbeit

In Abbildung 3.2 ist der Kontext des Systems mit den verwendeten Umsystemen anhand eines C4-Kontextdiagramms dargestellt. Die Akteure haben sich nicht verändert, jedoch kommuniziert der OST ResearchNavigator nun mit zwei externen Systemen:

- LLM-Provider: Je nach Konfiguration wird eine Request an die API von LLMHub oder OpenAI gesendet, welche anhand eines Prompts einen mitgesendeten Text zusammenfasst.
- OIDC Identity Provider: Für die Authentifizierung werden GitLab und das Protokoll OpenID Connect verwendet. Benutzer werden für das Login auf GitLab und danach im Erfolgsfall gemeinsam mit einem Cookie zurück auf die ResearchNavigator-Webseite weitergeleitet.

[System Context] OST ResearchNavigator

Business Kontext Diagramm des OST ResearchNavigator

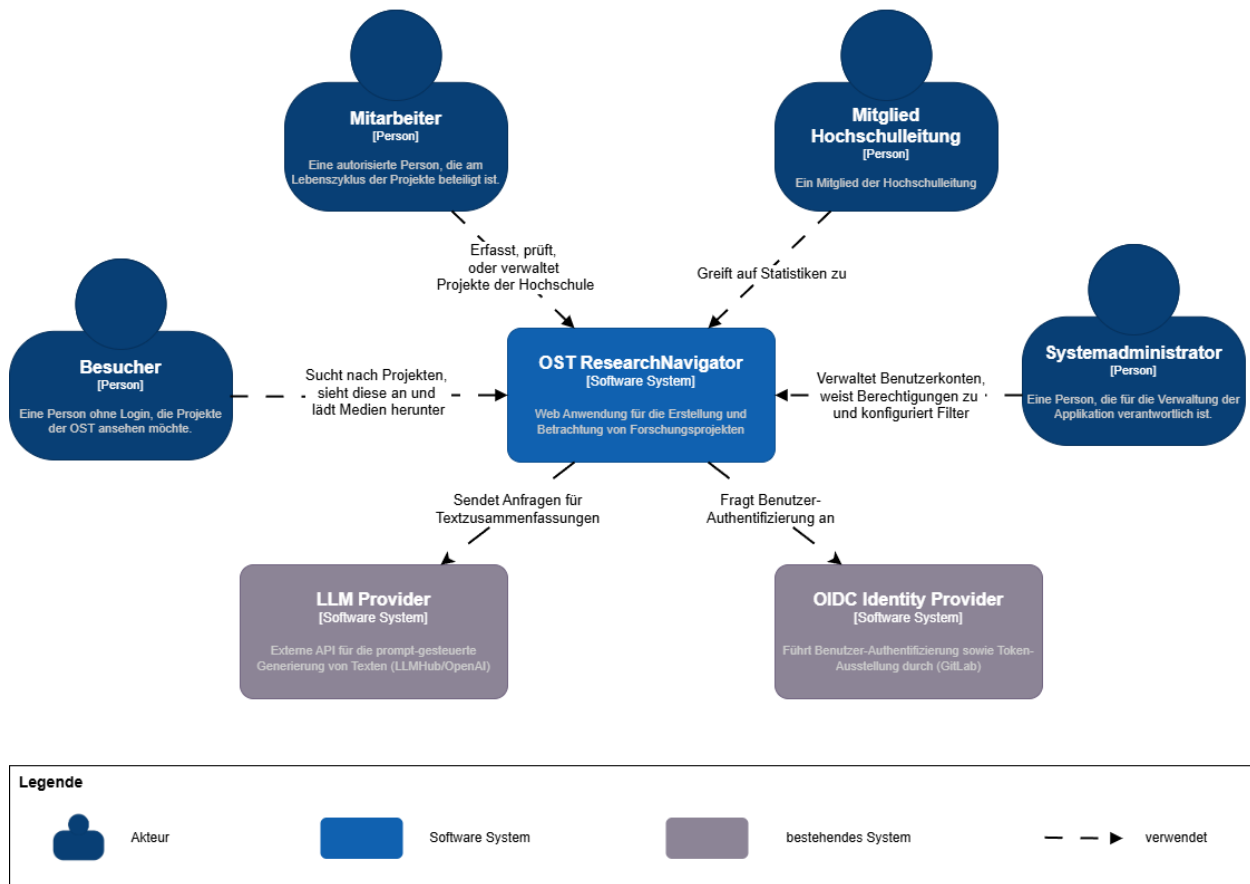


Abbildung 3.2: Kontext Diagramm

3.2.2 Lösungsstrategie

Ziel	Lösungsansatz
Flexibilität: Testdaten	Um die Befüllung der Datenbank mit Testdaten je nach Umgebung anpassen zu können, wurden diese aus den Migrations entfernt und beim Startup der Applikation nur dann in die Datenbank eingefügt, wenn die Einstellung in den AppSettings bzw. dem env-File gesetzt ist. Für das Seeding der Stammdaten wurde ausserdem die neue UseSeeding-Methode aus .NET/EF 9 verwendet.
Usability: Textsuche	Der tsvector zur Volltextsuche soll auch Attribute aus anderen Tabellen enthalten, beispielsweise die Namen der zugehörigen Autoren und Organisationseinheiten oder die Suchtags. Da dies genau wie die Erstellung von Triggers über die EF Core Methoden nicht möglich ist, wird dafür ein SQL-Script verwendet (siehe 4.1).
Wartbarkeit: Datenbank-Migrations	Damit das zuvor erwähnte SQL-Script nicht manuell in einer Migration hinzugefügt werden muss, wird vor jeder Anwendung der Migrationen auf die Datenbank ein Rolldown-Script und danach ein Rollup-Script ausgeführt.
Portabilität: Default-User	Damit der erste Administrator nicht umständlich mit einem direkten Datenbankzugriff freigeschaltet werden muss, kann in der Konfiguration die E-Mail eines Benutzers angegeben werden, welcher beim Startup direkt als Administrator hinzugefügt wird.
Wartbarkeit: Projektstruktur	Die Schichten der Onion-Architektur wurden in 4 eigene Projekte aufgeteilt, um die Verantwortlichkeiten klarer zu trennen und Dependency Inversion zu erzwingen.
Zuverlässigkeit: Backups	Ein Bash Script, welches mittels cron job täglich um 02:00 in einem zusätzlichen Postgres Container ausgeführt wird, legt die gzip-komprimierte Datenbank in einem Backup Volume ab.
Sicherheit: Reverse Proxy	Der einfach konfigurierbare Webserver Caddy wird für Pfad-basiertes Routing und die zentrale Verwaltung von HTTPS-Zertifikaten eingesetzt. Damit wird auch das CORS-Problem gelöst.
Austauschbarkeit: Swagger	Nswag wird verwendet, um aus den Enums und DTOs im Backend automatisch TS-Typen und -Interfaces zu generieren.
Wartbarkeit: State-Container	Im Frontend werden der Zustand des aktuellen Benutzers, des Dashboards und seiner Lesezeichen sowie der Admin-Metrics zentral in Pinia Stores verwaltet, um das Testing zu erleichtern, komponentenübergreifenden Zugriff zu ermöglichen und dabei Reactivity zu erhalten.

Tabelle 3.4: Lösungsstrategie

3.2.3 Bausteinsicht Level 1: Containers

Die Bestandteile des Systems und ihre Kommunikation untereinander und mit externen Systemen sind in Abbildung 3.3 als C4 Container Diagramm [13] dargestellt. Im Vergleich zur Vorgängerarbeit sind neben den unter 3.2.1 erwähnten externen Systemen sind die folgenden Container hinzugekommen:

- Als SMTP Server wird zum Testen ein interner MailDev-Container verwendet. Dieser besitzt eine Weboberfläche, welche alle versendeten E-Mails anzeigt. Für den produktiven Einsatz kann der Container entfernt und die Mail-Konfiguration angepasst werden.
- Für die Generierung von PDF-Dateien aus den HTML-formatierten Abstracts wird die Gutenberg API als Docker Container in das System integriert.

[Container] OST ResearchNavigator

Container Diagramm für die Webanwendung OST ResearchNavigator

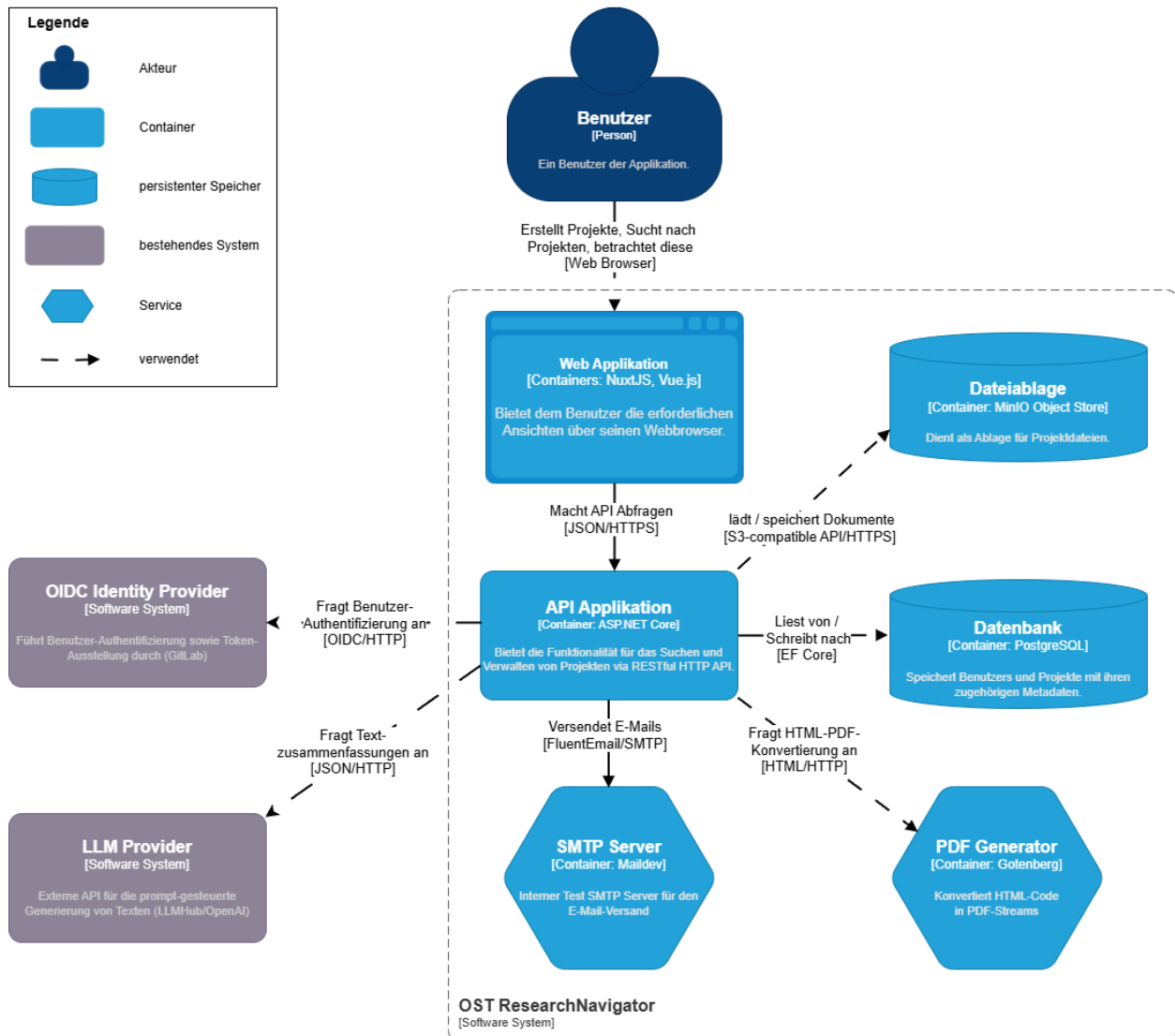


Abbildung 3.3: Container Diagramm

3.2.4 Bausteinsicht Level 2: Components

Der Container „API Applikation“ wird in Form eines C4 Component Diagrams [13] in Abbildung 3.4 im Detail visualisiert. Für die Kommunikation mit den Containern und externen Systemen wird jeweils der entsprechende Client, versteckt hinter einer Abstraktion, aus den Services aufgerufen. Eine Ausnahme bildet der OIDC-Provider; Dieser wird mit einer Authentication Challenge direkt aus dem AccountController aufgerufen und der Benutzer auf die entsprechende Seite (hier GitLab) weitergeleitet.

[Components] API Applikation

Component Diagramm für die OST ResearchNavigator ASP.NET Core Web API

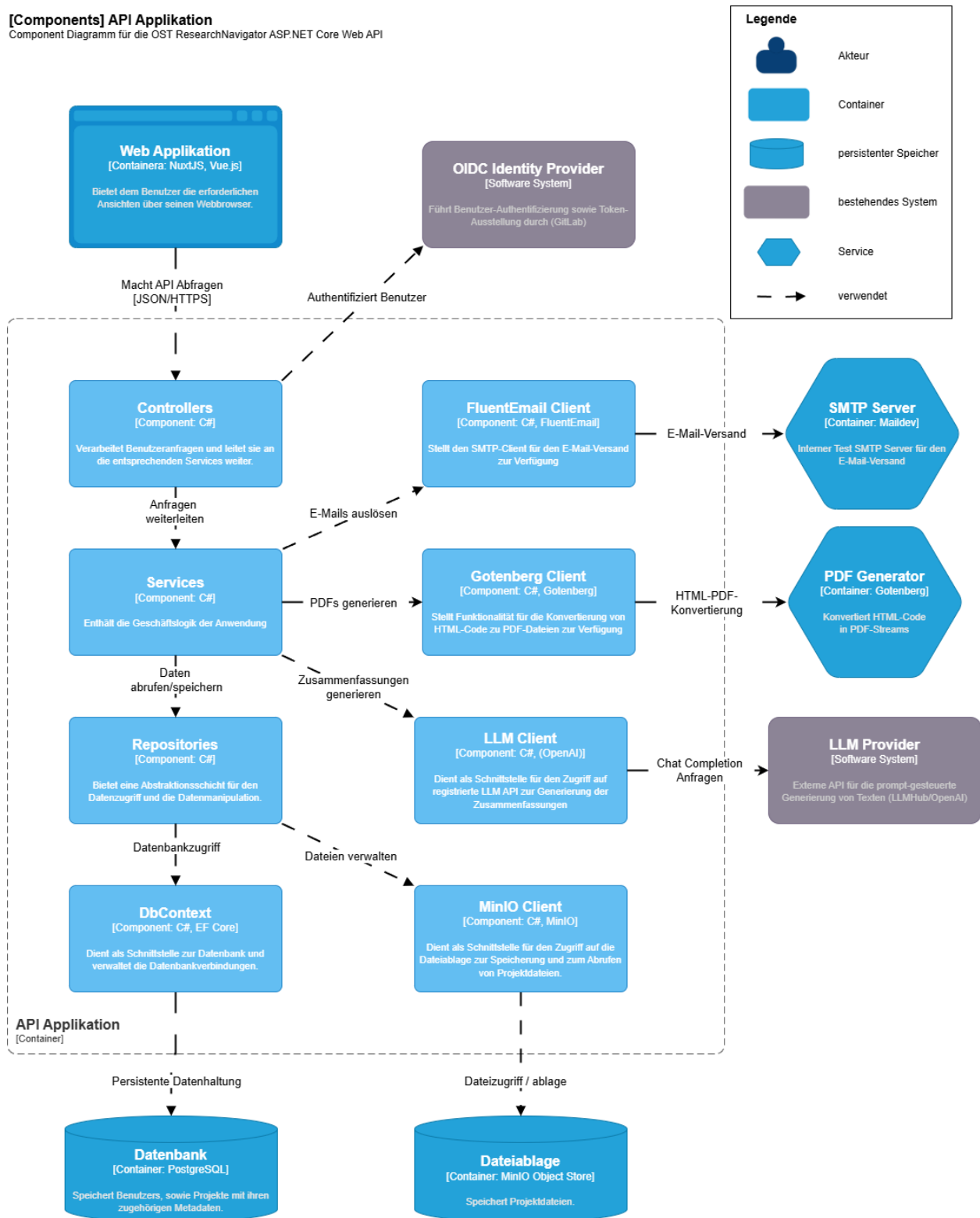


Abbildung 3.4: Component Diagramm

3.2.5 Bausteinsicht Level 3: Code

Backend - Onion Architektur

Das ASP.NET Core Backend aus der Vorgängerarbeit folgt bereits weitgehend der von Jeffrey Palermo konzipierten Onion-Architektur [14] [15], welche der bekannten Clean Architecture von Robert C. Martin [16] ähnelt, sich aber noch mehr auf das zentrale Domain Model und die Trennung der Verantwortlichkeiten (Separation of Concerns) fokussiert [1].

Im bestehenden Code wurden die Schichten in 3 Ordner aufgeteilt. Dabei wird jedoch noch keine strikte Schichtentrennung erzwungen, was zu einigen wenigen Architekturverletzungen führt, beispielsweise die Verwendung von DTOs im Service Layer. Um dies zu verhindern, gibt es neu ein eigenes Projekt pro Schicht, wobei die äusserste Schicht (Presentation & Infrastructure) in zwei Projekte aufgeteilt wird.

In der Abbildung 3.5 ist die Ordnerstruktur abgebildet. Die Projekte sind in der Farbe ihrer Schicht markiert. Einige Unterordner wurden in der Darstellung weggelassen. Zusätzlich ist auch das Unit-Test-Projekt eingezeichnet.

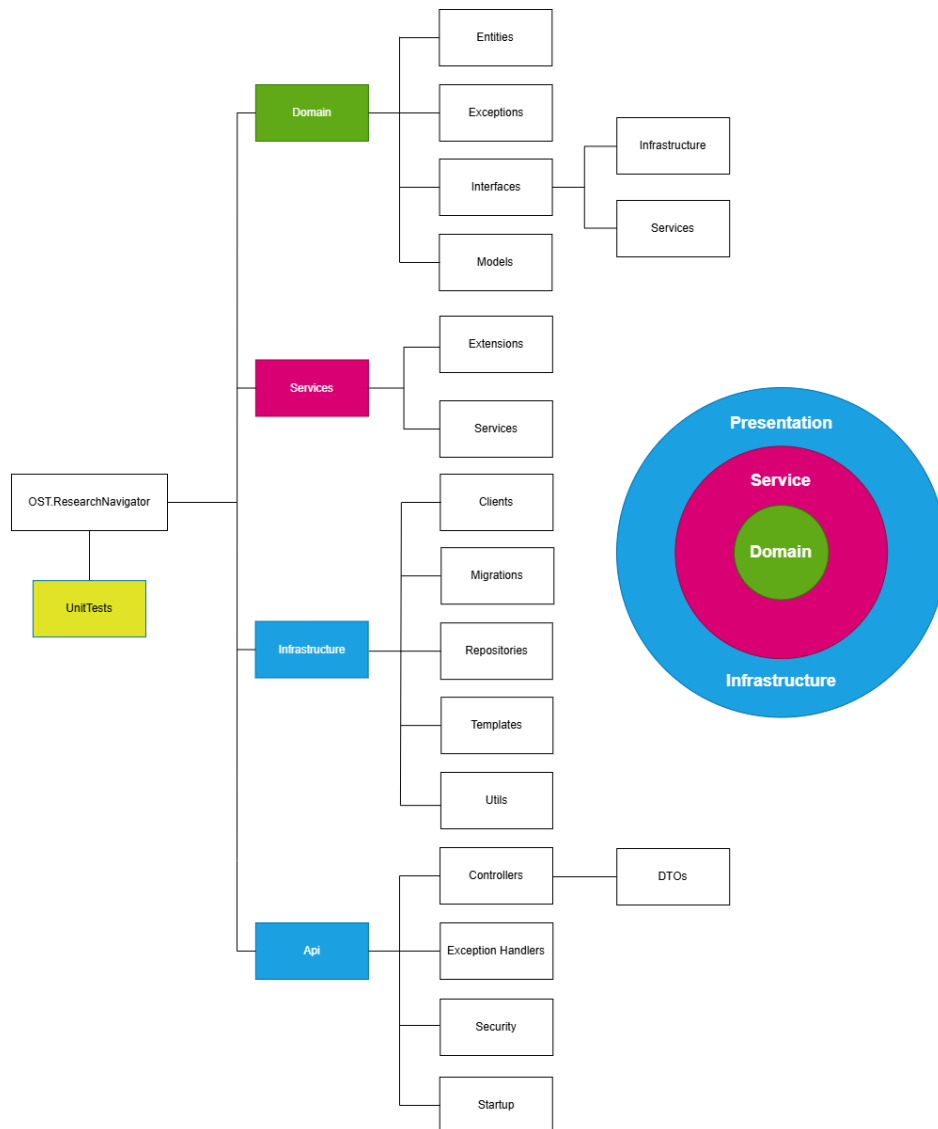


Abbildung 3.5: Onion Architektur

Frontend - Nuxt.js Ordnerstruktur

Die Ordnerstruktur im Frontend entsprach in der Vorgängerarbeit bereits weitgehend der von Nuxt empfohlenen Struktur [17]. Im Folgenden sind die Ordner beschrieben, welche in dieser Arbeit neu dazugekommen sind.

Ordner	Verwendungszweck
constants	Enthält TypeScript-Files, welche Konstanten eigener Typen enthalten. Diese werden hauptsächlich zur Initialisierung verwendet.
i18n	Enthält die Konfiguration der Internationalization-Frameworks i18n mit allen Übersetzungen.
stores	Enthält State Container in Form von Pinia Stores. Diese werden verwendet, um Zustand zu kapseln, aus allen Komponenten darauf zugreifen zu können. Reactivity bleibt dabei erhalten und UI Komponenten werden bei State-Änderungen sofort aktualisiert. [18]
tests	Enthält Unit Tests für alle Stores und Utils sowie vereinzelter Composables. Für die Tests wird das Framework vitest verwendet.

Tabelle 3.5: Erklärung der Ordnerstruktur

Zur Vollständigkeit folgen hier noch die Beschreibungen der weiteren Ordner aus der Studienarbeit.

Ordner	Verwendungszweck
assets	Dient zur Ablage von Ressourcen wie Stylesheets. Hier wird auch das anwendungsübergreifende Theme definiert, welches eine konsistente Gestaltung der Benutzeroberfläche ermöglicht.
components	Ablage von Vue-Komponenten [19]. Diese Komponenten sollen in unserer Applikation zu folgenden Zwecken erstellt/verwendet werden: 1. Zur Wiederverwendbarkeit von Code, um Redundanzen zu vermeiden. 2. Zur Reduktion der Komplexität einzelner Seiten, indem sie in überschaubare Elemente aufgeteilt wird. Die Komponenten können, ähnlich wie HTML-Elemente, durch die Verwendung von Tags in einer Seite eingebunden werden.
composables	Enthält wiederverwendbare Funktionen und Logiken, die mittels der Composition API von Vue 3 erstellt wurden. So können Funktionalitäten in der gesamten Anwendung geteilt werden, was die Wartbarkeit des Codes verbessert.
layouts	Enthält die allgemeinen Layouts der Anwendung. Layouts ermöglichen es, gemeinsame Strukturen wie (in unserem Fall den Header) über mehrere Seiten hinweg zu verwenden, ohne den Code zu duplizieren. Dadurch ein konsistentes Design erleichtert.

Tabelle 3.6: Erklärung der Ordnerstruktur (SA)

middleware	Im middleware-Ordner befinden sich Middleware-Funktionen, die vor dem Rendering einer Seite ausgeführt werden. Mittels Middleware können beispielsweise Authentifizierungsprüfungen und Umleitungen implementiert werden, was dabei helfen kann, den Zugriff auf bestimmte Bereiche der Anwendung zu regulieren.
pages	Enthält die Seiten der Anwendung. Nuxt generiert automatisch Routen basierend auf der Verzeichnisstruktur dieses Ordners.
plugins	Integriert Plugins, durch deren Registrierung zusätzliche Funktionalitäten global verfügbar gemacht werden können. Diese werden von Nuxt automatisch zur Erstellzeit der Anwendung geladen.
public	Enthält statische Dateien, die direkt vom Server ausgeliefert werden sollen. In unserem Fall sind dies Bilder und Fonts. Dateien in diesem Ordner sind über den Root-Pfad der Anwendung zugänglich, was die Verwaltung von öffentlichen Ressourcen vereinfacht.
types	Enthält die TypeScript Typdefinitionen der Anwendung. Durch die zentrale Ablage der Typen kann die Typisierung konsistent gehalten werden. Es ist jedoch nicht zwingend erforderlich, alle Typdefinitionen in diesem Ordner zu speichern. Falls klar ist, dass bestimmte Typen ausschliesslich lokal in einem spezifischen Modul verwendet werden und nicht für andere Teile der Anwendung relevant sind, können diese direkt in der entsprechenden Datei definiert werden. Dies gilt vor allem für weniger komplexe Typen, da die Definition direkt im File die Lesbarkeit und Verständlichkeit des Codes verbessern kann.
utils	Enthält allgemeine Hilfsfunktionen und Utilities, die in verschiedenen Teilen der Anwendung verwendet werden. Dies fördert die Wiederverwendbarkeit von Code und hilft, Redundanzen zu vermeiden. Im Gegensatz zu den composables, die Vue-spezifische Logik und Zustände mit der Composition API bereitstellen, enthalten utils reine JavaScript- oder TypeScript-Funktionen, die unabhängig von Vue sind.

Tabelle 3.7: Erklärung der Ordnerstruktur (SA)

Datenbank - Datenmodell

Die PostgreSQL-Datenbank wird mit EF Core Migrations automatisch aus den annotierten Entitäten und der FluentAPI-Konfiguration im ProjectContext im Backend generiert. Der ProjectContext erbt von IdentityDbContext, wodurch einige Tabellen für die Benutzerverwaltung automatisch generiert werden. Die Entitäten User und Roles erweitern die Standardklassen IdentityUser und IdentityRole, indem sie von diesen erben. Zudem ist noch etwas Konfiguration für die Zwischentabelle UserRoles im ProjectContext nötig.

In der Darstellung 3.6 ist das Entity-Relationship-Diagramm der generierten Datenbank zu sehen. Es erweitert das ERD aus der SA.

Die Tabellen User, Roles, UserRoles und UserLogins haben in der Datenbank den Prefix „AspNet“, welcher hier weggelassen wurde. Roles enthält die 4 Rollen (Administrator, OrgUnitEmployee, HeadOfOrgUnit, UniversityManager), welche zuvor als Enumwerte vorhanden waren. Die Zwischentabelle UserRoles wäre nicht zwingend nötig, da ein Benutzer nur eine Rolle gleichzeitig haben kann, ist jedoch standardmässig im IdentityDbContext enthalten und bietet zudem Möglichkeiten zur späteren Erweiterung der Benutzerverwaltung. UserLogins speichert den Login-Provider, über welchen sich ein Benutzer angemeldet

hat, gemeinsam mit dem Key. In der aktuellen Applikation ist der Provider immer GitLab und der Key die GitLab User-Id. Die Users-Tabelle wurde mit den Attributen GitLab-Id (siehe 3.3) und LoginProvider ergänzt, welche redundant gehalten werden, um die Verwendung des UserManagers zu reduzieren. Der Account-Status kann folgende Werte einnehmen:

- Neu
- Aktiv
- Gesperrt (hier wird die Deaktivierungsbemerkung gesetzt)

Durch die Vererbung der Identity-Klassen enthalten Roles und Users noch diverse weitere Attribute, welche aber im Kontext der ResearchNavigator-Applikation nicht relevant und deshalb nicht aufgelistet sind. Genauso verhält es sich mit den automatisch generierten Tabellen `AspNetRoleClaims`, `AspNetUserClaims` und `AspNetUserTokens`.

Die folgenden weiteren Tabellen sind neu dazugekommen:

- `Bookmarks`: Speichert die Lesezeichen mit den gespeicherten Suchanfragen im Dashboard. Der URL-Hash wird verwendet, um neue Lesezeichen mit bestehenden abzugleichen.
- `UserChangeRequests`: Speichert die Werte, die ein User beantragt. Der Status kann die Werte Pending, Approved und Rejected annehmen. Der comment dient als Begründung für die Ablehnung.
- `ProjectTypes`: Die möglichen Arbeitsergebnisse sollen in der Stammdatenverwaltung angepasst werden können und erhalten deshalb eine eigene Klasse statt eines Enums.
- `Sponsors`: Auch die Sponsoren sollen flexibel geändert werden können. Sie besitzen zudem eine rekursive Vorgänger-Nachfolger-Beziehung.

Die Tabelle `Institutes` wurde zu `OrganisationalUnits` umbenannt, da sie nun unterschiedliche Typen von Organisationseinheiten beinhaltet, welche mit dem Attribut `org_unit_type` unterschieden werden. Weitere Typen sind beispielsweise Studiengänge oder die Departemente, welche keine eigene Tabelle mehr besitzen. Dafür besitzt `OrganisationalUnit` nun eine rekursive Beziehung, mit welcher die untergeordneten Organisationseinheiten in einer Baumstruktur zusammengeschlossen werden können.

Alle Tabellen, deren Datensätze in der Stammdatenverwaltung bearbeitet werden können, besitzen neu ein Attribut `is_archived`. Dies betrifft die folgenden Tabellen:

- `Regions`
- `OrganisationalUnits`
- `SearchTags`
- `Sponsors`
- `Metadata`
- `ProjectTypes`

Neben dem Plain-Text-Abstract wird in `Project` nun auch der HTML-formatierte Abstract in `abstract_html` abgespeichert. Für die Volltextsuche wurden die beiden tsvector-Attribute `search_vector` und `file_search_vector` hinzugefügt. Weitere Informationen dazu sind unter 4.1 zu finden.

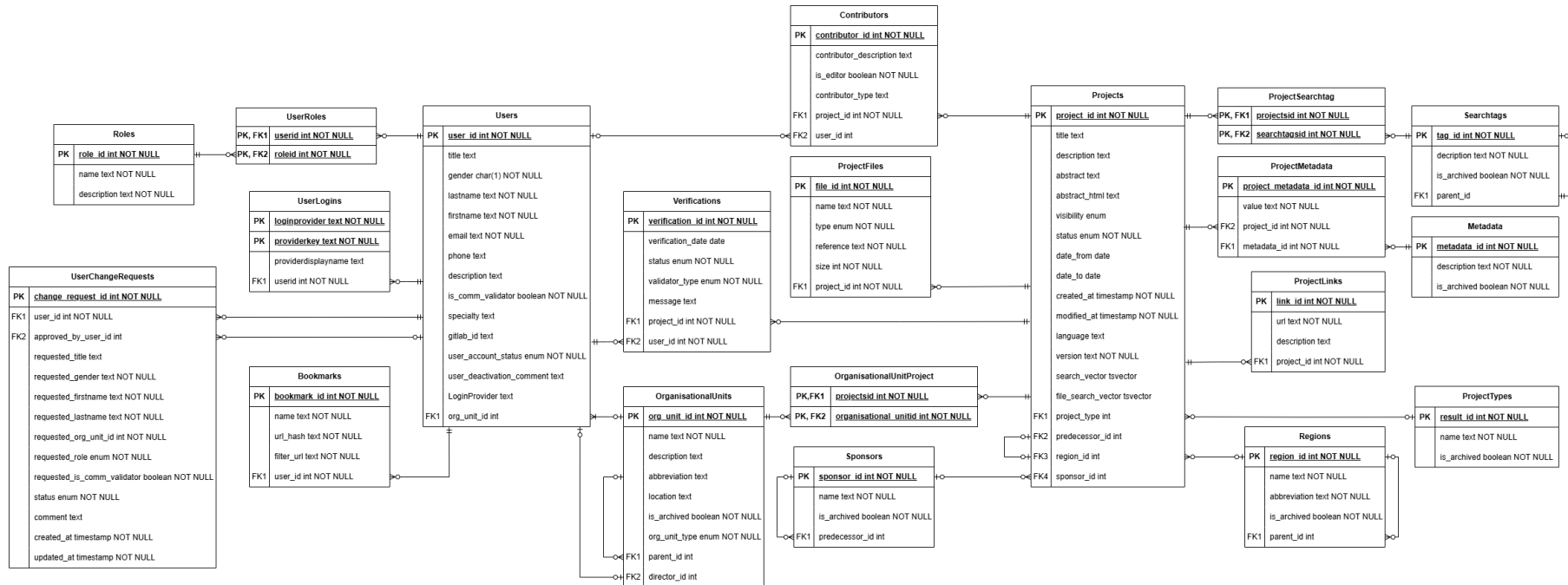


Abbildung 3.6: Entity Relationship Diagram

3.2.6 Laufzeitsicht

In Abbildung 3.7 ist der Ablauf der Benutzerauthentifizierung über den OpenID Connect Identity Provider GitLab in einem Sequenzdiagramm dargestellt. Beim Aufruf des Endpoints „`oauth/GitLab`“ wird der Benutzer zur GitLab-Anmeldung umgeleitet. Nach erfolgreicher Anmeldung stellt GitLab einen Access-Token, den Namen, die E-Mail und die ID des Benutzers bereit. Mit dem `SignInManager` wird ein Cookie erstellt, welches die User-Claims (E-Mail, Rolle, Organisationseinheit) enthält. Dieses wird bei allen Requests mitgesendet und zur Autorisierung genutzt.

Die Authorize- und Token-Endpoints liest ASP.NET Core mittels Discovery-Mechanismus aus der OpenID-Konfiguration des Providers [20] [21]. Die Authority, also die Basis-URL des Providers (hier <https://gitlab.ost.ch>) wird aus der Konfiguration gelesen, genau wie die `ClientId`, das `ClientSecret` und der `CallbackPath`, an welchen GitLab den Client nach der Authentifizierung weiterleitet (hier „`/api/signin-gitlab`“).

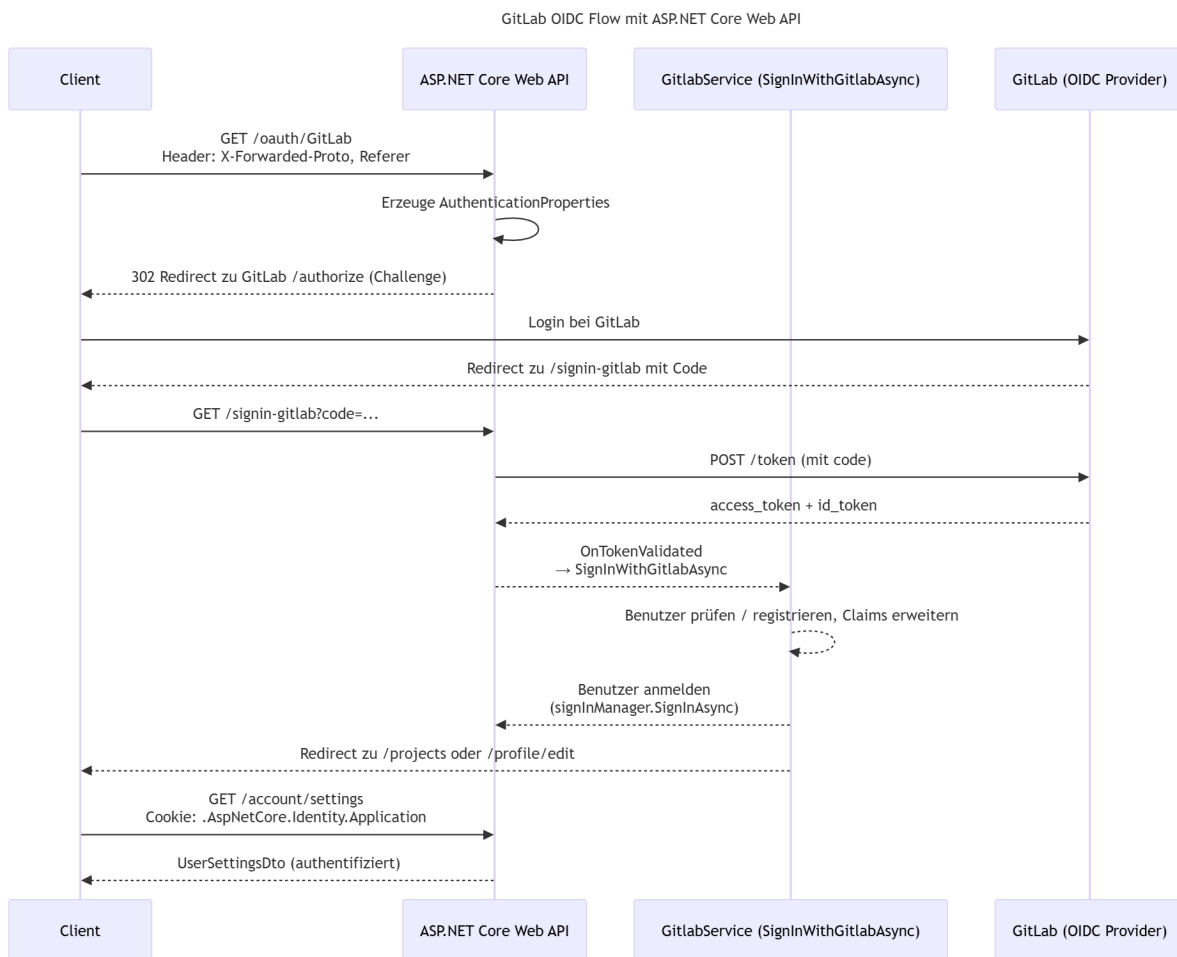


Abbildung 3.7: OIDC Authentication Flow zwischen Client, ASP.NET Core Web API und GitLab

Codeausschnitte und Beschreibungen des Setups und des `GitlabService` sind unter 4.2 zu finden.

3.2.7 Deployment-Sicht

Das in Abbildung 3.8 dargestellte C4 Deployment Diagramm [13] wurde mit den neuen Containern und den externen Systemen ergänzt. Die Docker Container laufen weiterhin auf einem virtuellen Server im internen Netzwerk der OST.

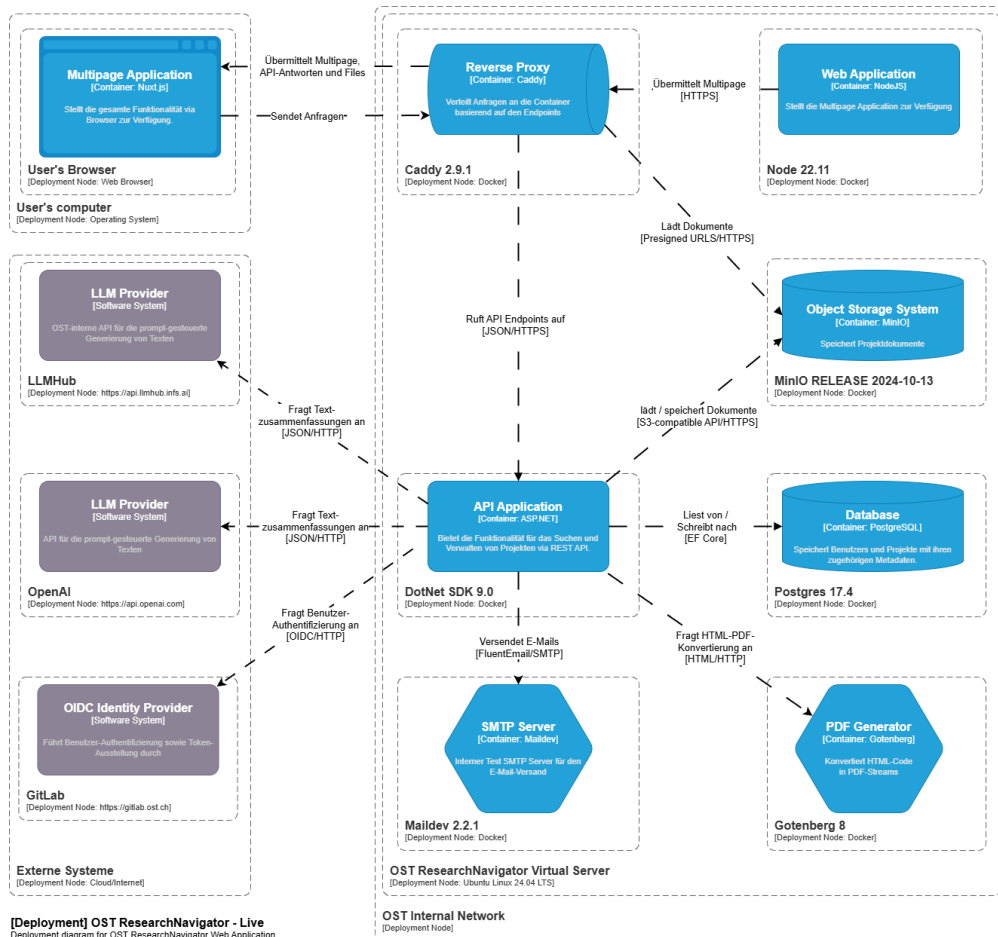


Abbildung 3.8: C4 Deployment Diagramm

Konkrete Informationen zum Live-System (entsprechen grösstenteils Deployment der SA)

- Link zur Webanwendung: **OST ResearchNavigator**
- Nur innerhalb von OST Netzwerk erreichbar
- OS: Linux Ubuntu 6.8.0-47-generic
- Hostname: srbsci-127.ost.ch
- IP-Adresse: 10.8.36.127
- Port für Webanwendung: 3000 (auch mit Port 80 erreichbar)
- Port für API: 5050 (auch mit Port 80 über Pfad /api erreichbar)
- Port für Minio GUI: 9001
- Port/Pfad für Maildev-Dashboard: 1080/maildev

- VM läuft bis: 31.08.2025

Hardware des Servers:

- 2 vCPU (empfohlen)
- 4 GB RAM (empfohlen werden jedoch 8GB, da die Docker Container den Arbeitsspeicher stark auslasten)
- 50 GB Disk Drive (dies ist für das Deployment alleine knapp ausreichend, muss jedoch noch je nach Anzahl Projekten, Dateien pro Projekt sowie durchschnittlicher/maximaler Dateigrösse hochskaliert werden)
- Eine Verfügbarkeit von >95% insgesamt und >99% während den Arbeitszeiten der OST sollte gewährleistet sein.

Eine Schritt-für-Schritt-Anleitung für das Deployment befindet sich im Anhang unter [9.2.3](#).

3.2.8 Querschnittliche Konzepte

Konfiguration

Um eine klare Trennung von Code und umgebungsspezifischen Einstellungen zu erreichen, erfolgt die Konfiguration der Umgebung über Environment-Variablen, die zentral in einer .env-Datei gepflegt werden. Beim Deployment werden diese Werte über Docker und docker-compose an die jeweiligen Container übergeben. Durch diesen Ansatz bleibt die Anwendung flexibel, lässt sich leicht an verschiedene Umgebungen anpassen und ist einfacher wartbar. Konfigurationen, welche sich seltener ändern und nicht kritisch sind, werden im Backend in der Datei appsettings.json im Projekt OST.ResearchNavigator.Api gesetzt.

Autorisierung

Um die Zugriffskontrolle feingranular zu steuern, werden vier verschiedene Benutzerrollen verwendet, die jeweils über unterschiedliche Berechtigungen verfügen. Das Benutzerprofil sowie die zugehörige Benutzerrolle werden im Frontend in einem Pinia-Store, dem UserStore, gespeichert. Für jede Seite im Frontend wird definiert, welche Rollen für einen Zugriff erforderlich sind. Beim Aufruf einer Seite überprüft eine Middleware, ob der aktuell angemeldete Benutzer über die benötigte Rolle verfügt, indem sie diese mit der im UserStore hinterlegten Rolle vergleicht.

Diese clientseitige Prüfung dient jedoch lediglich der Benutzerführung und ist allein nicht ausreichend, um einen sicheren Zugriffsschutz zu gewährleisten. Daher werden im Backend alle nicht-öffentlichen API-Endpunkte zusätzlich auf Rollenbasis geschützt. Dies geschieht mithilfe der [Authorize(Roles = ...)]-Annotation von ASP.NET Core, die sicherstellt, dass nur authentifizierte Benutzer mit einer entsprechenden Rolle Zugriff auf die jeweilige Methode oder den Controller erhalten.

3.2.9 Architekturentscheidungen

PDF Text Extractor

Datum: 18.04.2025

LLMHub unterstützt keine Datei-Uploads, weshalb für die Generierung der Zusammenfassungen aus PDF-Dateien erst der Text extrahiert werden muss. Dafür wurden verschiedene Libraries in Betracht gezogen:

- PdfPig [22]
- iText7 [23]
- PdfSharp [24]

Die Library iText7 sah auf den ersten Blick vielversprechend aus, ist jedoch unter [GNU Affero General Public License \(AGPL\)](#) lizenziert. PDFsharp legt den Fokus eher auf die Erstellung von PDFs, die Textextraktion ist nur in begrenztem Umfang möglich. PdfPig überzeugte bei ersten Tests durch die sehr einfache Einbindung und Verwendung in .NET. Die Ergebnisse waren ebenfalls zufriedenstellend. Somit war die Entscheidung für PdfPig eindeutig.

Status: Angenommen

E-Mail Library & SMTP Server

Datum: 01.05.2025

Für das Versenden der Benachrichtigungs-E-Mails gibt es in .NET verschiedene Möglichkeiten. Die Klasse System.Net.Mail bietet die grundlegende Funktionalität für den Mailversand, wird jedoch von Microsoft selbst nicht mehr empfohlen. Darauf aufbauend gibt es mehrere Libraries. Zwei der grössten sind FluentEmail [25] und MailKit [26]. Beides sind gute Optionen. Die Entscheidung fiel auf FluentEmail aufgrund der einfachen Verwendung des FluentEmail-Clients und der zugehörigen Razor-Templates. Ausserdem wurde FluentEmail bereits in uns bekannten Projekten an der OST verwendet, weshalb ein gewisses Know-How und Beispiele bereits vorhanden waren.

Im FluentEmail-Client muss ein SMTP-Server konfiguriert werden. Da in dieser Projektarbeit kein produktives Deployment enthalten ist und somit auch keine E-Mails tatsächlich versendet werden müssen, reicht ein Test-SMTP-Server in Form eines Maildev-Docker-Containers, welcher in der Konfiguration später mit einem echten SMTP-Server ausgetauscht werden kann.

Status: Angenommen

BackgroundTaskQueue für E-Mail-Versand

Datum: 13.05.2025

Der E-Mail-Versand führte zu deutlich längeren Antwortzeiten. Durch die Einführung einer BackgroundTaskQueue [27] können die E-Mails im Hintergrund versendet werden, wodurch die Antwortzeiten nicht beeinträchtigt werden. Nachteile sind die erhöhte Komplexität und das schwierigere Debugging.

Es wurde entschieden, dass eine BackgroundTaskQueue verwendet wird, da die erheblichen Performance-Unterschiede die Usability der Applikation stark beeinflussen.

Status: Angenommen

Code-Ausschnitte mit Beschreibungen dazu sind unter [4.6](#) zu finden.

PDF Generator

Datum: 20.05.2025

Die Generierung eines PDFs, welches Titel und Beschreibung eines Projektes enthält, kann ebenfalls mit verschiedenen Lösungen umgesetzt werden. Zuerst wurde erfolglos nach einer Open-Source-Library gesucht, welche sowohl die Extraktion als auch die Generierung übernimmt.

Danach wurden verschiedene Libraries ausprobiert. QuestPDF [\[28\]](#) überzeugte durch eine funktionsreiche und gut dokumentierte FluentAPI, hatte jedoch keine HTML-Funktionalität, um die formatierten Abstracts zu konvertieren. Die Libraries PDFsharp [\[24\]](#) und DinkToPdf [\[29\]](#) führten zu Kompilierfehlern und grossen Problemen beim Setup.

Die einzige funktionierende Lösung war Gotenberg [\[30\]](#), eine API welche in einem eigenen Docker-Container läuft und diverse Dateiformate konvertieren kann, darunter auch HTML zu PDF. Die Kommunikation funktioniert über den GotenbergSharpClient, welcher von der .NET Library zur Verfügung gestellt wird. Zur Gestaltung werden ausserdem HTML-Templates mit CSS-Code verwendet. Die CSS-Funktionalität von Gotenberg ist allerdings beschränkt, weshalb für das Styling einige Workarounds nötig waren.

Status: Angenommen

3.3 Technische Schulden

Titel	Beschreibung
EF Core in Domain	Durch die Annotations in den Entity-Klassen und den Postgres-Datentyp NpgsqlTsVector ist das Domain-Projekt abhängig von EF Core und PostgreSQL. Dies könnte durch Auslagerung der Konfiguration in die ProjectContext-Klasse mittels FluentAPI gelöst werden.
GitLab in Domain	Die User-Klasse enthält ein Attribut mit dem Namen GitlabId, welches bei der Anmeldung auf die Gitlab-UserId gesetzt wird. Falls der Provider gewechselt werden sollte, ist der Name nicht mehr korrekt. Das Attribut kann einfach in ProviderKey umbenannt werden, es bestehen keine Abhängigkeiten. Das Detail wurde nicht mehr korrigiert, da es erst kurz vor dem Abschlusstermin auffiel.
Projektsuch-Query	Wie bereits in den technischen Schulden der SA beschrieben, ist die Methode CreateProjectsSearchQuery ziemlich lang. Auch die Authorization-Logik befindet sich noch immer im selben Repository. Diese wurde nicht herausgelöst, da die Autorisierungsfiler entscheidend für die Gesamtanzahl der gefundenen Projekte sind, welche nur innerhalb des Repository über das IQueryable abgefragt werden kann.
i18n Type Checking	Die Verwendung fehlender Übersetzungen im Code wird nicht überprüft. Die Implementation von i18n type-safety in Nuxt.js führte zu Problemen, wodurch der Aufwand zu gross wurde.
Übersetzungen Backend	Exception-Messages im Backend werden nicht übersetzt. Sie sind teilweise in Englisch, teilweise in Deutsch gehalten. Bei den englischen Fehlermeldungen werden jedoch in den meisten Fällen im Frontend Nachrichten aus der i18n-Config angezeigt.
Grosse Vue-Components	Gewisse Components, beispielsweise die FilterSidebar, sind noch über 300 Zeilen lang. Der Aufwand, diese weiter aufzuteilen, war zu gross. Ausserdem bestehen diese auch zu einem Grossteil aus Template- und CSS-Code.

Tabelle 3.8: Technische Schulden

Die technischen Schulden aus der SA konnten abgesehen von der erwähnten Ausnahme vollständig abgearbeitet werden.

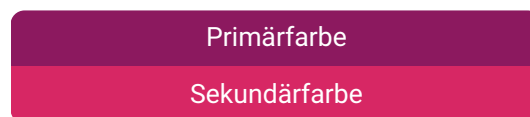
3.4 Design

In diesem Abschnitt wird das Design der Anwendung im Detail erläutert. Bereits im Rahmen der vorangegangenen Studienarbeit wurden ein allgemeines Konzept entwickelt sowie Richtlinien zum Umgang mit dem Thema Accessibility definiert. Für die vorliegende Arbeit wurden diese Grundlagen übernommen, überarbeitet und durch zusätzliche Aspekte ergänzt, um den spezifischen Anforderungen dieses Projekts gerecht zu werden.

3.4.1 Anforderungen und Allgemeines Konzept

Das Design der Anwendung spielt eine zentrale Rolle, da die Applikation öffentlich zugänglich sein wird und den Standard einer modernen technischen Hochschule widerspiegelt. Es soll ein professioneller und vertrauenswürdiger Eindruck vermittelt werden. Der Schwerpunkt liegt auf einem klaren und übersichtlichen Design, um die Benutzerfreundlichkeit zu maximieren.

Farblich orientiert sich das Design an den corporate-Farben der OST, um eine einheitliche visuelle Identität zu gewährleisten. Zum Einsatz kommen definierte Primär- und Sekundärfarben, wie in der folgenden Übersicht dargestellt:



Um die Anwendung auf unterschiedlichen Geräten und Bildschirmgrößen optimal nutzbar zu machen, wird ein responsives Design umgesetzt. Zur visuellen Unterstützung der Funktionalitäten werden Icons eingesetzt. Diese folgen einer konsistenten Stilrichtung und tragen zu einem ansprechenden Gesamtbild bei. Es ist wichtig, eine intuitive und benutzerorientierte Bedienung zu gewährleisten. Dazu wird das Feedback von Auftraggebern, Testpersonen und anderen Interessierten aktiv miteinbezogen. Bei der Gestaltung werden aktuelle Designtrends berücksichtigt, um ein modernes und zeitgemässes Erscheinungsbild zu schaffen.

3.4.2 Accessibility

Barrierefreiheit, auch als Accessibility bezeichnet, ist ein zentraler Aspekt moderner Softwareentwicklung. Ziel ist es, digitale Anwendungen und Inhalte so zu gestalten, dass sie von allen Menschen unabhängig von ihren individuellen Fähigkeiten oder Einschränkungen genutzt werden können. Die wichtigsten Leitlinien für barrierefreie Webanwendungen werden durch die [Web Content Accessibility Guidelines \(WCAG\)](#) des [World Wide Web Consortium \(W3C\)](#) definiert. Die Leitlinien basieren auf vier Prinzipien [31]. Diese sind in nachfolgender Tabelle festgehalten, sowie auch Massnahmen für die Entwicklung dokumentiert, um diese so weit wie möglich zu erfüllen:

Prinzip	Massnahme
Wahrnehmbarkeit: Informationen und Bestandteile der Benutzerschnittstelle müssen den Benutzern so präsentiert werden, dass diese sie wahrnehmen können.	• Für Bilder werden Alternativtexte angeboten • Schaltflächen beinhalten klare Beschreibungen, Informationen werden nie nur über Farben dargestellt • Wichtige Schaltflächen haben einen ausreichend hohen Farbkontrast
Bedienbarkeit: Bestandteile der Benutzerschnittstelle und Navigation müssen bedienbar sein.	• Webseite ist auch rein über die Tastatur bedienbar • Fokus-Zustand interaktiver Elementen ist sichtbar
Verständlichkeit: Informationen und die Bedienung der Benutzerschnittstelle müssen verständlich sein.	• Nutzen des i18n-Internationalisierungspakets, um Inhalte einfach auf weitere Sprachen erweitern zu können • Verständliche und konkrete Eingabefehlermeldungen • Einsatz von Tooltips und Dialogfenstern mit klaren Erklärungen • Einsatz von Hinweisen und Mitteilungen (z.B. „Änderungen wurden nicht gespeichert“)
Robustheit: Inhalte müssen robust genug sein, damit sie von einer grossen Auswahl an Benutzeragenten interpretiert werden können.	• Responsives Design • Kompatibilität mit unterschiedlichen Browsern

Tabelle 3.9: Accessibility Massnahmen

3.4.3 Mockups

Mockups werden gezielt eingesetzt, um neue Seiten zu visualisieren und das Design mit dem Auftraggeber abzustimmen. Sie dienen zudem als unterstützendes Werkzeug während der Entwicklung. Im Rahmen dieser Arbeit werden im Frontend zwei neue Funktionsbereiche, die Stammdaten- und die Benutzerverwaltung, vollständig neu entwickelt. Für beide Bereiche werden entsprechende Mockups erstellt.

Diese Mockups dienen als Hilfestellung und geben eine klare gestalterische Richtung vor. Sie stellen jedoch kein starres Regelwerk dar. Dem Entwicklungsteam ist es wichtig, während der Implementierung sinnvolle gestalterische Anpassungen vorzunehmen, sei es aufgrund von Kundenwünschen, zur Optimierung der Benutzerführung oder zur besseren Integration externer Komponenten. Die Mockups bildet somit eine Orientierungshilfe, sind aber nicht als unveränderliches Konzept zu betrachten.

Mockup Stammdatenverwaltung

In der Stammdatenverwaltung (siehe Abbildung 3.9) können Administratoren Filter- und Projektmetadaten hinzufügen und bearbeiten. Das Ziel besteht darin, eine einheitliche Verwaltungsoberfläche für sämtliche

Stammdaten wie beispielsweise Arbeitsergebnisse, Departemente oder Regionen bereitzustellen. In der Abbildung ist stellvertretend ein Mockup zur Verwaltung der Regionen dargestellt. Für alle konfigurierbaren Metadaten soll eine entsprechende Seite mit passenden Verwaltungsfunktionen implementiert werden.

Die Datensätze werden tabellarisch dargestellt und können dort durchsucht sowie nach ihrem Status gefiltert werden. Über einen Bearbeitungsdialog lassen sich einzelne Einträge anpassen.

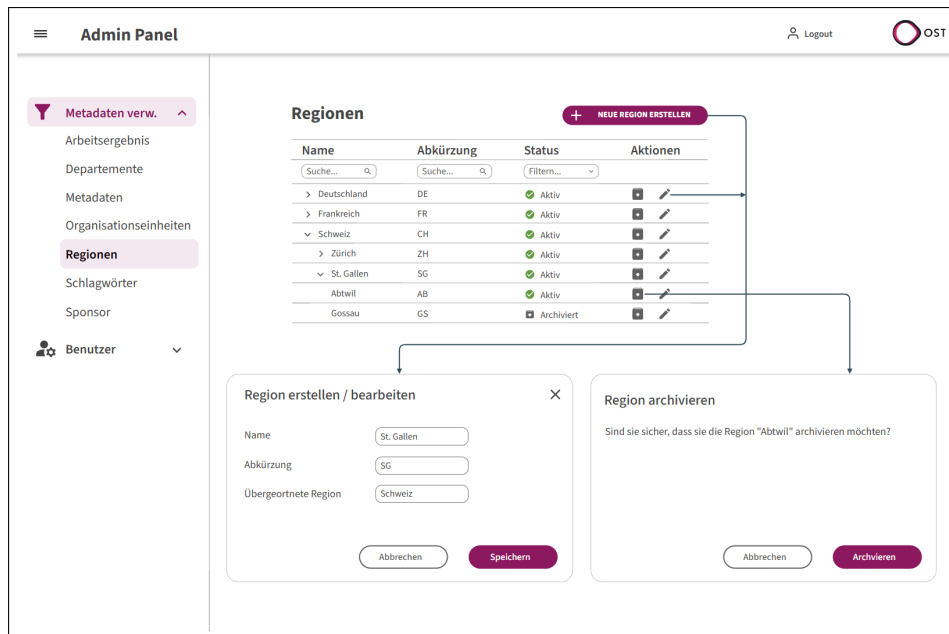


Abbildung 3.9: Mockup der Stammdatenverwaltung

Im Gegensatz zum Mockup, ist es in der finalen Anwendung zudem möglich, Datensätze direkt aus der Bearbeitungsansicht heraus zu archivieren. Darüber hinaus wurde die Option „Löschen“ ergänzt, die jedoch dann funktioniert, wenn der betreffende Datensatz in keinem bestehenden Projekt referenziert wird.

Mockup Benutzerverwaltung

Um eine übersichtliche Benutzerverwaltung zu gewährleisten, sind die verschiedenen Funktionen für Administratoren jeweils auf separaten Seiten dargestellt. Dazu gehören:

- **Neue Konten aktivieren**, um neue Benutzerkonten zu prüfen und freizugeben
- **Benutzer verwalten**, um bestehende Benutzerprofile zu bearbeiten
- **Gesperrte Benutzerkonten**, um gesperrte Konten einzusehen und sie bei Bedarf wieder zu aktivieren

Der Aufbau der Seiten zur Benutzerverwaltung folgt einem einheitlichen Gestaltungskonzept. Abbildung 3.10 zeigt exemplarisch das Mockup der Seite zur Aktivierung neuer Benutzerkonten. Die Benutzerinformationen werden dabei konsistent in einem tabellarischen Format dargestellt. Bei der Ausführung einer Aktion, wie etwa „Neues Konto aktivieren“, „Konto bearbeiten“ oder „Konto reaktivieren“ öffnet sich jeweils ein Dialogfenster. In diesem kann der Administrator die entsprechenden Änderungen vornehmen und anschliessend bestätigen.

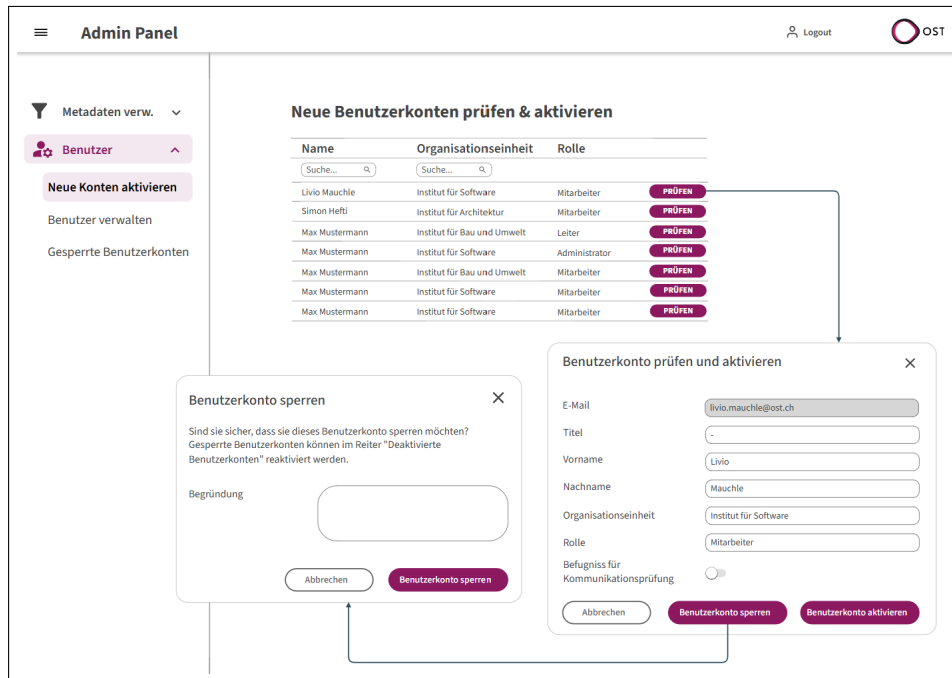


Abbildung 3.10: Mockup der Benutzerverwaltung

In der finalen Version der Anwendung haben Benutzer die Möglichkeit, ihr Profil selbstständig zu bearbeiten. Änderungen müssen jedoch weiterhin von einem Administrator freigegeben werden. Zu diesem Zweck wurde die Seite **Änderungsanfragen prüfen** ergänzt. Zudem werden in der Bearbeitungsansicht jeweils die aktuelle und die bearbeitete Version eines Profils gegenübergestellt. So kann der Administrator beispielsweise direkt erkennen, welche Angaben geändert wurden. Die ursprüngliche Version wird links, die vorgeschlagene oder überarbeitete rechts angezeigt. Zusätzlich informieren Badges darüber, ob neue Benutzerkonten oder Änderungsanfragen zur Prüfung vorliegen.

3.4.4 Verbesserung der Projektsuche

Um den Prototypen in ein vollständig intuitives und benutzerfreundliches Produkt zu überführen, werden verschiedene Aspekte des Designs und der Bedienbarkeit gezielt optimiert. Im Folgenden werden die Verbesserungen im Bereich der Projektsuche exemplarisch vorgestellt. Da die Projektsuche als besonders zentral gilt, liegt hier auch ein wichtiger Fokus der Verbesserungen. Andere Komponenten werden ebenfalls optimiert, werden hier aber der Übersichtlichkeit halber nicht im Detail behandelt.

In der Filterleiste werden nun auch im zugeklappten Zustand die aktiven Filter angezeigt. Jeder angewählte Filter erscheint dabei mit einem eigenen Tag. Sobald jedoch zu viele Filter gesetzt sind oder die Leiste aufgeklappt ist, wird stattdessen die Gesamtanzahl der Filter angezeigt. Über die Tags lassen sich die Filter leicht wieder abwählen. Im untenstehenden Vergleich (Abbildung 3.13) ist links die ursprüngliche Variante aus der Studienarbeit und rechts die neue Version zu sehen.

Abbildung 3.11 zeigt die ursprüngliche Filterleiste. Sie besteht aus einer Liste von Filterkategorien, die jeweils mit einem nach unten gerichteten Pfeil (▼) versehen sind:

- Arbeitsergebnis
- Schlagwörter
- Orte
- Sponsor
- Durchführungszeitraum
- Datum der Veröffentlichung
- Organisationseinheiten

Unter der Liste befinden sich zwei Buttons: "Filter zurücksetzen" und "Filter anwenden".

Abbildung 3.11: Filter vor den Verbesserungen

Abbildung 3.12 zeigt die verbesserte Filterleiste. Sie ist in mehrere Abschnitte unterteilt:

- Arbeitsergebnis:** Eine Liste von Filterkategorien mit Checkboxes. Die Kategorien "Patent" und "Software" sind ausgewählt und haben einen roten Haken. Rechts daneben steht eine rote Ziffer "2" mit einem nach oben gerichteten Pfeil (▲).
- Schlagwörter:** Zwei Tags "Umwelt und Energie" und "Klima und Energie" mit einem roten X zum Entfernen.
- Regionen:** Ein Tag "12" mit einem roten X zum Entfernen.
- Sponsor:** Zwei Tags "Eureka" und "Innosuisse" mit einem roten X zum Entfernen.
- Durchführungszeitraum:** Ein Tag "Letzte 3 Monate" mit einem roten X zum Entfernen.
- Datum der Veröffentlichung:** Zwei Tags "Ab: 1.1.2025" und "Bis: 1.12.2025" mit einem roten X zum Entfernen.
- Organisationseinheiten:** Ein Tag "Institut für Cybersecurity" mit einem roten X zum Entfernen.
- Departemente:** Ein Tag mit einem roten X zum Entfernen.

Unter der Liste befinden sich zwei Buttons: "Alle Einklappen" und "Filter zurücksetzen".

Abbildung 3.12: Filter nach den Verbesserungen

Abbildung 3.13: Vergleich der Filterleisten vor und nach den Usability-Verbesserungen

Bei den Datumsfiltern steht nun eine Schnellauswahl zur Verfügung. Mit dieser kann mit nur einem Klick nach Projekten in einem von vier vordefinierten Zeiträumen gesucht werden. Die Datumsfilter vor und nach der Verbesserung werden im untenstehenden Vergleich 3.16 abgebildet.

Abbildung 3.14: Datumsfilter vor Verbesserungen

Abbildung 3.15: Datumsfilter nach Verbesserungen

Abbildung 3.16: Vergleich der Datumsfilter vor und nach den Verbesserungen

Neu werden auch gefundene Suchbegriffe in der Projektsuche hervorgehoben. Das Hauptbild eines Projekts erscheint zudem direkt in der Trefferleiste. Dabei liefert das Backend die Bilder für eine gute Suchperformance als presigned URLs, die der Client dann nachlädt, sodass die Suche nicht verzögert wird. Die Ergebnisse aktualisieren sich nun automatisch bei jeder Filteränderung. Deshalb entfällt die Schaltfläche „Filter anwenden“ und es steht stattdessen die neue Schaltfläche „Alle einklappen“ zur Verfügung. Während des Ladevorgangs einer Suche wird dem Benutzer zudem durch einen Skeleton-Loader angezeigt, dass eine Suche durchgeführt wird. In den nachfolgenden Abbildungen ist die Projektsuche vor den Verbesserungen 3.17 sowie nach den Verbesserungen 3.18 zu sehen.

Abbildung 3.17: Screenshot der Projektsuche vor den Verbesserungen

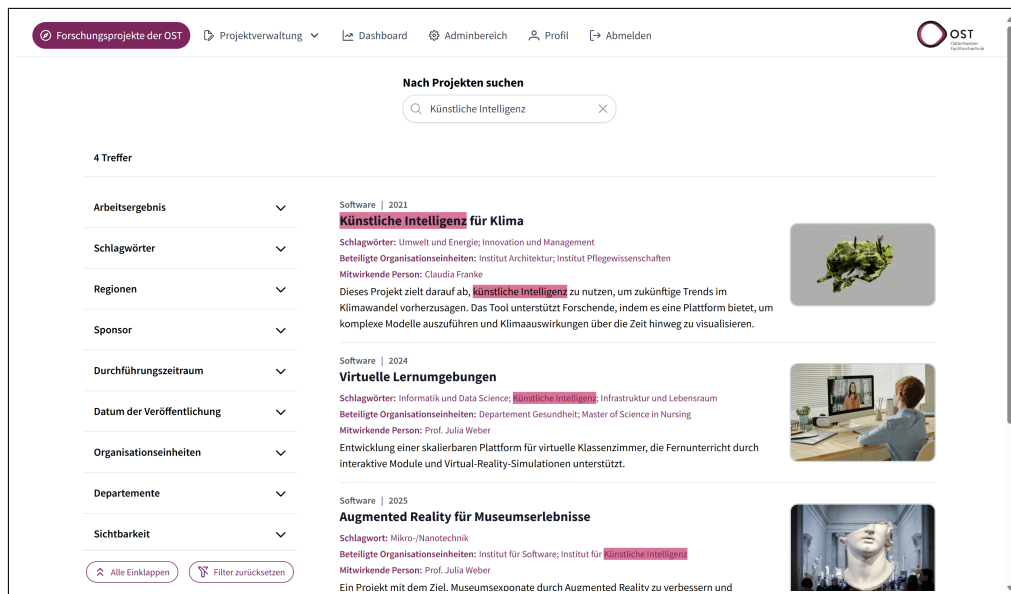


Abbildung 3.18: Seite der Projektsuche nach den Verbesserungen

3.4.5 Designentscheide

In Tabelle 3.10 sind jene Entscheidungen festgehalten, die im Rahmen der UI-Konzeption getroffen wurden und in den vorausgegangenen Abschnitten nicht ausführlich behandelt sind.

Thema	Entscheidung mit Begründung
Visualisierung aktiver Filter	Um die in der Projektsuche gesetzten Filter darzustellen, wurden zwei Ansätze geprüft: 1. Einzelne Tags direkt oberhalb der jeweiligen Filter 2. Eine zentrale Tag-Leiste, die alle aktiven Filter beinhaltet und oberhalb der Trefferliste anzeigt Obwohl Variante 2 mehr Platz und Gestaltungsspielraum bietet, wirkte sie zu unübersichtlich. Deshalb wurde Variante 1 gewählt: Sie bietet den Vorteil, dass die einzelnen Tags auf einen Blick ersichtlich machen, welche Filter aktuell aktiv sind.
Favorisieren von Filtereinstellungen im Dashboard	Um die Favorisierungsfunktion möglichst intuitiv zu gestalten, orientiert sie sich an der Lesezeichen-Funktion moderner Webbrowser und wurde entsprechend benannt. Filter-Lesezeichen lassen sich wie im Browser hinzufügen und wieder entfernen.
Verschachtelte Elemente in der Stammdatenverwaltung	Für die Darstellung verschachtelter Elemente (z. B. Regionen) wurden zwei Ansätze verglichen: 1. Flache Tabelle mit Navigation: Länder werden in der Haupttabelle gelistet. Ein Klick auf ein Land öffnet eine separate Kindtabelle mit den zugehörigen Kantonen. 2. Verschachtelte Tabelle: Länder und Kantone befinden sich in derselben Tabelle. Durch Aufklappen eines Landes werden die Kantone direkt darunter angezeigt. Option 2 wurde aufgrund der übersichtlicheren Darstellung aller Elemente in einer einzigen Tabelle ausgewählt. Zwar steigt dadurch die Implementierungskomplexität leicht an, doch gleicht die verwendete Komponentenbibliothek diesen Mehraufwand durch eingebaute Unterstützung für verschachtelte Tabellen wieder aus.
Navigation auf Admin- und Profilseite	Um die Hauptnavigation übersichtlich zu halten, erfolgt der Wechsel zwischen Benutzer- und Profilseiten über eine separate Seitennavigation. Die Header-Leiste wird bewusst nicht weiter ausgebaut, um eine Überladung mit Navigationselementen zu vermeiden. Insbesondere auf der Profilseite schafft dieses Konzept Raum für allfällige Erweiterungen.

Tabelle 3.10: Designentscheide

4 Umsetzung

Dieses Kapitel zeigt die technische Umsetzung von ausgewählten Teilen des Systems. Dabei wird besonders auf die Implementierung sowie spezifische Technologien und Frameworks eingegangen.

4.1 Suchoptimierung

Die Projektsuche nutzt den PostgreSQL-Datentyp `tsvector`. Ein Projekt enthält jedoch nicht nur eigene Textfelder, sondern zusätzlich Inhalte aus Regionen, Sponsoren, Organisationseinheiten, Metadaten, Schlagwörtern und Mitwirkenden (siehe [Abbildung 3.6](#)). Würde man diese Texte erst zur Laufzeit über mehrere Joins durchsuchen, würden Suchabfragen nur schlecht skalieren. PostgreSQL kann keinen GIN-Index anlegen, der mehrere Tabellen gleichzeitig abdeckt. Jede zusätzliche Join-Tabelle zwingt den Planer zu sequentiellen Scans und erhöht die Latenz somit deutlich.

Um diesen Nachteil zu umgehen, wird in der Tabelle der Projekte ein einziges, indiziertes Feld `search_vector` abgelegt, das alle relevanten Begriffe vorab aggregiert. Dieses Muster vereinfacht zudem auch die Implementierung der Suchabfrage sowie das Ranking, da nur noch ein Datenbankfeld abgefragt werden muss. Das Feld wird dabei über einen Datenbank Trigger aktuell gehalten.

4.1.1 Datenbank Trigger

Ein Datenbanktrigger sorgt dafür, dass bei jeder Aktualisierung eines Projekts automatisch eine Hilfsfunktion aufgerufen wird, die den Volltext-Suchvektor (`tsvector`) basierend auf den aktuellen Projektdaten neu berechnet. Darüber hinaus existieren zusätzliche Trigger auf allen relevanten abhängigen Tabellen, beispielsweise für Sponsoren, Regionen oder Organisationseinheiten. Dadurch wird sichergestellt, dass auch Änderungen an diesen Metadaten unmittelbar die zugehörigen `search_vector` der betroffenen Projekte aktualisieren. Ein typisches Beispiel hierfür wäre eine Umbenennung einer Organisationseinheit durch einen Administrator in der Stammdatenverwaltung.

Der folgende Codeausschnitt [4.1](#) zeigt den Anfang der Funktion `calc_project_search_vector`, die zur Berechnung des Suchvektors verwendet wird. Diese Funktion wird sowohl beim Aktualisieren eines Projekts aufgerufen, wobei die aktualisierten Felder als Parameter übergeben werden (z.B. `p_title`, als auch bei Änderungen an abhängigen Metadaten. In letzterem Fall erfolgt der Aufruf für jedes betroffene Projekt anhand seiner ID, wobei dann die aktuell gespeicherten Projektwerte (z.B. `pr.title`) verwendet werden.

Zur Berechnung des Suchvektors wird die PostgreSQL-Funktion `to_tsvector`. Sie wird auf sämtliche relevanten Metadaten des Projekts angewendet. Die dabei erzeugten Teilvektoren werden anschliessend gewichtet (z.B. Titel und Mitwirkende mit Gewicht A, Suchtags als Gewicht B usw.) und aggregiert um den Suchvektor zu erstellen.

```

1 -CREATE OR REPLACE FUNCTION calc_project_search_vector(
2     p_project_id bigint,
3     p_title text DEFAULT NULL,
4     p_abstract text DEFAULT NULL,
5     p_description text DEFAULT NULL,
6     p_project_type bigint DEFAULT NULL,
7     p_sponsor_id bigint DEFAULT NULL,
8     p_region_id bigint DEFAULT NULL
9 )
10 RETURNS tsvector
11 LANGUAGE plpgsql STABLE AS
12 $$
13 DECLARE
14 v_vec tsvector;
15 BEGIN
16     SELECT
17     /* A-weight: title -----*/
18     setweight(to_tsvector('german', coalesce(p_title, pr.title, '')), 'A') ||
19
20     /* A-weight: Contributor Names -----*/
21     setweight(
22         to_tsvector('german', coalesce((
23             SELECT string_agg(
24                 CASE
25                     WHEN c.user_id IS NULL
26                     THEN coalesce(c.contributor_description, '')
27                     ELSE
28                         concat_ws(' ', u.title, u.firstname, u.lastname)
29                     END, ' ')
30             FROM "Contributors" c
31             LEFT JOIN "AspNetUsers" u ON u.user_id = c.user_id
32             WHERE c.project_id = p_project_id
33         ), '')), 'A') ||
34     ...
35     INTO v_vec
36 FROM "Projects" pr
37 WHERE pr.project_id = p_project_id;
38
39 RETURN v_vec;
40 END;
41 $$;

```

Listing 4.1: Ausschnitt Trigger

4.1.2 Suche in PDF-Dokumenten

Um die Volltextsuche auch auf hochgeladene PDF-Dokumente auszuweiten, wird der Inhalt von PDFs beim Hochladen automatisch extrahiert und in einem eigenen tsvector-Feld gespeichert. Dieses Feld wird ausschliesslich für den Textinhalten von PDF-Dateien verwendet.

Die Trennung vom allgemeinen search_vector wird gemacht, weil tsvector keine Herkunftsinformationen zu einzelnen Begriffen speichert. Es ist daher im Nachhinein nicht mehr nachvollziehbar, ob ein Element im Vektor beispielsweise aus dem Projekttitel oder einem PDF-Dokument stammt. Änderungen wie etwa das Löschen oder Aktualisieren einzelner Begriffe würden potenziell auch andere Inhalte unbeabsichtigt beeinflussen. Eine konsistente Aktualisierung des Vektors ist deshalb nur durch eine vollständige

Neuerzeugung möglich. Diese könnte jedoch bei Projekten mit vielen oder grossen PDF-Dateien ineffizient werden, wenn sie bei jeder Änderung am Projekt oder den verwendeten Metadaten ausgeführt wird.

4.2 Authentifizierung

Der Ablauf der Authentifizierung mit OpenID-Connect wurde bereits in 3.7 visualisiert. Hier folgen dazu noch einige Code-Ausschnitte mit Beschreibungen.

4.2.1 Setup

Der AddAuthentication-Code befindet sich in einer WebApplicationBuilder-Extension-Methode, welche in Program.cs aufgerufen wird. Mit AddCookie() wird die Cookie-Authentifizierung aktiviert. AddOpenIdConnect wird ein AuthenticationScheme hinzugefügt, welches hier nach dem Provider GitLab benannt ist. Der Name wird später für die Challenge im Controller verwendet. Die GitlabSettings kommen aus der Konfiguration. Das Scope definiert, welche Berechtigungen der Benutzer auf GitLab der Applikation erteilen soll. Diese müssen in den Scopes enthalten sein, welche in der Application auf GitLab ausgewählt wurden (siehe 9.2.2). Die Json-Keys id und email, welche in den von GitLab erhaltenen Claims enthalten sind, werden auf .NET-ClaimTypes gemappt. Schliesslich wird ein EventListener definiert, der ausgelöst wird, sobald der erhaltene Token erfolgreich validiert wurde.

```

1 builder.Services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)
2     .AddCookie().AddOpenIdConnect("GitLab", options => {
3         options.Authority = gitlabSettings.Authority;
4         options.ClientId = gitlabSettings.ClientId;
5         options.ClientSecret = gitlabSettings.ClientSecret;
6         options.CallbackPath = gitlabSettings.CallbackPath;
7         options.ResponseType = "code";
8         options.SaveTokens = true;
9         options.Scope.Add("openid");
10        options.Scope.Add("profile");
11        options.Scope.Add("email");
12
13        options.ClaimActions.MapJsonKey(ClaimTypes.NameIdentifier, "id");
14        options.ClaimActions.MapJsonKey(ClaimTypes.Email, "email");
15
16        options.Events.OnTokenValidated = async context => {
17            var gitlabService = context.HttpContext.RequestServices.GetRequiredService<
18                GitlabService>();
19            await gitlabService.SignInWithGitlabAsync(context);
20        };
21    });

```

Listing 4.2: Registrierung der OIDC Authentication Services

4.2.2 GitlabService

Die Methode SignInWithGitlabAsync wird im OnTokenValidatedEvent aufgerufen. Zuerst wird geprüft, ob ein AccessToken und eine E-Mail in der Antwort enthalten sind. Dann werden die Claims Name und User-ID ausgelesen. Falls der User noch nicht existiert oder noch nicht mit GitLab verknüpft ist, wird über den UserManager ein neuer Benutzer angelegt bzw. seine GitLab-ID gespeichert und ein UserLogin für GitLab hinzugefügt. Am Ende wird der Benutzer über den SignInManager eingeloggt und weitergeleitet. Falls der

User bereits eine Rolle besitzt, wird er auf die Startseite weitergeleitet, sonst auf sein Profil, um dieses zu vervollständigen.

```

1 public async Task SignInWithGitlabAsync(TokenValidatedContext context)
2 {
3     var accessToken = context.TokenEndpointResponse?.AccessToken;
4     if (accessToken == null)
5     {
6         context.Fail("No access token provided by GitLab.");
7         return;
8     }
9
10    var email = context.Principal?.FindFirst(ClaimTypes.Email)?.Value;
11    if (email == null)
12    {
13        context.Fail("No email provided by GitLab.");
14        return;
15    }
16
17    var name = context.Principal?.FindFirst("name")?.Value;
18
19    var gitlabId = GetGitlabId(context, email);
20    var redirectUri = context.Properties?.Items["profileUri"] ?? "/";
21
22    var user = await userManager.FindByEmailAsync(email);
23
24    if (user == null)
25    {
26        var newUser = await CreateUserFromGitLab(email, name, gitlabId);
27        await SignInAsync(context, newUser, redirectUri);
28        return;
29    }
30
31    if (string.IsNullOrEmpty(user.GitlabId))
32        await AddGitLabLoginAsync(user, gitlabId);
33
34    var roles = await userManager.GetRolesAsync(user);
35    if (roles.Any()) redirectUri = context.Properties?.RedirectUri ?? "/";
36    await SignInAsync(context, user, redirectUri);
37 }
38
39 private async Task SignInAsync(TokenValidatedContext context, User user, string redirectUri)
40 {
41     await signInManager.SignInAsync(user, isPersistent: false);
42     context.Response.Redirect(redirectUri);
43     context.HandleResponse();
44 }
45
46 private async Task<User> CreateUserFromGitLab(string email, string? name, string gitLabId)
47 {
48     var user = new User { ... };
49     await userManager.CreateAsync(user);
50     await AddGitLabLoginAsync(user, gitLabId);
51     return user;
52 }

```

Listing 4.3: GitlabService, welcher im OnTokenValidated-Event aufgerufen wird.

4.2.3 Provider-Wechsel

Um den OIDC-Provider zu wechseln, müsste ein neues OpenIdConnect-AuthenticationScheme mit passender Konfiguration hinzugefügt werden. Im Code müssten einige Namen (z.B. GitlabService und SignInWithGitlabAsync) angepasst und der Provider statt hard-codiert in den AuthenticationProperties übergeben werden. Die GitlabId wird in der User-Klasse nicht benötigt, wenn stattdessen seine UserLogins mit dem UserManager nach dem übergebenen Provider durchsucht werden.

4.3 KI-Zusammenfassungen

Für die KI-Zusammenfassungen kann flexibel zwischen verschiedenen Anbietern gewechselt werden. Aktuell sind LLMHub und OpenAI implementiert. Im Folgenden sind einige Code-Ausschnitte beschrieben, welche das Setup und die Verwendung der beiden APIs implementieren.

4.4 Setup

Auch der AddLlm Code wird mittels Extension-Methode in Program.cs aufgerufen. In der Konfiguration ist ein LLM-Provider enthalten, welcher übergeben wird. Die Konfiguration wird nach eine Section mit dem Namen des Providers. Dann wird je nach Provider eine andere Action mit der Konfiguration durchgeführt. Für OpenAI wird ein IChatClient über die Methoden der Libraries Extensions.AI und OpenAI registriert. Für LLMHub existiert keine Library, weshalb stattdessen ein HttpClient mit der URL und einem Header für den API-Key registriert wird.

```

1 public static void AddLlm(this WebApplicationBuilder builder, IConfiguration configuration,
2     string llmProvider)
3 {
4     var llmSettings = configuration.GetSection(llmProvider).Get<LlmSettings>()
5         ?? throw new InvalidConfigurationException($"Configuration for LLM Provider {llmProvider} is missing");
6
7     builder.Services.AddSingleton(llmSettings);
8
9     var registrations = new Dictionary<string, Action<IServiceCollection, LlmSettings>>(
10         StringComparer.OrdinalIgnoreCase) {
11         ["OpenAI"] = RegisterOpenAi,
12         ["LLMHub"] = RegisterLlmHub
13     };
14
15     if (!registrations.TryGetValue(llmProvider, out var register))
16         throw new InvalidConfigurationException($"No registration for LLM provider {llmProvider}");
17
18     register(builder.Services, llmSettings);
19
20 private static void RegisterOpenAi(IServiceCollection services, LlmSettings settings)
21 {
22     services.AddScoped<IChatClient>(_ => new OpenAIClient(settings.ApiKey)
23         .GetChatClient(settings.Model)
24         .AsIChatClient());
25     services.AddScoped<ILlmClient, OpenAiLlmClient>();
26 }

```

```

26
27 private static void RegisterLlmHub(IServiceCollection services, LlmSettings settings)
28 {
29     services.AddHttpClient<ILlmClient, LlmHubClient>(client => {
30         client.BaseAddress = new Uri(settings.BaseUrl);
31         client.DefaultRequestHeaders.Add("llmhub-api-key", settings.ApiKey);
32     });
33 }

```

Listing 4.4: Registrierung des konfigurierten LLM-Clients

Da sich für die beiden Provider sowohl die verwendeten Clients als auch die Struktur der API-Responses stark unterscheiden, sehen auch die Implementationen sehr verschieden aus, wie die folgenden beiden Abschnitte zeigen.

4.4.1 LLMHubClient

Über eine statische Methode wird ein Prompt mit Metadaten und dem Input-Text erstellt. Mit `GetSummarizationPrompt` (siehe unten) wird der `HttpClient` aufgerufen. Die Antwort enthält einen Stream von Tokens, wobei pro Token eine neue Zeile übermittelt wird. Am Anfang der Antwort des LLM kann in einem `<think>`-Tag ein Thinking-Prozess enthalten sein. Dieser soll nicht zurückgegeben werden, weshalb die Weiterleitung der Antwort erst danach beginnt. Dann wird mit `yield return` jeder Token einzeln zurückgegeben.

```

1 public async IAsyncEnumerable<string> SummarizeAsync(string text, string language,
2   SummaryType summaryType, int maximumLength = Project.AbstractCharacterLimit,
3   [EnumeratorCancellation] CancellationToken cancellationToken = default)
4 {
5     var prompt = ILlmClient.GetSummarizationPrompt(text, language, summaryType, maximumLength,
6       InputTokenLimit);
7     using var response = await GetResponseFromPrompt(prompt, cancellationToken);
8     using var reader = new StreamReader(await response.Content.ReadAsStreamAsync(
9       cancellationToken));
10
11     var isThinking = false;
12     while (await reader.ReadLineAsync(cancellationToken) is { } line) {
13         if (string.IsNullOrEmpty(line)) continue;
14         var obj = JsonSerializer.Deserialize<LlmHubStreamChunk>(line, _serializerOptions);
15
16         var content = obj?.Content;
17         if (content == "<think>") {
18             isThinking = true;
19             continue;
20         }
21         if (isThinking) {
22             if (content == "</think>")
23                 isThinking = false;
24             continue;
25         }
26         yield return content ?? "";
27     }
28 }

```

Listing 4.5: SummarizeAsync-Implementation im LLMHub-Client

In der `GetResponseFromPrompt`-Methode wird die Request erstellt und mit `SendChatRequest` an die API gesendet. In `SendChatRequest` wird die `SendAsync`-Methode auf dem `HttpClient` aufgerufen. Falls die Request erfolgreich war, wird die Response zurückgegeben. Ansonsten wird geprüft, ob das in der Request enthaltene Model, welches aus der Konfiguration stammt, nicht existiert. In diesem Fall wird in `GetFirstAvailableModel` versucht, ein existierendes Model abzufragen (siehe unten). Falls ein Model gefunden werden konnte, wird es für die Request verwendet und in die Konfiguration eingetragen.

```

1 private async Task<HttpResponseMessage> GetResponseFromPrompt(string prompt,
2   CancellationToken cancellationToken = default)
3 {
4   var messages = new List<LlmHubChatMessage> { new("user", prompt) };
5   var request = new LlmHubChatRequest(messages, llmSettings.Model, true);
6   var response = await SendChatRequest(request, cancellationToken);
7
8   if (response.IsSuccessStatusCode)
9     return response;
10
11   if (await ModelNotFound(response, cancellationToken))
12   {
13     var newModel = await GetFirstAvailableModel(cancellationToken);
14     if (!string.IsNullOrEmpty(newModel))
15     {
16       llmSettings.Model = newModel;
17       request.Model = newModel;
18       response = await SendChatRequest(request, cancellationToken);
19     }
20   }
21   response.EnsureSuccessStatusCode();
22   return response;
23 }

```

Listing 4.6: `GetResponseFromPrompt`-Methode im LLMHub-Client

In der `GetFirstAvailableModel`-Methode wird der `/models`-Endpoint der API abgefragt und der Name des ersten gefundenen Models zurückgegeben.

```

1 private async Task<string?> GetFirstAvailableModel(CancellationToken cancellationToken = ↵
2   default)
3 {
4   using var httpRequest = new HttpRequestMessage(HttpMethod.Get, "/models");
5   using var response = await httpClient.SendAsync(httpRequest, cancellationToken);
6   response.EnsureSuccessStatusCode();
7
8   var body = await response.Content.ReadAsStringAsync(cancellationToken);
9   var models = JsonSerializer.Deserialize<LlmHubModelResponseItem[]>(body, ↵
10     _serializerOptions);
11
12   LlmHubModelResponseItem? firstModel = models?.FirstOrDefault();
13   return firstModel?.Model;
14 }

```

Listing 4.7: `GetFirstAvailableModel`-Methode im LLMHub-Client

4.4.2 OpenAIClient

Die Implementation im OpenAI-Client ist wesentlich einfacher, da der IChatClient den grössten Teil der Arbeit übernimmt. Während Tests fiel jedoch auf, dass OpenAI wesentlich besser funktioniert, wenn in der übergebenen ChatHistory vor dem eigentlichen Prompt bereits eine Nachricht von der Rolle „System“ enthalten ist, welche die gewünschte Struktur beschreibt, obwohl im Summarization-Prompt aus der statischen Methode dieselben Anweisungen enthalten sind. Auch hier wird ein Stream ausgelesen und jeder Token einzeln zurückgegeben.

```

1 public async IEnumerable<string> SummarizeAsync(string text, string language,
  SummaryType summaryType, int maxLength = Project.AbstractCharacterLimit, [
  EnumeratorCancellation] CancellationToken cancellationToken = default)
2 {
3     var prompt = ILLMClient.GetSummarizationPrompt(text, language, summaryType, maxLength,
  InputTokenLimit);
4     List<ChatMessage> chatHistory =
5     [
6         new (ChatRole.System, <Prompt containing structure, length, language>),
7         new (ChatRole.User, prompt)
8     ];
9     await foreach (var responseUpdate in aiClient.GetStreamingResponseAsync(chatHistory, null,
  cancellationToken))
10    {
11        var content = responseUpdate.Text;
12        yield return content;
13    }
14 }

```

Listing 4.8: SummarizeAsync-Implementation im OpenAI-Client

4.4.3 FileService

In der Implementation von GetSummary im FileService wird erst der Text aus dem gewünschten PDF-Dokument extrahiert. Damit wird die oben beschriebene Methode SummarizeAsync des ILLMClient aufgerufen. Bei jedem Token wird geprüft, ob die maximale Länge der Zusammenfassung nicht überschritten wird, ansonsten wird der Vorgang beendet. Ausserdem wird das scharfe S mit einem Doppel-S ersetzt.

```

1 public async IEnumerable<string> GetSummary(int projectId, int fileId,
2 SummaryType summaryType, int maxLength = Project.AbstractCharacterLimit,
3 [EnumeratorCancellation] CancellationToken cancellationToken = default)
4 {
5     var project = await GetProjectAsync(projectId, cancellationToken);
6     var file = GetFileFromProject(fileId, project) ??
7         throw new NotFoundException($"File with id {fileId} not found.");
8
9     if (!file.Reference.EndsWith(".pdf"))
10        throw new BadRequestException($"File with id {fileId} is not a PDF document.");
11
12    using var stream = await fileRepository.GetFileStream(file.Reference, cancellationToken);
13    var fileContent = pdfTextExtractor.ExtractText(stream);
14
15    var summary = "";
16    await foreach (var token in llmClient.SummarizeAsync(fileContent, project.Language ?? "of
  the text", summaryType, maxLength, cancellationToken))

```

```
17 {
18     if (summary.Length < 1 && string.IsNullOrEmpty(token)) continue;
19     var partialSummary = token.Replace("B", "ss");
20     summary += partialSummary;
21     if (summary.Length > maximumLength)
22         break;
23     yield return partialSummary;
24 }
25 }
```

Listing 4.9: Aufruf von SummarizeAsync in der FileService-Methode GetSummary

4.5 Texteditor

Für die Bearbeitung des Abstracts wird die Editor-Komponente von PrimeVue eingesetzt, die intern den Quill-Text-Editor verwendet. Die Einbettung über das auch sonst verwendete UI-Framework PrimeVue sorgt für ein einheitliches Styling innerhalb des Frontends.

4.5.1 Content Sanitizing (Backend)

Vor dem Speichern in der Datenbank wird das im Editor erzeugte HTML serverseitig bereinigt. Zum Einsatz kommt das Open-Source-Projekt HtmlSanitizer, mit dem gezielt **Cross-Site-Scripting (XSS)**-Payloads entfernt werden [32]. Die Konfiguration erfolgt anhand einer Positivliste. Lediglich die in Listing 4.10 definierten Tags, Attribute und URI-Schemata bleiben erhalten. Zudem werden nur CSS-Klassen mit dem Präfix „ql-“ akzeptiert, sodass intern verwendete Quill-Styles nicht verloren gehen, während fremde Klassen entfernt werden.

```
1 public class HtmlContentSanitizer : IHtmlContentSanitizer
2 {
3     public string SanitizeAbstract(string html)
4     {
5         var sanitizer = new HtmlSanitizer();
6
7         sanitizer.AllowedTags.UnionWith(["h2", "h3", "p", "strong", "em", "u", "ol", "li", "a", "span"]);
8         sanitizer.AllowedAttributes.UnionWith(["href", "target", "rel", "class"]);
9         sanitizer.UriAttributes.Add("href");
10        sanitizer.AllowedSchemes.UnionWith(["http", "https", "mailto", "tel"]);
11        sanitizer.RemovingCssClass += (_, e) =>
12        {
13            if (e.CssClass.StartsWith("ql-", StringComparison.Ordinal))
14                e.Cancel = true;
15        };
16
17        return sanitizer.Sanitize(html);
18    }
19 }
```

Listing 4.10: Konfiguration des HtmlSanitizer

4.5.2 Content Sanitizing (Frontend)

Gemäss dem **Defence in Depth**-Prinzip wird das bereits gereinigte HTML im Client zusätzlich gefiltert, bevor es gerendert wird. Dies erfolgt mittels DOMPurify. Die Sanitizer-Instanz wird zur Laufzeit als Nuxt-Plugin registriert. Dabei wird auf dem Server eine Jsdom-Umgebung initialisiert, damit die Funktionalität auch während der serverseitigen Generierung verfügbar ist. Die Konfiguration erfolgt analog zur oben gezeigten Positivliste im Backend. Durch diese zweistufige Sicherheit bleibt der HTML-Output selbst dann unkritisch, wenn ein potenzieller Angreifer Schwachstellen während des Transports oder bei der Speicherung ausnutzen könnte.

4.6 E-Mail-Versand

Für den E-Mail-Versand wird die Library FluentEmail verwendet. Die E-Mails werden aus RazorTemplates generiert, in welche Werte aus einem Model eingefügt werden. Wie in 3.2.9 erläutert, wird eine BackgroundTaskQueue für das Versenden der E-Mails verwendet.

Unten aufgeführt ist als Beispiel die Methode SendUserChangeRequestNotificationAsync, welche eine E-Mail an das Administrator-Postfach sendet, wenn ein Benutzer einen Account anlegt oder anpasst. Das Model für das Razor-Template wird initialisiert und gemeinsam mit dem Template-Pfad, dem Empfänger und dem Betreff an die Methode EnqueueEmail (siehe unten) übergeben.

```

1 public async Task SendUserChangeRequestNotificationAsync(UserChangeRequest changeRequest,
2     CancellationToken cancellationToken = default)
3 {
4     var templatePath = FileHelper.GetTemplateFile("Email", "UserChangeRequestNotification. ↵
5         cshhtml");
6     var orgUnit = await organisationalUnitRepository
7         .GetByIdAsync(changeRequest.RequestedOrgUnitId ?? 0, cancellationToken);
8     var orgUnitName = orgUnit?.Name ?? "";
9
10    var model = new MailUserChangeRequestModel
11    {
12        FullName = GetFullName(changeRequest.RequestedTitle ?? "",
13            changeRequest.RequestedFirstName, changeRequest.RequestedLastName),
14        UserEmail = changeRequest.User.Email ?? "",
15        IsCommValidator = changeRequest.RequestedIsCommValidator,
16        OrganisationalUnitName = orgUnitName,
17        Role = changeRequest.RequestedRole.GetDescription(),
18        Link = GetLink("admin/user-management/manage-changeRequests")
19    };
20    var to = settings.Value.AdminInbox;
21    EnqueueEmail(to, "OST ResearchNavigator: Aenderungsanfrage für Benutzerkonto", ↵
22        templatePath, model);
23 }
```

Listing 4.11: SendUserChangeRequestNotification-Implementierung im EmailSender

In der EnqueueEmail-Methode wird der Task-Queue ein neues Work-Item hinzugefügt. Für die Verwendung von Dependencies wie IFluentEmail ein Scope mit einer ServiceScoperFactory erstellt werden. Mit FluentEmail wird dann aus dem übergebenen Template, Betreff, Empfänger und Model eine E-Mail erstellt. Für das Logo im Footer muss ein Anhang mit dem Flag IsInline = true hinzugefügt werden. Dann wird das E-Mail abgesendet, mit der aktuellen Konfiguration an den Maildev-Server.

```
1 private void EnqueueEmail<T>(string to, string subject, string templatePath, T model)
2 {
3     taskQueue.QueueBackgroundWorkItem(async _ => {
4         using var scope = serviceScopeFactory.CreateScope();
5         var fluentEmail = scope.ServiceProvider.GetRequiredService<IFluentEmail>();
6
7         var email = fluentEmail.To(to)
8             .Subject(subject)
9             .UsingTemplateFromFile(templatePath, model);
10
11         var logoBytes = await FileHelper.GetImageBytes("logo-de.png");
12         using var memoryStream = new MemoryStream(logoBytes);
13
14         email.Attach(new Attachment {
15             ContentId = "logo",
16             IsInline = true,
17             ContentType = "image/png",
18             Data = memoryStream
19         });
20
21         var response = await email.SendAsync();
22
23         if (response.Successful)
24             logger.LogInformation("Sent email with subject {Subject} to {Email}", subject, to);
25         else
26             logger.LogWarning("Failed to send email to {Email}", to);
27     });
28 }
```

Listing 4.12: EnqueueEmail-Methode im EmailSender

Die Implementierung der BackgroundTaskQueue in Kombination mit einem BackgroundHostedService folgt der offiziellen Microsoft-Dokumentation [27].

5 Qualitätsmassnahmen

In diesem Abschnitt werden die Massnahmen und Strategien zur Qualitätssicherung beschrieben. Einige dieser Ansätze kamen bereits im Vorgängerprojekt zum Einsatz und werden in diesem Projekt weiter verwendet oder ausgebaut. Sie werden der Vollständigkeit halber erneut aufgeführt und entsprechend gekennzeichnet.

5.1 Code Qualität

Um eine hohe Code-Qualität sicherzustellen, sind Massnahmen und Richtlinien erforderlich, die die Entwickler während des Entwicklungsprozesses unterstützen und an die sich alle Beteiligten halten. In nachfolgenden Tabellen werden die im Rahmen der Studienarbeit definierten Massnahmen sowie neu hinzugekommene Ergänzungen und Anpassungen vollständig aufgeführt.

Massnahme	Beschreibung
Backend	Im Backend werden die von Microsoft vorgegebenen Coding Guidelines für C# verwendet. Diese werden in einem .editorconfig File festgehalten. Anhand einer eingerichteten run-config kann der Code so formatiert werden, dass er den von Microsoft empfohlenen Regeln zur Code-Formatierung entspricht.
Frontend	Im Frontend werden die von Nuxt ESLint vorgegebenen Standardrichtlinien verwendet. Auch diese Coding-Richtlinien können durch ESLint automatisiert geprüft und Probleme in vielen Fällen direkt behoben werden. Der Code wird zudem mittels einer ESLint Konfiguration automatisch formatiert.
NSwag	Mithilfe von NSwag werden auf Basis der OpenAPI-Spezifikation des Backends automatisch Typescript Typen, Interfaces und Enums für das Frontend generiert. Dadurch wird die Konsistenz zwischen Frontend und Backend sichergestellt, mögliche Fehlerquellen reduziert und die Entwicklung erleichtert.
Test Coverage	Für die Unit Tests im Backend sowie auch im Frontend kann je ein Test-Coverage-Report generiert werden. Dieser beinhaltet Informationen zur Branch coverage und Line coverage sowie auch andere interessante Statistiken und Kennzahlen. Anhand dieser Berichte kann eingesehen werden, ob die Testabdeckung ausreichend ist, und herausgefunden werden, an welchen Stellen im Code potentiell noch Tests fehlen.

Tabelle 5.1: Massnahmen zur Verbesserung der Code Qualität Teil 1/2

Massnahme	Beschreibung
Dokumentation	Für Entwickler relevante Informationen werden in einem README festgehalten. Schnittstellen werden mit Extensible Markup Language (XML) -Kommentaren dokumentiert, für die API -Dokumentation wird Swagger eingesetzt (siehe auch 5.5). Kommentare im Code sollen gezielt und nur bei Bedarf eingesetzt werden. Grundsätzlich sollte erst geprüft werden, ob Unklarheiten im Code durch die Aufteilung in kleinere Methoden und eine präzisere Benennung behoben werden können.
Automation	Diese Massnahmen werden zudem in der Pipeline automatisiert ausgeführt. Dies wird im Abschnitt 5.2.3 genauer beschrieben. Sind die oben definierten Massnahmen zur Einhaltung der Code Qualität nicht erfüllt, so kann dieser Code auch nicht in den main-branch eingeführt werden.

Tabelle 5.2: Massnahmen zur Verbesserung der Code Qualität Teil 2/2

5.2 Versionskontrolle

Das im Rahmen der Studienarbeit eingerichtete Repository auf der GitLab-Instanz der OST wird weiterhin verwendet. Es ermöglicht eine strukturierte Verwaltung des Codes und erleichtert die Zusammenarbeit im Team.

5.2.1 Monorepository Ansatz

Im Rahmen der Studienarbeit wurde das Projekt als Monorepository aufgebaut, bei dem Frontend und Backend gemeinsam in einem Repository gehalten werden. Diese Struktur erlaubt es, zusammenhängende Änderungen an beiden Teilen der Anwendung gebündelt in einem Branch oder Commit umzusetzen. Aufgrund der begrenzten Projekt- und Teamgrösse treten typische Herausforderungen eines Monorepos, wie etwa Performance-Einbussen oder häufige Merge-Konflikte, in diesem Fall nicht auf [33].

5.2.2 Workflow

Für die Entwicklung wird der Gitflow-Workflow verwendet, ein bewährtes Modell für die Versionsverwaltung mit Git [34]. Dieser Workflow erleichtert die parallele Entwicklung mehrerer Funktionalitäten und unterstützt einen strukturierten Prozess mit regelmässigen Code-Reviews.

Folgende Branches werden verwendet:

- **feature-Branches:** Neue Funktionen und Änderungen werden je in einem eigenen Feature-Branch entwickelt. Diese Branches werden von dev abgezweigt und nach Fertigstellung auch wieder in den dev-Branch integriert.
- **dev-Branch:** Ist der Hauptentwicklungszweig. Hier werden alle fertiggestellten Features zusammengeführt.
- **main-Branch:** Enthält ausschliesslich stabile Versionen der Anwendung. Nur nach einer erfolgreichen Prüfung durch den technischen Betreuer wird der stabile und getestete Code von dev in main übernommen.

Feature-Branches werden nach folgendem Schema benannt:

feature/<Jira Ticketnummer>-<Branchname>

5.2.3 Pipeline

Zur automatischen Prüfung der definierten Richtlinien und Massnahmen zur Sicherstellung der Codequalität wurde im Rahmen der vorangegangenen Studienarbeit eine GitLab-Pipeline eingerichtet. Diese wird weiterhin gepflegt und aktuell gehalten. Die wichtigsten Schritte der Pipeline sind:

- **build:** Überprüfung, ob der Code erfolgreich kompiliert werden kann.
- **lint:** Validierung der Einhaltung definierter Formatierungsregeln.
- **test:** Ausführung von Unit Tests, ergänzt um Tests des Frontends im Rahmen dieser Arbeit.

Sollte einer der in der Pipeline durchgeführten Schritte fehlschlagen, wird die Ausführung abgebrochen. Dadurch ist sichergestellt, dass nur Änderungen, die sämtliche Anforderungen der Pipeline erfüllen, in den `main`-Branch übernommen werden können.

5.3 Code Reviews

Dank der Durchführung von Code Reviews durch einen Fachexperten der OST erfolgt eine zusätzliche Überprüfung von Quellcode und Architektur der Anwendung. Mithilfe dieser Reviews sollen potenzielle Schwachstellen frühzeitig identifiziert und gezielt behoben werden. Darüber hinaus tragen sie zur Verbesserung der Codequalität bei, indem bewährte Praktiken gefördert und architekturbezogene Entscheidungen kritisch hinterfragt werden. Die im Rahmen der Reviews identifizierten Mängel sowie die daraus abgeleiteten Massnahmen und Verbesserungen sind im Anhang unter 9.1.2 dokumentiert.

5.4 Testing

Dieser Abschnitt gibt einen Überblick über die durchgeführten Testmethoden.

5.4.1 Unit Tests

Unit-Tests dienen dazu, die korrekte Funktionalität einzelner Komponenten isoliert voneinander zu überprüfen. Sie werden fortlaufend im Entwicklungsprozess erstellt, um frühzeitig Fehler zu erkennen.

Unit Tests Backend

Wie in NFR-4 definiert (siehe 2.18), wird für die Unit Tests im Backend eine **Branch Coverage** von über 80% angestrebt. Der Fokus liegt dabei auf den Service-Klassen, da in diesen die zentrale Geschäftslogik implementiert ist. Die Wahl der **Branch Coverage** als Metrik soll sicherstellen, dass möglichst viele Entscheidungszweige des Codes getestet werden. Die tatsächlich erreichte Abdeckung der Service-Klassen ist in Abbildung 5.1 zu sehen.

OST.ResearchNavigator.Services.Services.AccessControlService	77	2	79	121	97.4%		72	74	97.2%	
OST.ResearchNavigator.Services.Services.AccountService	59	0	59	91	100%		11	12	91.6%	
OST.ResearchNavigator.Services.Services.BookmarkService	18	0	18	34	100%		2	2	100%	
OST.ResearchNavigator.Services.Services.CoreDataService	63	40	103	144	61.1%		10	10	100%	
OST.ResearchNavigator.Services.Services.FileService	149	3	152	229	98%		44	50	88%	
OST.ResearchNavigator.Services.Services.MetricsService	12	0	12	23	100%		0	0		
OST.ResearchNavigator.Services.Services.OrganisationalUnitService	4	0	4	14	100%		0	0		
OST.ResearchNavigator.Services.Services.ProjectService	402	21	423	578	95%		72	84	85.7%	
OST.ResearchNavigator.Services.Services.SearchService	70	13	83	116	84.3%		15	18	83.3%	
OST.ResearchNavigator.Services.Services.StatisticsService	156	3	159	239	98.1%		28	30	93.3%	
OST.ResearchNavigator.Services.Services.UserManagementService	148	0	148	218	100%		25	26	96.1%	
OST.ResearchNavigator.Services.Services.ValidationService	190	6	196	283	96.9%		67	80	83.7%	

Abbildung 5.1: Testabdeckung der Service-Klassen im Backend

Unit Tests Frontend

Auch wenn Unit Tests im Frontend nicht im Fokus dieser Arbeit stehen und daher nicht in den **NFR** aufgeführt sind, werden grundlegende Tests aus Gründen der Wartbarkeit dennoch umgesetzt. Getestet werden insbesondere Utility-Methoden, State-Container sowie, wo sinnvoll auch Composables. Es wird dabei eine Abdeckung von 80% Branch-Coverage oder mehr angestrebt. Die erzielte Testabdeckung dieser Komponenten ist in Abbildung 5.2 dargestellt. In dieser Abbildung werden aus Gründen der Übersichtlichkeit nur jene Composables aufgeführt, die explizit getestet wurden.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	58.88	85.71	81.74	58.88	
publikationsapp-frontend/composables	15.41	65	48.27	15.41	
useTreeSelection.ts	95.53	85.36	100	95.53	18-19,46,100-101
publikationsapp-frontend/stores	94.55	84.09	85.71	94.55	
adminMetricsStore.ts	100	100	100	100	
bookmarkStore.ts	97.26	88.88	100	97.26	67-68
dashboardStore.ts	91.66	100	50	91.66	31-33
userStore.ts	92.03	70	83.33	92.03	85-86,88-89,107-111
publikationsapp-frontend/utils	99.03	94.44	100	99.03	
chartUtils.ts	100	100	100	100	
dateUtils.ts	100	100	100	100	
debounceUtils.ts	100	100	100	100	
errorUtils.ts	97.29	89.47	100	97.29	38
highlightUtils.ts	100	100	100	100	
mappingUtils.ts	100	90.9	100	100	29
streamUtils.ts	100	100	100	100	
treeUtils.ts	96.42	92.3	100	96.42	32

Abbildung 5.2: Testabdeckung der Unit Tests im Frontend

5.4.2 Usability Tests

Zur Bewertung der Benutzerfreundlichkeit und Bedienbarkeit der entwickelten Anwendung werden gezielte Usability-Tests mit realen Nutzerinnen und Nutzern durchgeführt. Das Ziel dieser Tests besteht darin, mögliche Schwierigkeiten in der Benutzerführung zu identifizieren und zu beheben. Die Tests finden nicht nur am Ende des Projekts, sondern auch während der Entwicklung statt, um frühzeitige Verbesserungen zu ermöglichen. Gegen Ende des Projekts findet eine letzte Testrunde statt, um finale Rückmeldungen aufzunehmen und das Produkt abschliessend zu optimieren.

Getestet wird die Applikation mit insgesamt fünf potenziellen Anwendern, die unterschiedliche Perspektiven auf das System mitbringen. Die Testgruppe setzt sich aus Mitarbeitenden eines Instituts, Studierenden der OST sowie einer Person aus dem administrativen Bereich zusammen. Durch diese Durchmischung werden verschiedenartige Nutzungsszenarien und Perspektiven abgedeckt und es kann vielfältiges Feedback zur Verbesserung des Systems gesammelt werden.

Die Testfälle sowie die detaillierte Vorgehensweise sind im beigefügten Testprotokoll dokumentiert (siehe 9.1.1). Die einzelnen Testdurchführungen sind im Anhang unter 9.1.5 zu finden. Eine Übersicht über die im Rahmen der Tests identifizierten Schwachstellen und die daraus abgeleiteten Verbesserungen ist im Anhang in Tabelle 9.9 beigelegt.

5.4.3 System Tests

Um sicherzustellen, dass alle relevanten Funktionen der Anwendung einwandfrei funktionieren, wurde das im Vorgängerprojekt erstellte Systemtestprotokoll um Testfälle zu den neu implementierten Funktionalitäten erweitert und vollständig überprüft. Das Protokoll der finalen Testdurchführung ist im Anhang 9.1.3 zu finden.

5.4.4 Load Tests

Für das Testen der Performance wurde wie in der Studienarbeit das Tool Apache JMeter verwendet. Der Testplan wurde mit Tests für die Stammdatenverwaltung ergänzt. Dabei werden die Neuerfassung, Bearbeitung sowie das Löschen von Arbeitsergebnissen sowie das Laden aller und die Bearbeitung eines verwendeten Search-Tags 50 mal innerhalb von 2s ausgeführt. Für die ID beim Löschen und die Version beim Bearbeiten wurde eine CSV Data Set Config hinzugefügt. In den CSV-Dateien sind 50 aufeinanderfolgende Zahlen für die Version bzw. für die zuvor erstellten Arbeitsergebnisse enthalten.

Für den Autorisierungs-Cookie wurde ein neuer Header hinzugefügt und der Wert eines gültigen Cookies aus dem Browser kopiert.

Auch die bestehenden Tests wurden ausgeführt. Die Anforderungen der Studienarbeit werden weiterhin erfüllt.

Screenshots von Apache JMeter mit den angepassten Einstellungen und allen Ergebnissen befinden sich im Anhang (9.1.4).

5.5 Dokumentation des Codes

Ein nachvollziehbar dokumentierter Code erleichtert es jedem neuen Teammitglied, sich rasch in das Projekt einzuarbeiten. Gleichzeitig darf die Pflege der Dokumentation nicht zur Belastung werden. Jede Änderung am Code sollte sich ohne grossen Zusatzaufwand in der Dokumentation widerspiegeln. Die automatisierten Werkzeuge, die dafür seit dem Vorgängerprojekt existieren, werden weiterhin verwendet, gepflegt und ausgebaut. Im Folgenden werden die verwendeten Hilfsmittel kurz beschrieben.

Wie gehabt wird beim lokalen Start des Backends automatisch eine Swagger-UI-Instanz erstellt und im Browser geöffnet [35]. Sie listet sämtliche REST-Endpunkte mitsamt Parameter-, Request- und Response-Beschreibungen auf und erlaubt direkte Testaufrufe. Damit sind manuelle Tests der API ohne Zusatztools möglich (siehe Anhang 9.2.5).

Zudem wird das Tool Doxygen weiterhin für das Generieren einer durchsuchbaren Dokumentation des Programms verwendet [36]. Dazu werden alle Schnittstellen mit XML-Kommentaren versehen. Auf Basis dieser Kommentare generiert Doxygen eine Hypertext Markup Language (HTML)-Dokumentation, die Entwicklern als Hilfsmittel dient. Die generierte Dokumentation kann bequem über eine vordefinierte Run-Configuration geöffnet werden. Ein Auszug aus der Dokumentation befindet sich im Anhang (9.2.6).

6 Projektplan

6.1 Vorgehen im Projekt

Das Projekt orientiert sich hinsichtlich der übergeordneten Planung und Phasenstruktur am **Rational Unified Process (RUP)**. Für die kurzfristige, operative Planung kommt ergänzend das Scrum-Framework zum Einsatz, um eine höhere Flexibilität sowie iterative Weiterentwicklungen zu ermöglichen. Diese Kombination vereint die Vorteile strukturierter Planung mit den Prinzipien agiler Entwicklung und hat sich bereits im Vorgängerprojekt bewährt.

Während der gesamten Projektlaufzeit finden wöchentliche Meetings mit den Betreuungspersonen statt. Zusätzlich organisiert das Projektteam mindestens einmal pro Woche interne Besprechungen, die bei Bedarf auch kurzfristig und situationsabhängig durchgeführt werden.

6.2 Zeitplan

Der Zeitplan visualisiert sowohl die Projektphasen gemäss **Rational Unified Process (RUP)** als auch die inhaltlichen Arbeitspakete. Die Phasen nach RUP sind durch dunkelblaue Balken gekennzeichnet, während die gelben Balken die vom Team definierten Epics repräsentieren. Grün markierte Balken stehen für Anforderungen, die als „Could-Have“ eingestuft wurden. Diese optionalen Anforderungen werden nur dann umgesetzt, wenn der Projektfortschritt dies zulässt.

Zur besseren Nachverfolgbarkeit des tatsächlichen Projektverlaufs wurde unter jedem Planungselement ein entsprechend gefärbter Fortschrittsbalken ergänzt. Diese sogenannte Ist-Darstellung ermöglicht eine direkte visuelle Gegenüberstellung von geplanter und realisierter Umsetzung und erleichtert so das Erkennen von Abweichungen.

Der Zeitplan konnte im Wesentlichen eingehalten werden. Einige Features wurden schneller als erwartet umgesetzt, während bei anderen geringfügige Verzögerungen aufkamen. In Einzelfällen waren kurzfristige Anpassungen erforderlich. So wurde ein grösseres Refactoring im Rahmen der technischen Schulden aus Gründen der Parallelentwicklung um zwei Wochen verschoben. Die Anforderungen „PDF-Erstellung“ und „PDF-Textsuche“ waren ursprünglich nicht vorgesehen, konnten jedoch aufgrund des zügigen Projektfortschritts und auf Wunsch des Auftraggebers zusätzlich eingeplant und realisiert werden.

Publikationsapp für Forschungsprojekte

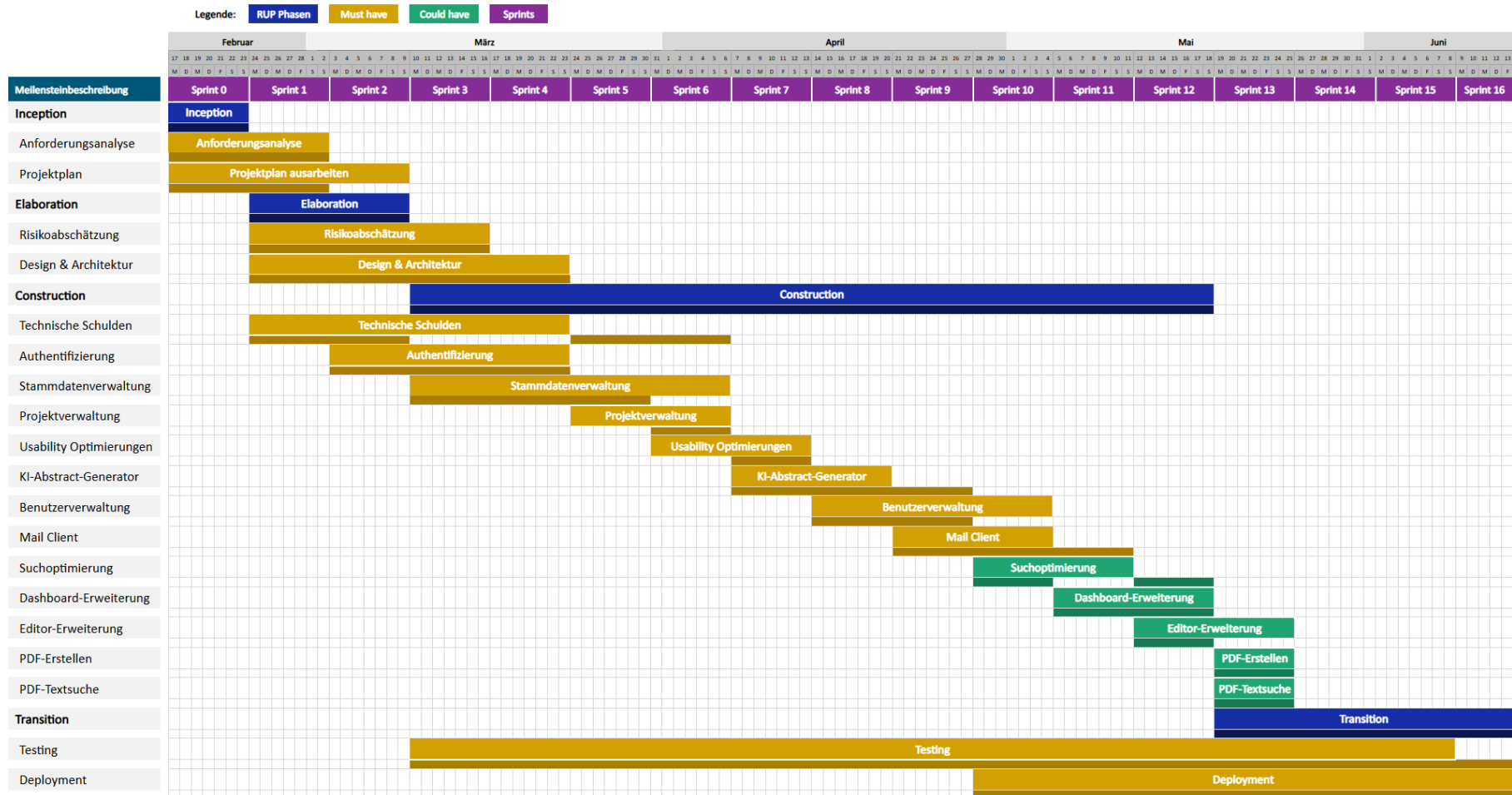


Abbildung 6.1: Projektplan

6.2.1 Meilensteine

Jedes im Gantt-Diagramm gelb hervorgehobene Epic endet in einem wichtigen Meilenstein. Die exakten Termine dieser Meilensteine sind in der nachstehenden Tabelle aufgeführt.

Datum	Meilenstein
23.02.2025	Ende der Inception Phase
02.03.2025	Anforderungsanalyse abgeschlossen
09.03.2025	Projektplan ist fertig ausgearbeitet
09.03.2025	Ende der Elaboration Phase
16.03.2025	Initiale Risikoabschätzung ist erfasst
23.03.2025	wichtigste Technische Schulden wurden aufgearbeitet
23.03.2025	Design & Architektur sind fertig ausgearbeitet
23.03.2025	Epic: Authentifizierung abgeschlossen
06.04.2025	Epic: Stammdatenverwaltung abgeschlossen
06.04.2025	Epic: Projektverwaltung abgeschlossen
13.04.2025	wichtigste Usability-Optimierungen (aus SA) wurden umgesetzt
20.04.2025	Epic: KI-Abstract-Generator abgeschlossen
04.05.2025	Epic: Benutzerverwaltung abgeschlossen
04.05.2025	Epic: Mail-Client abgeschlossen
18.05.2025	Ende der Construction Phase
08.06.2025	Testingphase abgeschlossen
13.06.2025	Projekt ist veröffentlicht und einsatzbereit
13.06.2025	Ende der Transition Phase (Abgabe)

Tabelle 6.1: Meilensteine

6.3 Risikomanagement

In diesem Abschnitt werden die Projektrisiken (siehe Tabelle 6.3.5) dokumentiert. Da sich Risiken im Verlauf des Projekts verändern können, sind hier auch mehrere Risikoanalysen zu unterschiedlichen Zeitpunkten festgehalten.

6.3.1 Initiale Risikoauswertung vom 24.02.2025 - Ende der Inception Phase

W/I ¹	Vernachlässigbar	Geringfügig	Kritisch	Katastrophal
Sicher	-	-	-	-
Wahrscheinlich	-	-	-	-
Möglich	-	Bugs	Usability / Umfang	-
Unwahrscheinlich	-	-	-	Security
Selten	-	-	KI-Integration	-
Eliminiert	-			

¹ Wahrscheinlichkeit/Impact

Tabelle 6.2: Risikomatrix 24.02.2025

6.3.2 Risikoauswertung vom 09.03.2025 - Ende der Elaboration Phase

W/I ¹	Vernachlässigbar	Geringfügig	Kritisch	Katastrophal
Sicher	-	-	-	-
Wahrscheinlich	-	-	Nuxt Performance	-
Möglich	-	Bugs	Usability	-
Unwahrscheinlich	-	-	Umfang	Security
Selten	-	-	KI-Integration	-
Eliminiert	-			

¹ Wahrscheinlichkeit/Impact

Tabelle 6.3: Risikomatrix 09.03.2025

Anmerkungen

Zur Minderung des Risikos „**Umfang**“ erstellte das Entwicklerteam basierend auf Aufwandschätzungen einen Zeitplan, der den erwarteten Aufwand pro Feature definiert. Zusätzlich wurde eine zeitliche Reserve eingeplant, die vorrangig für optionale Features, im Bedarfsfall aber auch für „Must-Have“ Features genutzt werden kann. Dadurch konnte das Risiko von „möglich“ auf „unwahrscheinlich“ herabgestuft werden. Im Zuge der Planung wurde jedoch ein neues Risiko „**Nuxt-Performance**“ identifiziert, welches in Abschnitt 6.3.5 erläutert wird.

6.3.3 Risikoauswertung vom 20.04.2025 - Mitte der Construction Phase

W/I ¹	Vernachlässigbar	Geringfügig	Kritisch	Katastrophal
Sicher	-	-	-	-
Wahrscheinlich	-	-	-	-
Möglich	-	-	-	-
Unwahrscheinlich	-	Bugs	Usability	-
Selten	-	-	Umfang	Security
Eliminiert	Nuxt Performance, KI-Integration			

¹ Wahrscheinlichkeit/Impact

Tabelle 6.4: Risikomatrix 20.04.2025

Anmerkungen

Das Risiko „**Bugs**“ wurde auf „unwahrscheinlich“ herabgestuft. Ausschlaggebend dafür sind umgesetzte technische Verbesserungen, welche im Vorgängerprojekt nicht umgesetzt werden konnten, sowie auch zusätzliche Optimierungen, die im Rahmen einer Architekturanalyse und Code-Reviews durch eine technische Betreuungsperson entdeckt und vom Entwicklerteam umgesetzt wurden. Das Risiko „**Umfang**“ wurde auf „selten“ herabgestuft, da sich das Projekt aktuell im Zeitplan befindet und ausreichend Zeit für alle „Must-Have“ Features eingeplant ist. Die verbleibende Projektdauer dürfte voraussichtlich auch für einige oder sogar alle „Could-Have“-Features ausreichen. Das Risiko „**Usability**“ wurde durch erste Usability-Tests sowie durch umgesetzte Verbesserungen, die aus den Tests des Vorgängerprojekts abgeleitet wurden, auf „unwahrscheinlich“ reduziert. Das Risiko „**Security**“ wurde auf „selten“ herabgestuft. Die Authentifizierung wurde erfolgreich implementiert, wobei bewährte Mechanismen wie eine Microsoft-Bibliothek und der GitLab Authentication Provider zum Einsatz kamen. Das Risiko „**Nuxt Performance**“ konnte vollständig eliminiert werden, da entsprechende Massnahmen erfolgreich

umgesetzt wurden. Das Risiko „**KI-Integration**“ gilt ebenfalls als eliminiert, da die Integration mit Hilfe von einem OST-internen **LLM** bereits erfolgt ist und nur noch kleinere Anpassungen ausstehen.

6.3.4 Risikoauswertung vom 01.06.2025 - Mitte der Transition Phase

W/I ¹	Vernachlässigbar	Geringfügig	Kritisch	Katastrophal
Sicher	-	-	-	-
Wahrscheinlich	-	-	-	-
Möglich	-	-	-	-
Unwahrscheinlich	-	-	-	-
Selten	-	Bugs	-	Security
Eliminiert	Nuxt Performance, KI-Integration, Usability, Umfang			

¹ Wahrscheinlichkeit/Impact

Tabelle 6.5: Risikomatrix 20.04.2025

Anmerkungen

Das Risiko „**Bugs**“ wurde auf „selten“ herabgestuft. Die zentralen Funktionalitäten der Anwendung wurden im Rahmen mehrerer Benutzertests intensiv geprüft. Zudem ist die Anwendung durch automatisierte Tests gut abgedeckt. Sowohl im Backend, gemäss der in den **NFR** definierten Testabdeckung, als auch im Frontend durch ergänzende Unit-Tests. Zusätzlich wurde ein umfassendes Protokoll für Systemtests erstellt, anhand dem die Anwendung gründlich getestet wurde. Eine vollständige Eliminierung des Risikos ist jedoch nicht möglich, da das Auftreten einzelner Fehler nie mit absoluter Sicherheit ausgeschlossen werden kann.

Das Risiko „**Usability**“ wurde als eliminiert eingestuft. Im Zuge von Usability-Tests konnten potenzielle Probleme identifiziert und gezielt behoben werden. Die durchgeführten Optimierungen führten zu positivem Feedback seitens der Testpersonen, was auf eine insgesamt benutzerfreundliche Gestaltung der Anwendung hinweist.

Das Risiko „**Umfang**“ konnte ebenfalls vollständig eliminiert werden. Sämtliche „Must-Have“- und „Could-Have“-Features wurden im Projektverlauf erfolgreich umgesetzt.

6.3.5 Identifizierte Risiken

Nr	1
Titel	Umfang
Beschreibung	Das zu entwickelnde System hat sich seit der Studienarbeit im geplanten Umfang ausgeweitet. Zusätzlich zu den geplanten Funktionalitäten sollen technische Optimierungen und eine verbesserte Benutzeroberfläche für die finale Auslieferung umgesetzt werden. Die Gefahr besteht insbesondere darin, sich in Details zu verlieren und dadurch den Fokus auf die Kernfunktionen zu vernachlässigen.
Vorbeugung	Eine klare Priorisierung der Anforderungen in Abstimmung mit dem Kunden, um die zentralen Features für das fertige Produkt festzulegen. Technische und UI-Verbesserungen sollten innerhalb eines festgelegten Zeitrahmens priorisiert abgearbeitet werden. So wird sichergestellt, dass genügend Zeit für die Entwicklung der Kernfunktionen verbleibt.

Tabelle 6.6: Risiko - Umfang

Nr	2
Titel	Bugs
Beschreibung	Durch Refactorings im Rahmen technischer Verbesserungen sowie auch durch neue Features wie beispielsweise die manuelle Bearbeitungsmöglichkeit von Stammdaten können neue Bugs entstehen.
Vorbeugung	Weiterhin eine hohe Testabdeckung im Backend sicherstellen. Grössere Refactorings frühzeitig einplanen und durchführen. Zudem ausreichend Zeit für manuelle Tests vor der Einführung einplanen, um potenzielle Fehler rechtzeitig zu identifizieren und zu beheben.

Tabelle 6.7: Risiko - Bugs

Nr	3
Titel	Usability
Beschreibung	Die entwickelten Funktionen könnten nicht intuitiv bedienbar sein.
Vorbeugung	Durchführung von Usability-Tests mit mehreren Testpersonen, um Schwachstellen in der Benutzerführung zu identifizieren. Zudem sollten während der Entwicklung sowie vor dem Release gezielt Zeit für Optimierungen der Benutzeroberfläche eingeplant werden.

Tabelle 6.8: Risiko - Usability

Nr	4
Titel	Security
Beschreibung	Sicherheitslücken wie beispielsweise eine unzureichende Eingabevalidierung oder Probleme in der Benutzerverwaltung können dazu führen, dass die Anwendung anfällig für Angriffe (z.B. SQL-Injections) ist.
Vorbeugung	Durchführung manueller Sicherheitschecks mit SwaggerUI. Anwenden von Best Practices der sicheren Programmierung. Verwenden eines externen Authentifizierungsanbieters (GitLab).

Tabelle 6.9: Risiko - Security

Nr	5
Titel	KI-Integration
Beschreibung	Durch die Verwendung von Künstlicher Intelligenz zur automatischen Generierung von Abstracts besteht das Risiko, dass vertrauliche oder schützenswerte Daten unbeabsichtigt nach aussen gelangen.
Vorbeugung	Verwendung von einem OST-internen LLM, um die Verarbeitung sensibler Daten innerhalb einer gesicherten Umgebung zu gewährleisten.

Tabelle 6.10: Risiko - KI-Integration

Nr	6
Titel	Nuxt Performance
Beschreibung	Derzeit führt ein Deployment in der produktiven Umgebung dazu, dass alle Seiten des Frontends beim ersten Aufruf nach dem Neustart der Anwendung merklich verzögert laden. Nach dem erstmaligen Laden verbessert sich die Performance und entspricht den Erwartungen.
Vorbeugung	Es soll geprüft werden, ob ein Preloading der einzelnen Seiten möglich ist. Dies ist insbesondere relevant, um eine durchgehend flüssige Nutzererfahrung auch nach einem Release oder einem kurzzeitigen Ausfall der Anwendung sicherzustellen.

Tabelle 6.11: Risiko - Nuxt Performance

7 Ergebnisse

7.1 Übersicht der Ergebnisse

In diesem Abschnitt werden die im Rahmen dieser Arbeit entwickelten Funktionalitäten anhand von Screenshots und kurzen Beschreibungen zusammengefasst.

Die Startseite (siehe Abbildung 7.1) ermöglicht dank implementierter Suchoptimierungen eine Volltextsuche über alle relevanten Projektmetadaten. Die Suche bleibt auch bei Umbenennungen von Metadaten durch Administratoren stets aktuell. Neu ist zudem, dass Projekte auch anhand von Informationen aus angehängten PDF-Dateien gefunden werden. Die Suchergebnisse werden gewichtet und die Projekte in der Reihenfolge ihrer Relevanz dargestellt. In der Filterleiste sind aktive Filter anhand von Tags ersichtlich. In der Trefferliste werden nun auch Titelbilder der Projekte angezeigt und gefundene Suchbegriffe farblich hervorgehoben.

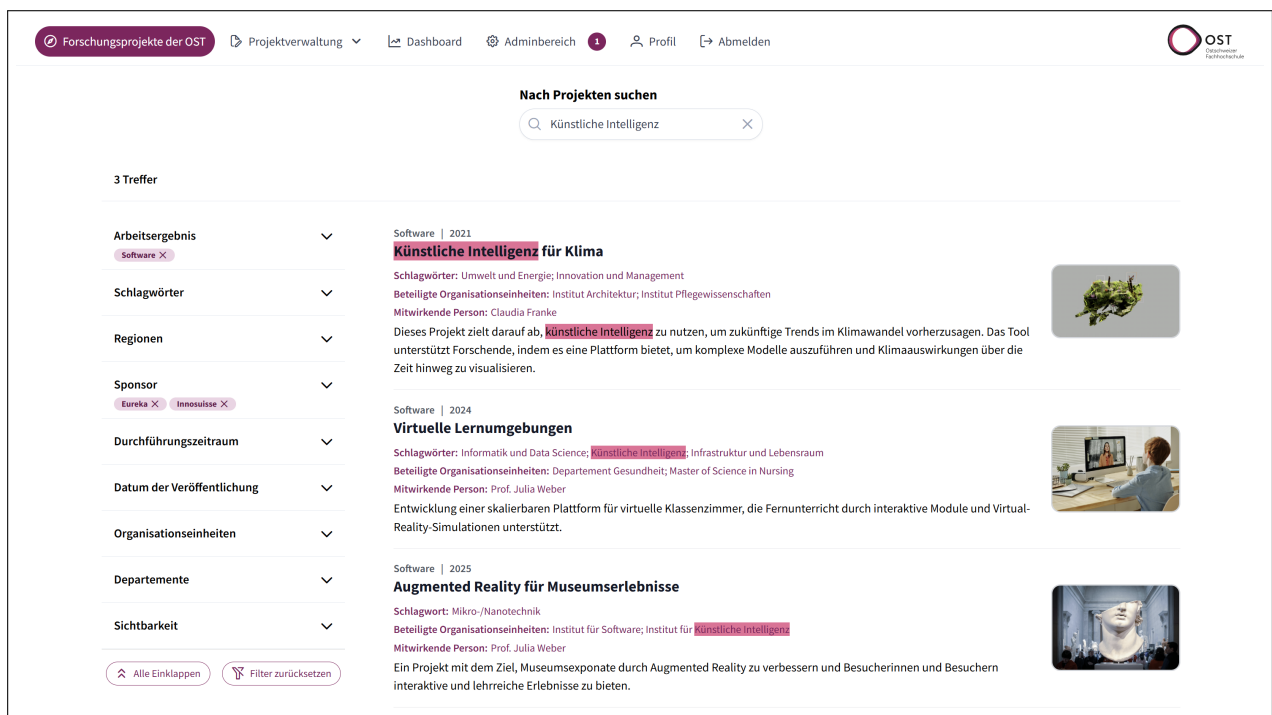


Abbildung 7.1: Screenshot Startseite

Im Statistik-Dashboard werden, wie in den Abbildungen 7.2 zu sehen ist, vier zusätzliche Statistiken angezeigt, deren Darstellung sich mithilfe der Filterleiste anpassen lässt. Die Auswertung zur Anzahl der Projektbeteiligungen pro Person kann über eine Konfiguration flexibel ein- oder ausgeschaltet werden.

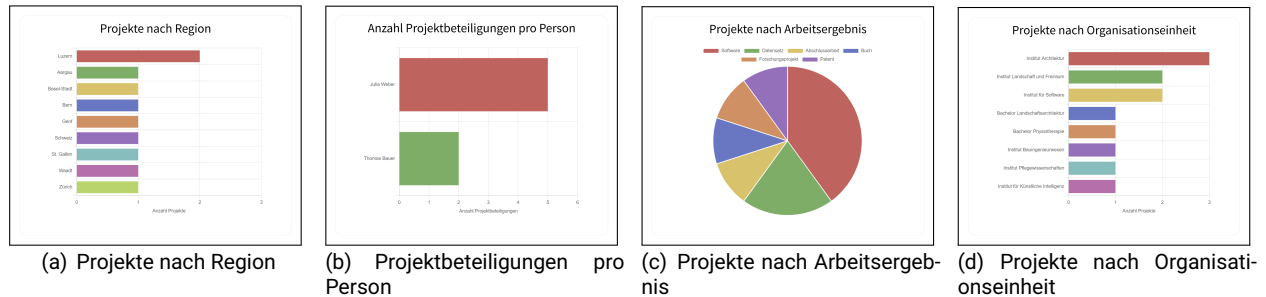


Abbildung 7.2: Zusammenstellung der vier neuen Statistiken

In Abbildung 7.3 ist zu sehen, wie durch das Setzen von Lesezeichen Filtereinstellungen gespeichert werden können. Ein Benutzer kann so mit einem Klick zwischen verschiedenen gespeicherten Filtern wechseln.

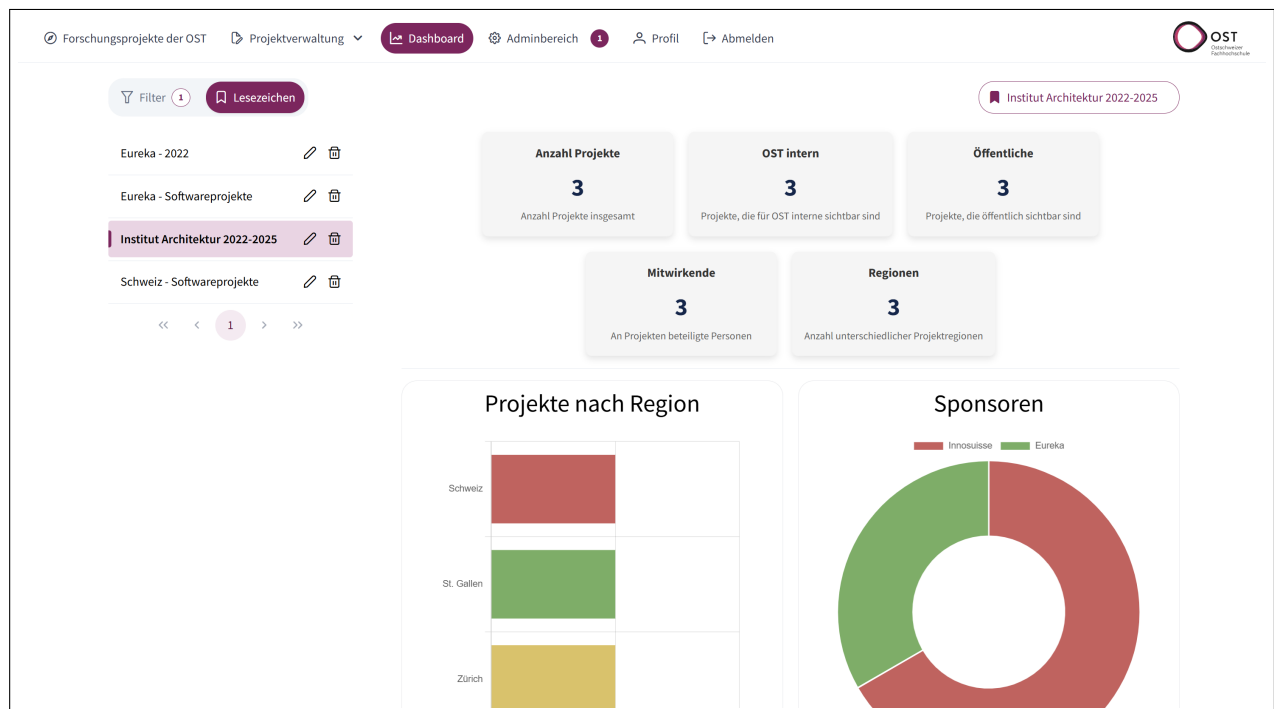


Abbildung 7.3: Screenshot Lesezeichenleiste

Die in Abbildung 7.4 gezeigte Tabelle veranschaulicht exemplarisch die Stammdatenverwaltung am Beispiel der Regionen. Administratoren können neue Einträge für Arbeitsergebnisse, Metadaten, Organisationseinheiten, Regionen, Schlagwörter und Sponsoren selbstständig anlegen und verwalten. Bereits in Projekten verwendete Einträge können zudem archiviert werden, sodass sie nicht mehr in den Filtereinstellungen oder auf der Projekterfassungsseite erscheinen.

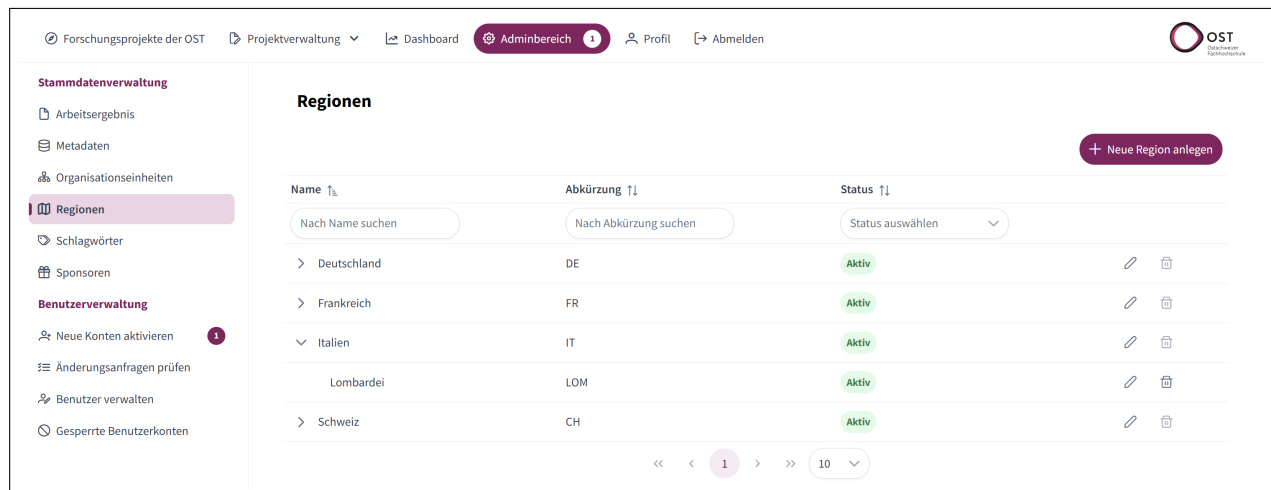


Abbildung 7.4: Screenshot Stammdatenverwaltung: Seite „Regionen“

Benutzer können sich nun mit ihrem GitLab-Konto an der Anwendung authentifizieren. Die Zugriffsmöglichkeiten innerhalb der Anwendung werden über fünf verschiedene Benutzerrollen gesteuert. Jede Rolle verfügt über spezifische Berechtigungen für bestimmte Funktionalitäten.

Die Benutzerverwaltung, wie in Abbildung 7.5 dargestellt, ermöglicht es Administratoren, neue Benutzerkonten zu aktivieren, bestehende Benutzer zu bearbeiten sowie Konten bei Bedarf zu sperren. Gesperrten Benutzerkonten werden sämtliche Berechtigungen entzogen. Sie können jedoch jederzeit durch einen Administrator reaktiviert werden, beispielsweise im Falle eines Austritts mit anschließendem Wiedereintritt eines Mitarbeiters.

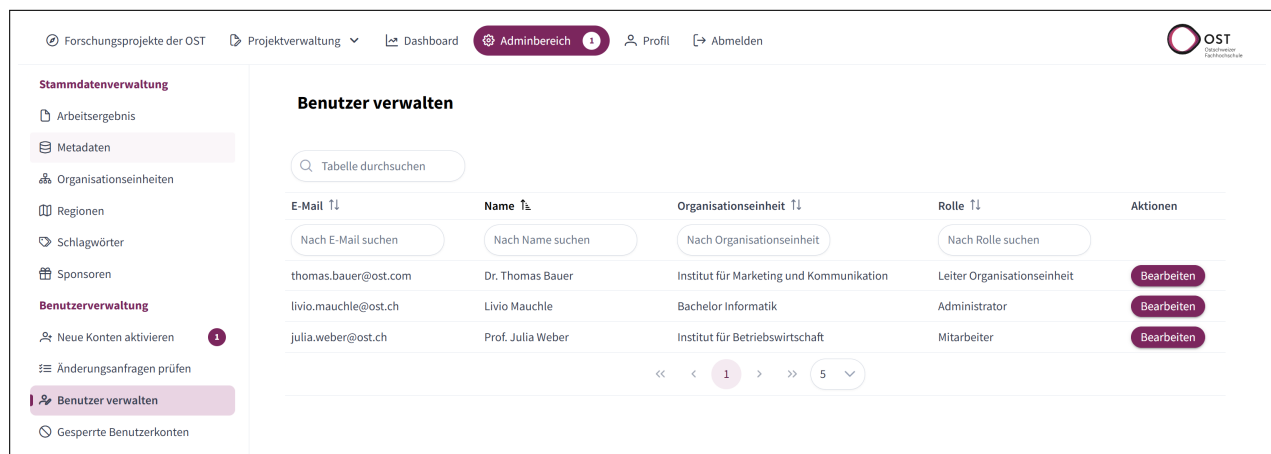


Abbildung 7.5: Screenshot Benutzerverwaltung: Seite „Benutzer verwalten“

Wie in Abbildung 7.6 zu sehen ist, haben Benutzer auch die Möglichkeit, selbstständig Änderungen an ihrem Profil vorzunehmen. Diese Änderungen müssen jedoch in jedem Fall von einem Administrator geprüft und bestätigt werden.

Forschungsprojekte der OST | Dashboard | Adminbereich | **Profil** | Abmelden

Benutzerprofil | **Profil bearbeiten**

Profil bearbeiten

Ihre Profiländerung wird von einem Administrator geprüft und nach dessen Freigabe übernommen.

Aktuelle Version	Änderungsanfrage
Anrede Herr	Anrede* Herr
Titel -	Titel Titel
Vorname Livio	Vorname* Livio
Nachname Mauchle	Nachname* Mauchle
Organisationseinheit Bachelor Informatik	Organisationseinheit* Bachelor Architektur
Rolle Administrator	Rolle* Administrator
Prüfer Kommunikation Nein	Prüfer Kommunikation ☐

Änderungsanfrage anpassen

Abbildung 7.6: Screenshot der Profilsite „Profil bearbeiten“

In der Projekterfassung kann die Projektbeschreibung mithilfe eines hochgeladenen PDF-Dokuments automatisch durch KI generiert werden. Das zugrunde liegende Modell kann über eine Konfigurationsdatei angepasst werden, ohne dass Änderungen am Code erforderlich sind. Zusätzlich kann die Beschreibung manuell über einen Texteditor bearbeitet und formatiert werden. Die verfügbaren Formatierungsoptionen sind so gestaltet, dass die Beschreibungen verschiedener Projekte stilistisch einheitlich wirken. Während der Bearbeitung steht den Anwendern eine Vorschau der Projektdetailseite zur Verfügung. Sowohl in dieser Vorschau als auch in der finalen Projektdetailansicht kann das Abstract als PDF-Dokument exportiert werden. Der Texteditor für die Formatierung von Projektbeschreibungen ist in Abbildung 7.7 zu sehen.

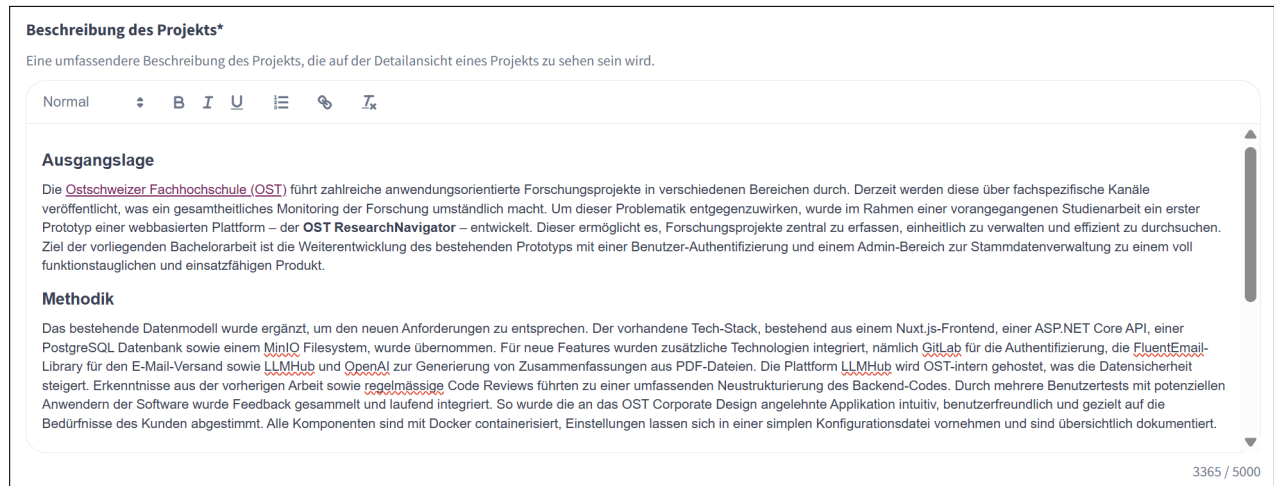


Abbildung 7.7: Screenshot des Texteditors

Nach der Veröffentlichung eines Projekts ist es möglich, dieses zu bearbeiten und erneut zu veröffentlichen. Zudem können veröffentlichte Projekte archiviert werden. Beide Funktionen stehen wie in [Abbildung 7.8](#) dargestellt über entsprechende Schaltflächen auf dem als deaktiviert dargestellten Eintrag des veröffentlichten Projekts zur Verfügung.

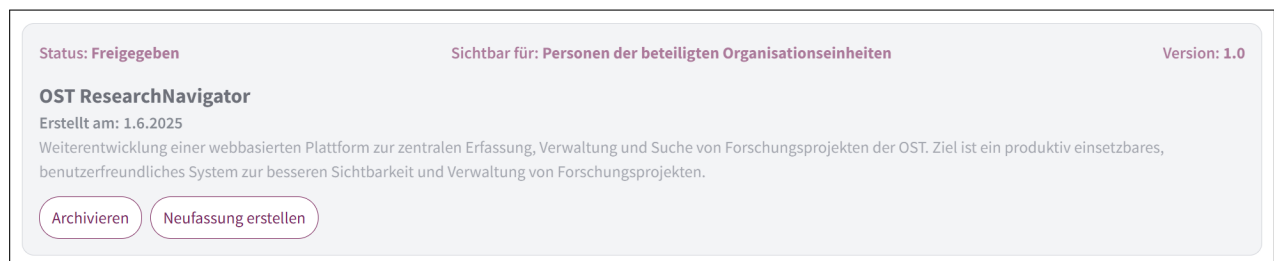


Abbildung 7.8: Screenshot archivieren und Neufassung erstellen

Die Abläufe der Projektfreigabe sowie die Benutzerverwaltung werden durch den Versand von Benachrichtigungs-E-Mails unterstützt. [Abbildung 7.9](#) zeigt beispielhaft eine E-Mail, die versendet wird, wenn ein Projekt im Rahmen des Freigabeprozesses zurückgewiesen wird.



Abbildung 7.9: Screenshot eines Benachrichtigungs-Mails

7.1.1 Statistiken zum Code

Umfang

Zur groben Abschätzung des Projektumfangs wurden die Anzahl der Dateien sowie die Anzahl an Codezeilen ermittelt. Die Vorgehensweise zur Bestimmung der Codezeilen entspricht derjenigen aus der Studienarbeit und wird in Abschnitt 9.2.1 näher erläutert. Dadurch ist ein direkter Vergleich zwischen dem Prototypen und dem finalen Produkt möglich. Tabelle 7.1 zeigt die Anzahl der Codezeilen ohne Leerzeilen und Kommentare sowie die Veränderung im Vergleich zur Studienarbeit.

Projekt	Sprache/Technologie	Files	Anzahl Codezeilen	Veränderung
Frontend	Vuejs Component	83	9290	(+50/+4669)
Frontend	TypeScript	56	2280	(+28/+1233)
Frontend	Total	139	11570	(+78/+5902)
Backend	C#	192	8481	(+94/+4448)
Unit Tests Frontend	TypeScript	13	1055	(+13/+1055)
Unit Tests Backend	C#	27	3628	(+14/+1537)
Projekt Total	Alle	371	24734	(+201/+12960)

Tabelle 7.1: Auflistung der Anzahl Codezeilen

Verglichen mit der Studienarbeit hat sich der Umfang etwa verdoppelt, was insgesamt nachvollziehbar ist. Zwar ist der Rahmen dieser Arbeit grösser, dennoch entstehen beim Neuaufbau eines Projekts typischerweise viele Codezeilen durch die Erstellung von Grundstrukturen und initialen Implementierungen. In diesem Projekt standen Erweiterungen und Verbesserungen im Vordergrund, die nicht zwangsläufig zu einem proportionalen Anstieg der Codezeilen führen.

Testabdeckung

Die erzielte Testabdeckung, die in Abschnitt 5.4.1 detailliert beschrieben wird, beläuft sich auf:

- 88.7% **Branch Coverage** bei den Service-Klassen im Backend
- 85.7% **Branch Coverage** im Frontend

7.1.2 Zielerreichung

Alle sechs Must-Have Use Cases sowie alle fünf optionalen Could-Have Use Cases konnten vom Entwicklerteam erfolgreich implementiert, getestet und auf Basis von Benutzerfeedback validiert sowie bei Bedarf optimiert werden. Zwei der optionalen Use Cases, der PDF-Export und die Suche innerhalb von PDF-Dokumenten waren zu Beginn des Projekts nicht vorgesehen. Aufgrund des positiven Projektfortschritts konnten sie jedoch flexibel in den Projektumfang aufgenommen und erfolgreich umgesetzt werden. Die umgesetzten Funktionalitäten umfassen:

- UC01 - Authentifizierung (Must Have)
- UC02 - Stammdatenverwaltung (Must Have)
- UC03 - veröffentlichtes Projekt verwalten (Must Have)
- UC04 - KI-Abstract-Generator (Must Have)
- UC05 - Benutzerverwaltung (Must Have)
- UC06 - Benachrichtigungs-Mail versenden (Must Have)
- UC07 - Suchoptimierung (Could Have)
- UC08 - Dashboard-Erweiterung (Could Have)
- UC09 - Editor-Erweiterung (Could Have)
- UC10 - PDF Generieren (Could Have)
- UC11 - PDF Textsuche (Could Have)

Darüber hinaus konnte auch die Bedienbarkeit der Anwendung in mehreren Bereichen gezielt verbessert werden. Beispiele hierfür sind die überarbeitete Projektsuche (siehe Abschnitt 3.4.4) sowie die Optimierungen, die sich aus den Usability-Tests ergeben haben (siehe Abschnitt 9.1.1).

Alle im Zuge dieser Arbeit definierten **NFR** wurden verifiziert und sind vollständig erfüllt. Sie umfassen Anforderungen an die folgenden Qualitätsmerkmale:

- Performance Efficiency (Response Time)
- Usability (Affordance)
- Security (Authentication)
- Maintainability (Test Coverage)
- Transferability (Konfigurationen)
- Reliability (Backups)

Darüber hinaus sind auch sämtliche **NFR**, die bereits im Zuge der vorangegangenen Studienarbeit definiert wurden, weiterhin erfüllt.

8 Ergebnisdiskussion

8.1 Fazit

Ziel dieser Arbeit war es, den im Rahmen der vorangegangenen Studienarbeit entwickelten Prototyp zu einem voll funktionsfähigen und einsatzbereiten Produkt weiterzuentwickeln.

Das fertige Produkt implementiert sämtliche als Must-have definierten Use Cases. Darüber hinaus wurden auch alle optionalen (Could-have) Anwendungsfälle erfolgreich umgesetzt, einschliesslich zweier ursprünglich nicht geplanter Erweiterungen (PDF-Export und PDF-Textsuche), die dank einer effizienten Umsetzung zusätzlich implementiert werden konnten.

Auch die nicht-funktionalen Anforderungen wurden in vollem Umfang erfüllt. Die im Rahmen der Studienarbeit entstandenen technischen Schulden konnten nahezu vollständig abgebaut werden. Zudem wurde die Codequalität in mehreren Code Reviews gemeinsam mit dem technischen Betreuer überprüft, wobei das erhaltene Feedback grösstenteils direkt umgesetzt wurde. In diesem Zusammenhang wurde unter anderem das Backend klar strukturiert und in eigenständige Projekte aufgeteilt, wodurch sich die Wartbarkeit und Erweiterbarkeit des Systems verbessert haben.

Die Anwendung erfüllt die Sicherheitsanforderungen für den produktiven Einsatz durch eine zuverlässige Authentifizierung mit GitLab. Darüber hinaus bietet die Administrationsoberfläche autorisiertem Personal der OST die Möglichkeit zur Verwaltung von Benutzerkonten. Zeitkritische Prozesse der Benutzerverwaltung und Projektveröffentlichung werden zudem durch automatisierte E-Mail-Benachrichtigungen unterstützt.

Die integrierte Stammdatenverwaltung erlaubt es, Änderungen an Metadaten auch im laufenden Betrieb problemlos vorzunehmen. Die Projektsuche wurde so gestaltet, dass nicht nur alle Projektmetadaten, sondern auch Inhalte aus beigefügten PDF-Dokumenten durchsucht werden können. Eine übersichtlich gestaltete Filterleiste mit Tags sowie eine Hervorhebung von gesuchten Begriffen verbessern zusätzlich das Benutzererlebnis.

Veröffentlichte Projekte können neu archiviert und bei Bedarf erneut bearbeitet und veröffentlicht werden. In veröffentlichten Projekten sowie auch während der Bearbeitung eines Projekts kann die Projektbeschreibung als PDF exportiert werden.

Auch das Statistik-Dashboard wurde erweitert. Die Anzahl der dargestellten Grafiken wurde von vier auf acht erhöht. Zusätzlich können eigens definierte Filter als Lesezeichen gespeichert und mit nur einem Klick zu einem späteren Zeitpunkt erneut abgerufen werden.

Die Eingabeoberfläche zur Projekterfassung wurde um mehrere Funktionen erweitert. So ist es nun möglich, Beschreibungen automatisch mittels künstlicher Intelligenz generieren zu lassen. Darüber hinaus

unterstützt ein integrierter Editor die formatierte Eingabe der Projektbeschreibung. Während der Bearbeitung kann zudem eine Vorschau des Projekts angezeigt werden.

Alle zentralen Funktionen wurden in Usability-Tests mit potenziellen Anwendern auf ihre Benutzerfreundlichkeit hin überprüft und bei Bedarf optimiert.

Auf Basis dieser Evaluation sowie des durchweg positiven Feedbacks aus den Usability-Tests lässt sich die Arbeit als Erfolg werten. Das Ziel der Arbeit wurde erreicht, und die Anwendung steht für die Übernahme durch ein Institut der OST bereit.

8.2 Ausblick

8.2.1 Erweiterungen

Um die Applikation von einer betriebsbereiten Lösung in ein aktiv genutztes System zu überführen, ist insbesondere eine enge Integration in die Systemlandschaft der OST von Vorteil. Erweiterungen, die einen unkomplizierten Datenaustausch mit bestehenden Systemen und Datenbanken ermöglichen, sind dabei besonders relevant.

Zum einen könnten Funktionen entwickelt werden, die die Übernahme bestehender Projekte in das Ökosystem des OST ResearchNavigators erleichtern. Zum anderen wären Erweiterungen denkbar, die es ermöglichen, aktuelle Projektdaten aus dem ResearchNavigator in andere Anwendungen zu übertragen.

Ein konkretes Beispiel, wie in Use Case 12 (siehe Tabelle 2.14) beschrieben, ist die Integration mit SAP. Dadurch könnten dort erfasste Projekte automatisch übernommen und im ResearchNavigator um fehlende Metadaten ergänzt werden. Eine solche Schnittstelle würde insbesondere die Vollständigkeit der im System enthaltenen Projekte verbessern.

8.2.2 Verbesserungsmöglichkeiten

Im Rahmen der Usability Tests und Code Reviews konnten bereits viele Verbesserungen umgesetzt werden. Dennoch bestehen weiterhin technische Verbesserungsmöglichkeiten, die sowohl unter den technischen Schulden (siehe Abschnitt 3.3) als auch in den bislang nicht umgesetzten Empfehlungen aus den Code Reviews (siehe Tabelle 9.15) dokumentiert sind.

8.3 Weiterentwicklung

Der OST ResearchNavigator wird zunächst innerhalb der OST evaluiert. Bei positivem Ergebnis ist zu erwarten, dass ein Institut der OST die Verantwortung für den operativen Betrieb übernimmt und die Weiterentwicklung sowie Wartung des Systems sicherstellt.

Danksagung

Fabio Daniel Marti danken wir für die kompetente Betreuung und die wertvolle Unterstützung, die wesentlich zum Erfolg des Projekts beigetragen haben.

Clemens Meier danken wir für das Teilen seiner Expertise, seine konstruktiven Anregungen und die gründlichen Code Reviews, durch die die Qualität der Anwendung wesentlich verbessert wurde.

Prof. Laurent Metzger und Prof. Alex Simeon danken wir für ihr wertvolles Feedback und die grossartigen Ideen, die das Projekt in seiner aktuellen Form ermöglicht haben.

Literaturverzeichnis

- [1] S. H. und Livio Mauchle, "Ost researchnavigator: Projekt-tracking-software für die ostschweizer fachhochschule," Online: <https://eprints.ost.ch/id/eprint/1259/>, 2024.
- [2] C. Larman, "Use cases," in *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3rd ed. Upper Saddle River, NJUnited States: Prentice Hall PTR, 2004.
- [3] B. Yan, K. Li, M. Xu, Y. Dong, Y. Zhang, Z. Ren, and X. Cheng, "On protecting the data privacy of large language models (llms) and llm agents: A literature review," *High-Confidence Computing*, vol. 5, no. 2, p. 100300, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2667295225000042>
- [4] (2025, März) Openai data controls. OpenAI. [Online]. Available: <https://platform.openai.com/docs/guides/your-data>
- [5] (2025, März) Openai pricing. OpenAI. [Online]. Available: <https://platform.openai.com/docs/pricing>
- [6] (2025, März) Llmhub. IFS at OST Ostschweizer Fachhochschule. [Online]. Available: <https://llmhub.infs.ai/about/>
- [7] (2025, März) Ollama. Ollama. [Online]. Available: <https://ollama.com/>
- [8] H. Jin, Y. Zhang, D. Meng, J. Wang, and J. Tan, "A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods," *ArXiv*, vol. abs/2403.02901, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:268248096>
- [9] A. Barabanov and D. Makrushin, "Authentication and authorization in microservice-based systems: Survey of architecture patterns," *Voprosy kiberbezopasnosti*, pp. 32–43, 01 2020.
- [10] D. Fett, R. Küsters, and G. Schmitz, "The web sso standard openid connect: In-depth formal security analysis and security guidelines," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 2017, pp. 189–202.
- [11] (2025, April) Text search types - tsvector. PostgreSQL Development Group. [Online]. Available: <https://www.postgresql.org/docs/current/datatype-textsearch.html#DATATYPE-TSVECTOR>
- [12] D. G. S. Dr. Peter Hruschka. (2025) arc42 template. Abgerufen am 25.02.2025. [Online]. Available: <https://www.arc42.de/overview/>
- [13] S. Brown. (2025) C4 model. Abgerufen am 02.03.2025. [Online]. Available: <https://c4model.com/>
- [14] J. Palermo. (2013, August) Tag archives: onion architecture. Abgerufen am 18.03.2025. [Online]. Available: <https://jeffreypalermo.com/tag/onion-architecture/>

- [15] (2024, Oktober) Onion architecture in asp.net core. Code Maze. Abgerufen am 09.03.2025. [Online]. Available: <https://code-maze.com/onion-architecture-in-aspnetcore/>
- [16] R. C. Martin, *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Pearson Studium, 2017.
- [17] (2025, März) Directory structure. Abgerufen am 23.03.2025. [Online]. Available: <https://nuxt.com/docs/guide/directory-structure/nuxt>
- [18] (2025, März) Pinia store. Abgerufen am 23.03.2025. [Online]. Available: <https://pinia.vuejs.org/>
- [19] (2025, April) Components basics. Abgerufen am 10.04.2025. [Online]. Available: <https://vuejs.org/guide/essentials/component-basics>
- [20] (2025, Februar) Gitlab docs. GitLab. [Online]. Available: https://docs.gitlab.com/integration/openid_connect_provider/
- [21] (2025, Februar) Openid foundation. OpenID Foundation. [Online]. Available: <https://openid.net/developers/how-connect-works/>
- [22] (2025, April) Pdgpig. [Online]. Available: <https://github.com/UglyToad/PdfPig>
- [23] (2025, April) itextpdf: Pdf library. Apryse. [Online]. Available: <https://itextpdf.com/>
- [24] (2025, April) Pdfsharp. empira Software GmbH. [Online]. Available: <https://pdfsharp.com/>
- [25] (2025, April) Fluentemail. [Online]. Available: <https://github.com/lukencode/FluentEmail>
- [26] (2025, April) Mailkit. [Online]. Available: <https://github.com/jstedfast/MailKit>
- [27] (2025, Mai) Hostedservices. Microsoft. [Online]. Available: <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/host/hosted-services?view=aspnetcore-9.0&tabs=visual-studio>
- [28] (2025, Mai) Questpdf. [Online]. Available: <https://www.questpdf.com/>
- [29] (2025, Mai) Dinktopdf. [Online]. Available: <https://github.com/rdvojmoc/DinkToPdf>
- [30] (2025, Mai) Gotenberg. [Online]. Available: <https://gotenberg.dev/>
- [31] (2025, April) Introduction to understanding wcag 2.2. W3C Web Accessibility Initiative (WAI). Abgerufen am 25.05.2025. [Online]. Available: <https://www.w3.org/WAI/WCAG22/Understanding/intro#understanding-the-four-principles-of-accessibility>
- [32] mganss. (2025, Mai) Htmlsanitizer. [Online]. Available: <https://github.com/mganss/HtmlSanitizer>
- [33] (2024, März) Monorepo vs. polyrepo: How to choose. Buildkite. Abgerufen am 17.03.2025. [Online]. Available: <https://buildkite.com/resources/blog/monorepo-polyrepo-choosing/>
- [34] (2025, März) Gitflow workflow. Atlassian. [Online]. Available: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>
- [35] (2025, März) Swagger ui. SmartBear Software. [Online]. Available: <https://swagger.io/tools/swagger-ui/>
- [36] (2025, Juni) Doxygen: Code documentation. automated. doxygen. [Online]. Available: <https://www.doxygen.nl/>

- [37] (2025, März) Thinking aloud: The #1 usability tool. Nielsen Norman Group. [Online]. Available: <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool>

Glossar

Branch Coverage Ein Mass für die Qualität von Softwaretests, welches angibt, wie viele der möglichen Verzweigungen (z.B. in If-Else-Abzweigungen) innerhalb des Codes durch Tests ausgeführt wurden. Eine hohe Branch Coverage deutet darauf hin, dass die Tests viele verschiedene Entscheidungspfade im Code abgedeckt haben, was die Wahrscheinlichkeit von Fehlerhafter Logik verringert. [20](#), [66](#), [81](#)

Defence in Depth Defence in Depth (dt. Verteidigung in der Tiefe) bezeichnet ein Sicherheitsprinzip, bei dem mehrere unabhängige Schutzmassnahmen auf verschiedenen Ebenen eines Systems eingesetzt werden, um Angriffe abzuwehren oder ihre Auswirkungen zu minimieren. Ziel ist es, selbst dann Schutz zu bieten, wenn eine einzelne Massnahme versagt. [62](#)

Fully Dressed Format Das Fully Dressed Format ist eine detaillierte und strukturierte Vorlage zur Beschreibung von Use Cases in der Softwareentwicklung. Es enthält spezifische Abschnitte wie Akteure, Vorbedingungen, Hauptszenario und Erweiterungen, um ein umfassendes Verständnis des Use Cases zu ermöglichen. Im Vergleich zu den kürzeren Formaten "Brief" und "Casual", die weniger ausführlich sind, bietet das Fully Dressed Format eine tiefere Analyse und Struktur. [6](#)

Rational Unified Process (RUP) Ein iteratives und inkrementelles Vorgehensmodell für Softwareentwicklung, das aus den Phasen Inception, Elaboration, Construction und Transition besteht, mit besonderem Fokus auf eine langfristige Planung und die schrittweise Fertigstellung des Projekts. [69](#)

Single Sign-On Ermöglicht die Verwendung von mehreren Applikationen mit den gleichen Anmeldeinformationen, d.h. mit einem Benutzerkonto. [25](#)

Akronyme

AGPL GNU Affero General Public License. [42](#)

API Application Programming Interface. [24](#), [65](#), [68](#)

BA Bachelorarbeit. [1](#)

CQRS Command Query Responsibility Segregation. [106](#)

CSS Cascading Style Sheets. [154](#)

HTML Hypertext Markup Language. [68](#)

IFS Institut für Software. [24](#)

KI Künstliche Intelligenz. [24](#), [79](#)

LLM Large Language Model. [24](#), [73](#), [75](#)

MVP Minimum Viable Product. [1](#)

NFR Non-functional Requirements. [18](#), [67](#), [73](#), [82](#)

OST Ostschweizer Fachhochschule. [1](#)

SA Studienarbeit. [1](#), [40](#), [44](#)

SEO Search Engine Optimization. [102](#)

SSR Server Side Rendering. [102](#)

W3C World Wide Web Consortium. [45](#)

WCAG Web Content Accessibility Guidelines. [45](#)

XML Extensible Markup Language. [65](#), [68](#)

XSS Cross-Site-Scripting. [61](#)

9 Anhang

9.1 Tests und Qualität

9.1.1 Usability Tests

In diesem Abschnitt wird zunächst die übergeordnete Herangehensweise an die Usability-Tests erläutert. Im Anschluss werden die einzelnen Tests zu den jeweiligen Funktionalitäten inklusive der formulierten Erwartungen an deren Ergebnisse beschrieben. Als Letztes werden die aus den Testdurchführungen gewonnenen Erkenntnisse und Verbesserungen dokumentiert. Das Protokoll der Testdurchführungen ist in Abschnitt 9.1.5 zu finden.

Allgemeine Vorgehensweise

Die vorbereiteten Testfälle werden gemeinsam mit den Testpersonen durchgearbeitet. Dabei werden die Testpersonen aufgefordert, ihre Gedanken während der Bearbeitung möglichst laut mitzuteilen, sogenanntes „Thinking Aloud“ [37]. Alle relevanten Beobachtungen wie Ablauf, auftretende Unklarheiten, spontane Rückmeldungen und sonstiges Feedback werden während des Tests protokolliert. Auf Basis dieser Aufzeichnungen werden anschliessend gezielte Optimierungen und Verbesserungsvorschläge abgeleitet.

Testfälle Authentifizierung

Nr	Aufgabe	Erwartetes Resultat
1	Als neuer Benutzer des Portals möchten Sie erst einmal einen Benutzer-Account anlegen. Erstellen Sie ein Benutzerkonto.	1. Der Benutzer findet das Navigationselement im Header, welches zur „Login“-Seite führt. 2. Der Benutzer klickt auf die Schaltfläche „Einloggen mit Gitlab“. 3. Der Benutzer meldet sich mit seinem Gitlab-Konto an, sofern er bereits ein Konto hat. 4. Sollte er noch kein Konto haben, versteht er anhand des Informationstextes auf der Anmeldeseite von Gitlab, ob er befugt ist, ein Konto zu erstellen und wie er dafür vorgehen soll.
2	Da Sie im Institut für Software als Mitarbeiter angestellt sind, möchten Sie dies auch in Ihrem Profil konfigurieren. Passen Sie das Benutzerprofil entsprechend an.	1. Der Benutzer bemerkt das Navigationselement „Profil“. 2. Der Benutzer verwendet die Felder „Organisationseinheit“ und „Rolle“, um sich als Mitarbeiter beim Institut für Software einzutragen.

Tabelle 9.1: Testfälle Authentifizierung

Testfälle Stammdatenverwaltung

Nr	Aufgabe	Erwartetes Resultat
1	Als Administrator der Anwendung möchten Sie die Arbeitsergebnisse bearbeiten. Dazu möchten Sie im ersten Schritt herausfinden, welche Arbeitsergebnisse es aktuell gibt.	1. Der Benutzer bemerkt das Navigationsfeld „Administration“ und verwendet dieses, um zur Admin-Seite zu gelangen. 2. Der Benutzer findet anhand der Seitennavigation oder der Übersichtsseite die Tabelle, in welcher alle Arbeitsergebnisse abgebildet werden.
2	Die Hochschule verlangt nach dem neuen Arbeitsergebnis-Typ „Bachelorarbeit“. Erstellen Sie dieses neue Arbeitsergebnis.	1. Der Benutzer sieht und verwendet die Schaltfläche „Neues Arbeitsergebnis anlegen“. 2. Im Dialog „Neues Arbeitsergebnis anlegen“ gibt er den Namen im entsprechenden Feld ein und lässt den Status auf „aktiv“.
3	Leider kam es zu einem Missverständnis und der Arbeitsergebnis-Typ „Bachelorarbeit“ wird doch nicht benötigt. Löschen Sie das vorhin erstellte Arbeitsergebnis wieder.	1. Der Benutzer klickt auf das Mülleimer-Icon, um das entsprechende Element zu löschen. 2. Es öffnet sich ein Dialogfeld, welches vom Benutzer gelesen wird. Der Benutzer entschliesst sich anschließend dazu, das Element zu löschen.
4	Anscheinend kam es auch beim Arbeitsergebnis-Typ „Software“ zu einem Missverständnis. Löschen Sie auch diesen Typ.	1. Der Benutzer löscht das Arbeitsergebnis vom Typ Software (gleiches Vorgehen wie im letzten Schritt). 2. Der Benutzer erhält eine Fehlermeldung, die ihn darauf hinweist, dass dieses Arbeitsergebnis verwendet wird und daher nicht gelöscht werden kann. 3. Fragt man den Benutzer, wie er weiter vorgehen würde, wäre das Archivieren der nächste vorgeschlagene Schritt.
5	Da das Arbeitsergebnis bereits verwendet wird, ist das Löschen leider nicht mehr möglich. Daher ist es erforderlich, das Arbeitsergebnis zu archivieren. Archivieren Sie das Arbeitsergebnis.	1. Der Benutzer klickt auf die Schaltfläche „Bearbeiten“. 2. Im Dialog „Arbeitsergebnis bearbeiten“ ändert er den Status von „Aktiv“ auf „Archiviert“.
6	Nun möchten Sie kurz nachsehen, wie viele und welche Arbeitsergebnisse bereits archiviert wurden. Filtern Sie nach denjenigen Arbeitsergebnissen, die den Status „Archiviert“ haben.	1. Der Benutzer findet das Filter-Dropdown mit dem nach einem Status gefiltert werden kann. 2. Der Benutzer wählt den Status „Archiviert“ aus.

Tabelle 9.2: Testfälle Stammdatenverwaltung

Nr	Aufgabe	Erwartetes Resultat
7	Es interessiert Sie, ob es bereits ein Arbeitsergebnis mit dem Namen „Datensatz“ gibt. Suchen Sie nach diesem Arbeitsergebnis.	Der Benutzer bemerkt das globale Suchfeld oder das Suchfeld nach Namen und verwendet dieses, um das gesuchte Arbeitsergebnis zu finden.
8	Ein verärgerter Mitarbeiter hat Ihnen mitgeteilt, dass sein Lieblingskanton „Thurgau“ nicht vorhanden ist. Erstellen Sie die neue Region „Thurgau“.	- Der Benutzer navigiert zur Verwaltungsansicht der Regionen. - Der Benutzer verwendet die Schaltfläche „Neue Region anlegen“. Es öffnet sich ein Dialog „Neue Ortschaft anlegen“. - Der Benutzer füllt den Dialog mit den entsprechenden Angaben ab. - Als übergeordnete Region wählt der Benutzer intuitiv die Region „Schweiz“.
9	Der Kanton Thurgau soll neu ein eigenständiges Land sein. Passen Sie die Stammdaten (Regionen) des Programms entsprechend an.	- Der Benutzer sucht nach dem Kanton Thurgau mit der Suchleiste oder findet diesen, indem er die Kind-Elemente der Region „Schweiz“ begutachtet. - Der Benutzer betätigt die Bearbeiten-Schaltfläche und ändert die übergeordnete Region auf „Keine übergeordnete Region“ ab.
10	Öffnen Sie die Bearbeitungsansicht der Region „Schweiz“ und warten Sie anschliessend auf weitere Anweisungen. (Schweiz insgeheim bearbeiten).	- Der Benutzer erhält eine Concurrency Fehlermeldung und bemerkt dadurch, dass eine zeitgleiche Bearbeitung stattgefunden hat. - Der Benutzer folgt der Anweisung der Fehlermeldung, lädt die Daten erneut und sieht die Änderungen, die der Kollege (Fragesteller) gemacht hat.

Tabelle 9.3: Testfälle Stammdatenverwaltung Teil 2

Testfälle Projektverwaltung

Nr	Aufgabe	Erwartetes Resultat
1	Vor einiger Zeit haben Sie ein Forschungsprojekt zum Thema „KI“ mit dem Titel „KI in der Softwareentwicklung“ veröffentlicht. Das Abstract weist jedoch massive inhaltliche Fehler auf. Erstellen Sie eine neue Version des Projekts, um darin den Fehler zu korrigieren.	1. Der Benutzer sucht und findet das Projekt in der Sparte „Meine Projekte“. 2. Der Benutzer klickt auf den Button „Neufassung erstellen“. 3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird. 4. Der Benutzer nimmt in der neuen Version des Projekts die entsprechenden Änderungen vor.
2	Veröffentlichen Sie diese neue Version organisationsintern.	1. Der Benutzer wählt die Projektsichtbarkeit „Organisationseinheit“, falls diese noch nicht gewählt ist. 2. Der Benutzer betätigt den Button „Organisationseinheitsintern veröffentlichen“ und bestätigt den Dialog, der sich anschliessend öffnet. 3. Der Benutzer versteht, dass das vorherige Projekt nun überarbeitet wurde.
3	Ihr Chef teilt Ihnen mit, dass Ihr Forschungsprojekt ein grober Fehler aufweist und in diesem Zustand nicht präsentiert werden darf. Archivieren Sie das Projekt.	1. Der Benutzer sucht und findet das Projekt in der Sparte „Meine Projekte“. 2. Der Benutzer klickt auf den Button „Archivieren“. 3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird.
4	Es hat sich herausgestellt, dass der vermeintlich grobe Fehler nicht in Ihrem Forschungsprojekt, sondern im Projekt eines Kollegen vorlag. Ihr Chef hat sich bei Ihnen entschuldigt und meinte, man solle das Projekt umgehend wieder veröffentlichen. Sorgen Sie dafür, dass das Projekt wieder veröffentlicht wird.	1. Der Benutzer sucht und findet das Projekt in der Sparte „Meine Projekte“. 2. Der Benutzer klickt auf den Button „Projekt reaktivieren“. 3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird. 4. Der Benutzer startet den Kontroll- / Freigabeprozess.

Tabelle 9.4: Testfälle Projektverwaltung

Testfälle Projektsuche

Nr	Aufgabe	Erwartetes Resultat
1	Sie möchten mehr über die Projekte an der OST erfahren. Suchen Sie nach Projekten mit dem Arbeitsergebnis „Software“.	• Der Benutzer sucht nach „Software“ oder wählt im Filter Arbeitsergebnis: „Software“ aus.
2	Suchen Sie nun nach den Softwareprojekten mit dem Sponsor „Eureka“.	• Der Benutzer setzt zusätzlich den Filter „Sponsor: Eureka“ oder fügt den Suchbegriff „Eureka“ hinzu.
3	Sie haben von Ihrer Kollegin Julia Weber erfahren, dass sie erst kürzlich Projekte veröffentlicht hat. Versuchen Sie, diese zu finden.	• Der Benutzer sucht nach „Julia Weber“.
4	Setzen Sie Ihre Suche zurück und suchen Sie nach Projekten des Instituts für Architektur und des Instituts für Software, die „künstliche Intelligenz“ erwähnen.	1. Der Benutzer klickt auf „Suche zurücksetzen“ unterhalb der Filterleiste. 2. Er filtert dann nach den Departementen „Informatik“ und „Gesundheit“. 3. Benutzer sucht nach „künstliche Intelligenz“ oder verwendet das Schlagwort „Künstliche Intelligenz“.

Tabelle 9.5: Testfälle Projektsuche

Testfälle Benutzerverwaltung

Nr	Aufgabe	Erwartetes Resultat
1	Navigieren Sie zur Adminseite und ermitteln Sie, welche Aufgaben aktuell anstehen.	Der Benutzer gelangt zur Admin-Seite und sieht anhand der Badges, dass sowohl eine Kontoaktivierung als auch eine Änderungsanfrage bearbeitet werden sollten.
2	Der Benutzer „Benutzer Neu“ möchte ein Konto anlegen. Prüfen Sie seine Anfrage, weisen Sie ihm das Institut Architektur zu und vergeben Sie die Rolle „Mitarbeiter“.	- Der Benutzer öffnet die Seite „Neue Konten aktivieren“. - Findet den Benutzer „Benutzer Neu“ und klickt auf „Prüfen“. - Legt als Organisationseinheit „Institut Architektur“ und als Rolle „Mitarbeiter“ fest und aktiviert das Konto.
3	Der Benutzer „Livio Mauchle“ möchte Änderungen an seinem Konto vornehmen. Finden Sie die Anfrage und nehmen Sie sie an.	- Der Benutzer öffnet die Seite „Änderungsanfragen prüfen“. - Findet den Benutzer „Livio Mauchle“ und klickt auf „Prüfen“. - Nimmt die Änderungsanfrage an.
4	Der Benutzer „Thomas Bauer“ hat bislang keine Berechtigungen. Vergeben Sie ihm die Rolle „Mitarbeiter“.	- Der Benutzer öffnet die Seite „Benutzer verwalten“. - Findet den Benutzer „Thomas Bauer“ und klickt auf „Bearbeiten“. - Weist ihm die Rolle Mitarbeiter zu und speichert die Änderungen.
5	Die ehemalige Mitarbeiterin „Julia Weber“ hat die OST verlassen und soll keine Änderungen mehr vornehmen können. Sperren Sie ihr Benutzerkonto.	- Der Benutzer öffnet die Seite „Benutzer verwalten“ - Findet „Julia Weber“ - Klickt auf „Bearbeiten“, wählt „Benutzerkonto sperren“ und gibt eine sinnvolle Begründung ein.
6	Sechs Monate nach ihrem Austritt kehrt „Julia Weber“ als normale Mitarbeiterin ans Institut für Software zurück. Reaktivieren Sie ihr Konto.	- Der Benutzer öffnet die Seite „Gesperrte Benutzer“ - Findet „Julia Weber“ - Klickt auf „Bearbeiten“, weist erneut die Organisationseinheit „Institut für Software“ und die Rolle „Mitarbeiter“ zu und klickt auf „Benutzerkonto reaktivieren“.

Tabelle 9.6: Testfälle Benutzerverwaltung

Testfälle KI-Beschreibungen und Editor Erweiterung

Nr	Aufgabe	Erwartetes Resultat
1	Legen Sie ein neues Projekt an. Verschaffen Sie sich zunächst einen kurzen Überblick und erfassen Sie dann eine Projektbeschreibung.	1. Der Benutzer legt über „Projektverwaltung“, dann „Neues Projekt anlegen“ ein neues Projekt 2. Benutzer bemerkt den Hinweis „Aus angehängtem PDF-Dokument generieren“.
2	Versuchen Sie nun, eine Beschreibung anhand eines PDF-Dokuments automatisch zu generieren.	1. Der Benutzer klickt auf den Link „Aus angehängtem PDF-Dokument generieren“. 2. Er lädt ein PDF hoch. 3. Er verwendet die Schaltfläche zum Erzeugen der Beschreibung aus dem Dokument.
3	Die Beschreibung wurde erstellt. Sehen Sie, wie das fertige Projekt später aussehen wird?	• Der Benutzer findet die Schaltfläche „Zur Vorschau“ und betrachtet das Projekt in der Vorschauansicht.
4	Die Beschreibung wirkt noch unstrukturiert. Unterteilen Sie sie in die Abschnitte Problemstellung, Lösung und Fazit und formatieren Sie die Überschriften entsprechend.	• Der Benutzer wendet die Formatierungsoptionen für Überschriften korrekt an und gliedert den Text in die drei geforderten Abschnitte.
5	Fügen Sie im Abschnitt Lösung eine Aufzählung aller durchgeführten Schritte hinzu und heben Sie wichtige Wörter (fett, kursiv oder unterstrichen) hervor.	1. Der Benutzer nutzt die Listenfunktion für die Schrittauflistung. 2. Der Benutzer versteht, wie er Hervorhebungen (Fettschrift, Kursivschrift, Unterstreichung) anwenden kann.

Tabelle 9.7: Testfälle KI-Beschreibungen und Editor

Testfälle Dashboard Erweiterung

Nr	Aufgabe	Erwartetes Resultat
1	Finden Sie heraus, in welchen Regionen Projekte mit dem Sponsor „Eureka“ durchgeführt wurden und welches Arbeitsergebnis bei diesem Sponsor am häufigsten erzielt wurde.	1. Der Benutzer setzt den Filter Sponsor = Eureka. 2. Der Benutzer sieht die Statistiken Projekte nach Region sowie Projekte nach Arbeitsergebnis. 3. Anhand dieser Statistiken lassen sich die gesuchten Antworten finden.
2	Speichern Sie die aktuellen Filtereinstellungen, da diese für zukünftige Abfragen erneut benötigt werden.	1. Der Benutzer klickt auf „Lesezeichen hinzufügen“. 2. Er vergibt einen aussagekräftigen Namen. 3. Er sichert die aktuellen Filtereinstellungen.
3	Sie möchten nun herausfinden welche Organisationseinheiten bereits Projekte im Raum St. Gallen durchgeführt haben. Setzen sie erst alle Filter zurück und führen Sie anschliessend die Abfrage durch.	1. Der Benutzer klickt auf „Filter zurücksetzen“. 2. Anschliessend stellt er im Filter „Region“ den Wert „St. Gallen“ ein.
4	Rufen Sie nun Ihre zuvor gespeicherte Filterkombination wieder auf.	• Der Benutzer öffnet die Lesezeichen-Anzeige und wählt das entsprechende Lesezeichen aus.
5	Sie sind mit dem aktuellen Namen des Lesezeichens nicht ganz zufrieden. Ändern Sie den Namen des gespeicherten Lesezeichens.	1. Der Benutzer klickt auf das Stiftsymbol in der Lesezeichenliste. 2. Er trägt im Dialog einen neuen Namen ein und bestätigt diesen.
6	Das Lesezeichen wird nicht weiter benötigt. Entfernen Sie das Lesezeichen aus der Lesezeichenleiste.	1. Der Benutzer klickt auf das Mülleimer-Symbol (oder den entsprechenden Button). 2. Er bestätigt im Dialog das Löschen. 3. Das Lesezeichen wird entfernt.

Tabelle 9.8: Testfälle Dashboard Erweiterung

Aus Usability Tests resultierende Verbesserungen

Mangel	Verbesserung
Bei Projektsuche fällt die Funktion der Filterleiste zu wenig auf. Es wurde gewünscht, dass Filter standardmässig aufgeklappt dargestellt werden.	Der erste Filter (nach Arbeitsergebnis) wird neu standardmässig aufgeklappt dargestellt, um sichtbar zu machen, was sich hinter dem Filterdropdown verbirgt. Alle anderen bleiben zugeklappt, sodass die Filterleiste übersichtlich bleibt.
Die globale Suche in der Stammdatenverwaltung war unklar, wurde nicht verwendet und auf Nachfrage als unnötig empfunden.	Die globale Suche wurde in allen Stammdatentabellen entfernt, da sie dort nur einen geringen Nutzen hat. In der Benutzerverwaltung jedoch beibehalten, da sie dort auch als sinnvoll empfunden wurde.
Bei der Fehlermeldung, die beim Löschversuch von referenzierten Daten erscheint, war nicht klar, dass die Daten archiviert werden sollen, wenn das Popup zuvor durch den Benutzer nicht gelesen wurde.	Fehlermeldung wurde ausführlicher gestaltet und enthält neu einen expliziten Aufruf zum Archivieren.
Gemischtes Feedback zu Toast-Meldungen: Sie wurden teilweise gelobt, aber auch kritisiert, da sie teilweise als schlecht sichtbar empfunden wurden.	Die Dauer der Toasts wurde erhöht: allgemeine Meldungen von 4 auf 5 Sekunden, Fehlermeldungen von 5 auf 8 Sekunden.
Bei Auswahlfeldern war die spezielle Option „Keine übergeordnete Region/Schlagwort/Organisations-einheit“ nicht sonderlich auffällig.	Die spezielle Option wird neu in fetter Schrift dargestellt.
Dashboard der Statistiken wurde nicht bei jedem Testdurchlauf intuitiv gefunden.	Symbol wurde durch ein besser verständliches Icon für Statistiken ersetzt.
Inkonsistente Gross-/Kleinschreibung	Alle Texte wurden mittels Spellchecker geprüft.
In der Projekterfassung war der Ablauf für das Erstellen einer durch KI generierten Zusammenfassung unklar.	Die Verlinkung zur entsprechenden Funktion wurde weiter oben platziert, neu beschriftet und mit einem Informationstext ausgestattet. Zudem wurde beim Dateiupload eine kurze Beschreibung hinzugefügt und die Schaltfläche für die Erstellung der Zusammenfassung beschriftet.
In der Projekterfassung war der Zurück-Button unklar beschriftet.	Buttontext geändert zu „Zurück zu meinen Projekten“.
In der Projekterfassung konnten ungespeicherte Änderungen leicht verloren gehen, wenn ein Benutzer ohne Speichern zu einer anderen Seite navigiert.	Es wurde ein Warnhinweis beim Verlassen ohne Speichern eingeführt.
Bezeichnung des Navigationselements „Administration“ war missverständlich.	Element wurde umbenannt zu „Adminbereich“.

Tabelle 9.9: Verbesserungen durch Usability Tests Teil 1

Mangel	Verbesserung
Standardanzahl angezeigter Datensätze in Admin-Tabellen war zu gering.	Die Anzahl wurde von 5 auf 10 erhöht.
Das Dateiuploadfeld in der Projekterfassung war zu klein.	Das Feld wurde vergrößert.
Eine leichte Rotation der Word Cloud wurde als unpassend empfunden.	Rotation der Begriffe wurde entfernt.
Validierung bei Datum „Date From > Date To“ funktionierte nicht immer.	Das Problem wurde korrigiert.
Die Admin-Übersichtsseite wurde durch die vielen Elemente als unübersichtlich empfunden.	Die Seite wird neu anhand der Überschriften für Stammdaten- und Benutzerverwaltung unterteilt, um die Übersichtlichkeit zu verbessern.

Tabelle 9.10: Verbesserungen durch Usability Tests Teil 2

9.1.2 Massnahmen aus Code Review

In den folgenden Tabellen sind alle Mängel und Fragen dokumentiert, die im Rahmen eines Code Reviews geäußert wurden. Sehr kleine oder nur auf eine spezifische Codestelle bezogene Mängel wurden nicht aufgenommen, diese wurden aber trotzdem behandelt. Die aufgeführten Mängel wurden sinngemäss zusammengefasst und entsprechen daher nicht in jedem Fall wörtlich dem ursprünglichen Feedback. Teilweise handelt es sich auch eher um Empfehlungen als um tatsächliche Mängel, zur Vereinfachung wurden jedoch alle Rückmeldungen einheitlich behandelt. Die Mängel sind unterteilt in solche, die vom Entwicklerteam behoben werden konnten (siehe Tabelle 9.11 und folgende), und solche, die aus unterschiedlichen Gründen nur teilweise oder gar nicht behoben wurden (siehe Tabelle 9.15).

Code Review: Behobene Mängel

Nr	Mangel / Vorschlag	Lösungsansatz / Verbesserung
1	Problem mit Docker: Wenn die Datenbank nicht initialisiert werden kann, startet das Backend nicht, da es von der Datenbank abhängig ist. Für Entwickler ist dies nicht unmittelbar ersichtlich.	Obwohl die Abhängigkeit im Docker-Compose-File bereits definiert war, konnte das Backend starten, bevor die Datenbank tatsächlich einsatzbereit war. Um dieses Problem zu beheben, wurde ein Healthcheck für die Datenbank ergänzt, der sicherstellt, dass sie wirklich erreichbar ist. Ist dies nicht der Fall erhält der Entwickler beim Start des Containers in der Konsole eine Fehlermeldung, die darauf hinweist, dass die Datenbank nicht gestartet werden konnte. Zudem wird im UI ein Netzwerkfehler angezeigt, der darauf hinweist, dass die Verbindung zum Server nicht hergestellt werden konnte, sofern das Backend nicht erreicht werden kann.
2	Im Docker den Postgres Standardport zu verwenden, kann für die lokale Entwicklung mühsam sein, es empfiehlt sich daher, diesen anzupassen.	Neu wird Postgres über Port 5433 anstelle des Standardports 5432 exponiert.
3	Gitignore sollte im Backend auf Solution-Ebene sein.	Dies wurde korrigiert.
4	Formatter funktioniert in Visual Studio nicht. Funktioniert dies für Rider?	Der Formatter funktioniert für Rider. Die Problematik wurde im Readme für andere Entwickler festgehalten.
5	Das Backend basiert derzeit auf .NET 8. Ein Umstieg auf .NET 9 wäre beispielsweise aufgrund der Einführung von Primary Constructors sinnvoll.	Das Projekt wurde auf .NET 9 migriert. Primary Constructors wurden schrittweise eingeführt und werden inzwischen überall verwendet.
6	Überarbeitung der Architektur: Es empfiehlt sich, das Backend in einzelne Projekte aufzuteilen (API, Infrastructure, Domain, Services evtl. Queries), diese klar voneinander zu trennen und Abhängigkeiten zu vermeiden.	Das Backend wurde in 4 Projekte (API, Infrastructure, Domain und Services) aufgeteilt. Weitere in diesem Rahmen vorgeschlagene Verbesserungen wie z.B. dass Repositories keine DTO's mehr zurückgeben, eine korrekte Projektzuordnung einzelner Komponenten etc. wurden umgesetzt.
7	Unbenutzte Referenzen, Source referenziert Moq.	Referenzen wurden geprüft, unbenutzte wurden entfernt.
8	Repository Wrapper verstösst gegen das Interface Segregation Prinzip.	Repository Wrapper wurde entfernt.
9	Warum wird Server Side Rendering (SSR) verwendet?	Wird verwendet, da Search Engine Optimization (SEO) zu Beginn der Studienarbeit als relevanter Aspekt der Anwendung galt. Da jedoch das vom OST ResearchNavigator unabhängig laufende Projekt der Publikationsdatenbank möglicherweise dafür verwendet wird, um Projekte der OST für Externe zugänglich zu machen, ist dieses Thema wieder etwas in den Hintergrund gerückt.

Tabelle 9.11: Code Review - Feedback und Massnahmen Teil 1/4

Nr	Mangel / Vorschlag	Lösungsansatz / Verbesserung
10	Im Frontend wird derzeit die Version „latest“ von Vue verwendet, was potenziell zu Breaking Changes bei zukünftigen Updates führen kann.	Es wurde darauf geachtet, im gesamten Projekt auf „latest“ zu verzichten. Stattdessen wurden feste Versionen mit leichtem Spielraum bei den Unterversionen definiert, um Stabilität zu gewährleisten.
11	Ein Scan des Frontends hat mehrere potenzielle Schwachstellen in verwendeten Abhängigkeiten aufgezeigt.	Die identifizierten Schwachstellen wurden behoben. Es wurde darauf geachtet, regelmässige Sicherheitsüberprüfungen durchzuführen, betroffene Pakete zu aktualisieren und insbesondere zum Projektabschluss sicherzustellen, dass keine bekannten Verwundbarkeiten mehr vorhanden sind.
12	TypeScript-Interfaces und -Typen, die sowohl im Frontend als auch im Backend benötigt werden, werden derzeit doppelt gepflegt. Dies stellt eine potenzielle Fehlerquelle dar und erhöht den Wartungsaufwand. Es empfiehlt sich, ein Tool wie Nswag einzusetzen.	Zur Reduktion von Redundanz und zur Erhöhung der Konsistenz wird nun Nswag eingesetzt, um TypeScript-Typen, Interfaces und Enums automatisch aus der OpenAPI-Spezifikation des Backends zu generieren.
13	Im Frontend befinden sich noch vereinzelt Texte, die nicht in die Übersetzung (i18n) eingebunden sind.	Das Frontend wurde erneut überprüft, und alle nicht übersetzten Texte wurden in die i18n-Konfiguration aufgenommen.
14	Im Frontend wurde bisher kein State-Container verwendet.	Statecontainer werden nun, wo sinnvoll, gezielt eingesetzt.
15	Tests im Frontend sind wünschenswert.	Im Frontend wurden Tests für Utility-Methoden, State-Container und, wo sinnvoll, auch für Composables implementiert.
16	Nach dem Deployment der Anwendung werden Seiten nur „lazy“ geladen, was zu einer langsamen Darstellung und einer schlechteren Benutzererfahrung führt.	Das Problem wurde analysiert und behoben; Seiten werden nun im Voraus geladen, um eine flüssigere Benutzererfahrung zu gewährleisten.
17	In TypeScript sollte der any-Typ vermieden werden. Falls er dennoch verwendet wird, sollte dies im Code begründet sein.	Die Verwendung von any wurde soweit möglich entfernt. Wo der Einsatz unvermeidbar war, wurden entsprechende Begründungen im Code ergänzt. In einigen Fällen wurde any durch unknown ersetzt, um weiterhin Type-Checking zu ermöglichen.
18	Caching wird an Stellen verwendet, an denen Änderungen (z.B. an Metadaten) zu Inkonsistenzen führen können.	Das Caching wurde überarbeitet, an kritischen Stellen (z.B. Filter) entfernt und nur dort beibehalten, wo keine direkten Konflikte zu erwarten sind (z.B. beim Laden von Organisationseinheiten in weniger kritischen Komponenten).
19	In RepositoryBase sowie auch anderen Repositorien werden Schnittstellen exponiert, die teilweise nicht gebraucht werden.	Schnittstellen von RepositoryBase sowie anderen Repositorien es wurden überprüft und so weit wie möglich reduziert.

Tabelle 9.12: Code Review - Feedback und Massnahmen Teil 2/4

Nr	Mangel / Vorschlag	Lösungsansatz / Verbesserung
20	Im Code gibt es teilweise Inkonsistenzen, unter anderem betreffend Namespaces, CancellationTokens oder Primary Constructors. Diese sollten behoben werden.	Mithilfe von Riders Code-Inspection-Tool wurden der Sourcecode analysiert und die beschriebenen Inkonsistenzen behoben. In einer .editorconfig Datei wurden die Einstellungen für die Code-Analyse festgehalten und sinnvoll angepasst. Ausnahmen in der Analyse wurden mit Kommentaren versehen.
21	Es empfiehlt sich, in den Repositories Postgreserweiterungen zu verwenden.	Es wurden Postgreserweiterungen für den Umgang mit dem Typ „NpgsqlVector“, der für die Projektsuche verwendet wurde, eingesetzt.
22	Sealed ist für dies Art von Anwendung nicht unbedingt nötig.	Sealed wurde entfernt.
23	NPM Pakete sollen, wenn einfach möglich, auf der aktuellen Major-Version und wenn nicht ohne Probleme möglich auf der aktuellen Minor-Version gehalten werden.	Gegen Projektende wurde versucht, alle Pakete auf die aktuellste Major-Version zu aktualisieren. Dies führte jedoch bei vielen Paketen zu Fehlern, die sich nicht in angemessener Zeit beheben liessen. Durch die Verwendung des Caret-Zeichens wird sichergestellt, dass Patch- und Minor-Versionen automatisch installiert werden.
24	Es wäre wünschenswert, wenn die KI-Modelle von OpenAI und dem LLMHum via Config austauschbar wären.	Wurde entsprechend umgesetzt, das Modell und der Provider lassen sich flexibel in der .env Konfigurationsdatei austauschen.
25	Die Datei Program.cs ist deutlich umfangreicher geworden. Zur besseren Lesbarkeit sollte der Code in geeignete Extension Methods ausgelagert werden.	Die Auslagerung in Extension Methods wurde wie besprochen umgesetzt.
26	Es ist sicherer, wenn Tokens in Cookies und nicht im Local-Storage gespeichert werden.	Neu werden Tokens in Cookies gespeichert.
27	Die Verwendung von LINQ oder Extension Method Syntax sollte über das gesamte Projekt gesehen einheitlich sein.	Es wurde entschieden und im Readme dokumentiert, dass die Extension Method Syntax verwendet werden soll. Zudem wurde versucht, mittels Konfiguration im .editorconfig File diese Regel während des Lintings zu forcieren. Die Erkennung hat aber leider nicht wie gewünscht funktioniert.

Tabelle 9.13: Code Review - Feedback und Massnahmen Teil 3/4

Nr	Mangel / Vorschlag	Lösungsansatz / Verbesserung
28	Weitere Verbesserung: • Bei Logging Formatier-Funktionalität mit Template und dann Variable verwenden • Teilweise besteht Chaos mit Nullable-Werten • Private Methoden sollten wenn möglich, auch static sein • Überall, wo möglich, Cancellation-Token verwenden • Unbenutzte Variablen entfernen • In Dockerfile auf Gross- / Kleinschreibung von Keywords achten • Im Frontend leere Script- und Style-Blöcke weglassen • Diverse kleine Unschönheiten im Frontend Code	Diese Unschönheiten wurden geprüft und verbessert.

Tabelle 9.14: Code Review - Feedback und Massnahmen Teil 4/4

Code Review: Nicht oder nur teilweise behobene Mängel und Vorschläge

Nr	Mangel / Vorschlag	Begründung / Massnahmen
1	Im Backend wäre der Einsatz von Command Query Responsibility Segregation (CQRS) eine interessante Option gewesen.	Aufgrund des im Verhältnis zum erwarteten Nutzen zu hohen Aufwands wurde jedoch darauf verzichtet.
2	Im Frontend fehlt eine Typprüfung für das i18n-Übersetzungsschlüssel.	Verschiedene Ansätze zur Integration mit dem Paket @nuxtjs/i18n wurden erprobt, führten jedoch zu Problemen oder erwiesen sich als nicht praktikabel. Daher wurde aus Priorisierungsgründen auf eine Umsetzung verzichtet.
3	Die Verwendung von switch-case deutet möglicherweise auf Verbesserungspotenzial durch ein geeignetes Entwurfsmuster hin. Eine Überprüfung wäre sinnvoll.	Aufgrund begrenzter Zeit und niedriger Priorität wurde auf eine nähere Analyse verzichtet.
4	Sehr grosse Komponenten im Frontend (über 300 Zeilen) sollten zur besseren Wartbarkeit aufgeteilt und verkleinert werden.	Während einige Komponenten entsprechend angepasst wurden, war ein Refactoring anderer Komponenten aus Zeitgründen nicht mehr möglich.
5	Der Domain Layer sollte von EF Core entkoppelt sein.	Da in den Entitäten Data Annotations verwendet werden, besteht derzeit eine Abhängigkeit zu EF Core. Eine Umstellung auf die Fluent API hätte die Lesbarkeit beeinträchtigt und wurde aus Zeitgründen nicht umgesetzt.

Tabelle 9.15: Code Review - nicht oder nur teilweise behobene Mängel

9.1.3 System Tests - Finale Durchführung

System Test Protokoll

Vorbereitung

- ✓ Aktuellste Version des Systems ist deployed.
- ✓ Adminbenutzerkonto für die Vergabe von Benutzerrechten ist erfasst.

Navigationsleiste

- ✓ OST-Logo sowie die Buttons für die Navigation werden korrekt angezeigt.
- ✓ Ohne Anmeldung wird Schaltfläche "Login" angezeigt, mit Anmeldung wird Schaltfläche "Abmelden" angezeigt.
- ✓ Navigationsschaltflächen reagieren bei Klick erwartungsgemäss.

Erfassen

Vorbedingung: Angemeldet als Mitarbeiter oder Leiter

- ✓ Über die Seite "Projektverwaltung" kann mittels der Schaltfläche "Neues Projekt anlegen" ein neues Projekt erfasst werden.
- ✓ Über die Navigationsleiste kann mittels "Neues Projekt anlegen" ein neues Projekt erfasst werden.
- ✓ Erfassungsformular wird korrekt dargestellt.
- ✓ Ein angelegtes Projekt ist persistiert und wird nach dem zurücknavigieren zu "Meine Projekte" in der Resultatliste angezeigt.
- ✓ In der Resultatliste der Suche auf der Seite "Projektverwaltung" werden für Projekte ohne Titel oder Beschreibung entsprechende Platzhalter angezeigt.
- ✓ Das Erstellungsdatum wird korrekt angezeigt.

- ✓ Durch einen Klick auf ein Projekt in der Projektverwaltung wird die Bearbeitungsansicht angezeigt.
- ✓ Es werden die Schaltflächen "Zurück", "Zur Vorschau", "Speichern" und "Löschen" korrekt dargestellt.
- ✓ Durch einen Klick auf "Löschen" öffnet sich ein Dialog zur Bestätigung des Löschvorgangs.
- ✓ Wird das Löschen bestätigt, wird das Projekt gelöscht. Der Benutzer erhält daraufhin eine visuelle Rückmeldung.
- ✓ Durch einen Klick auf "Speichern" wird das Projekt gespeichert und der Benutzer erhält eine visuelle Rückmeldung.
- ✓ Wird eine Änderung vorgenommen und versucht das Projekt zu verlassen, ohne diese Änderung zu speichern, so öffnet sich ein Dialog, der darauf hinweist, dass noch ungespeicherte Änderungen vorliegen.
- ✓ Informationen können in allen Feldern erfasst werden.
- ✓ Werden Informationen in Pflichtfeldern erfasst und anschliessend wieder gelöscht, so wird dies als Fehlermeldung angezeigt.
- ✓ Felder mit einer maximalen Zeichenlänge (Kurzbeschreibung und Beschreibung des Projekts) zeigen eine entsprechende Fehlermeldung an, wenn die maximale Zeichenlänge überschritten wird.
- ✓ Personen können über die Schaltfläche "Person hinzufügen" gesucht werden.
- ✓ Wird keine Person gefunden, kann eigenständig eine Person erfasst werden.
- ✓ Der Dialog zur Personenerfassung lässt keine Erfassung ohne Namen oder Rolle zu.
- ✓ Erfasste Personen werden in der Tabelle korrekt angezeigt und können über diese bearbeitet oder gelöscht werden.
- ✓ Organisationseinheiten können gesucht und hinzugefügt werden.
- ✓ Organisationseinheiten können über die Tabelle entfernt werden.
- ✓ Schlagwörter können gesucht und hinzugefügt werden.
- ✓ Schlagwörter können über die Tabelle entfernt werden.
- ✓ Erfassen von Start und Enddatum funktioniert. Ein Enddatum kann nicht vor einem Startdatum liegen.
- ✓ Erfassen von "Art des Projektergebnisses", "Sponsor des Projekts", "Sprache" und "Region" funktioniert.
- ✓ Links können über einen Dialog erfasst werden.
- ✓ Entspricht ein Link nicht einem korrekten URL-Schema, so wird dies dem Benutzer unterhalb der Tabelle angezeigt.
- ✓ Durch die Tabelle können Links bearbeitet und gelöscht werden.
- ✓ Im Abschnitt "Diverses" können weitere Metadaten optional erfasst werden.

- ✓ Das Projekt wird in der Vorschau korrekt angezeigt.

Erfassen von Medien

- ✓ Hochladen von Bild funktioniert korrekt.
- ✓ Hochladen von Videos funktioniert korrekt.
- ✓ Hochladen von Dokument funktioniert korrekt.
- ✓ Hochladen von Titelbild funktioniert korrekt.
- ✓ Hochgeladene Medien können mittels der Tabelle gelöscht werden.

Abstract und Kurzbeschreibung Generieren

- ✓ Der Link "Zum Dateiupload" im Bereich der Beschreibungen bringt den Benutzer direkt zum Dateiupload.
- ✓ Wird ein PDF-Dokument hochgeladen erscheint eine Schaltfläche, mit der die Beschreibungen generiert werden können.
- ✓ Nach einem Klick auf die Schaltfläche wird der Benutzer zurück zum Erfassungsfeld geleitet.
- ✓ Abstract und Kurzbeschreibung werden korrekt generiert.

Abstract Editor

- ✓ Die Bearbeitungsfunktionen für "Überschriften", "Fett", "Kursiv", "Unterstrichen", "Nummerierte Liste" und "Link" funktionieren korrekt.
- ✓ Das erstellte Abstract ist in der Vorschau korrekt formatiert.

Validierungsprozess

Organisationsinterne Veröffentlichung und Einreichung zur fachlichen Prüfung

- ✓ Ohne Eingabe einer Sichtbarkeit werden Buttons für die Veröffentlichung als deaktiviert angezeigt.
- ✓ Mit Setzen von Sichtbarkeit = "Organisationseinheit" wird der Button "Organisationseinheit-intern veröffentlichen" aktiviert.

- ✓ Sind noch nicht alle Pflichtfelder erfasst oder Eingaben ungültig, erscheint beim Klick auf die angezeigten Buttons für die Veröffentlichung eine Meldung, die auf die ungültigen Eingaben hinweist.
- ✓ Durch Klick auf den Button "Organisationseinheit-intern veröffentlichen" wird das Projekt als "Organisationseinheit-intern" veröffentlicht.
- ✓ Ein vollständig erfasstes Projekt mit Sichtbarkeit = "OST" kann durch Klick auf "Zur fachlichen Prüfung einreichen" eingereicht werden.
 - ✓ Es öffnet sich ein Dialog zur Bestätigung des Einreichvorgangs.
 - ✓ Die Einreichung kann nur dann geschehen, wenn ein Prüfer gewählt wurde.
 - ✓ Der Ersteller darf sich nicht selbst als Prüfer wählen können.

Fachliche Prüfung

- ✓ Das zur fachlichen Prüfung eingereichte Projekt wird bei der Prüfperson unter "Projekte prüfen" angezeigt.
- ✓ Die Prüfperson erhält eine E-Mail.
- ✓ Die Prüfperson kann das Projekt bearbeiten und speichern.
- ✓ Die Prüfperson kann das Projekt nicht löschen.
- ✓ Der Prüfer wird über seine Aufgabe als Prüfer informiert.
- ✓ Der angezeigte Projektstatus lautet "Fachliche Prüfung"
- ✓ Die Prüfperson kann das Projekt ablehnen und zurückweisen.
 - ✓ Beim Ablehnen öffnet sich ein Dialog in dem die Ablehnung begründet werden muss. Ohne Begründung kann das Projekt nicht abgelehnt werden.
 - ✓ Nach dem Zurückweisen sieht der Ersteller das Projekt wieder in der Projektverwaltung.
 - ✓ Der Ersteller sieht die Begründung der Ablehnung.
- ✓ Mit der gewünschten Sichtbarkeit = "OST" kann der Prüfer das Projekt OST-intern veröffentlichen.
 - ✓ Die OST-interne Veröffentlichung muss mittels Dialog bestätigt werden.
- ✓ Mit der gewünschten Sichtbarkeit = "Öffentlich" wird die Schaltfläche für das Einreichen zur Kommunikationsprüfung angezeigt.
 - ✓ Das Einreichen zur Kommunikationsprüfung muss mittels Dialog bestätigt werden.

Kommunikationsprüfung

- ✓ Das zur Kommunikationsprüfung eingereichte Projekt wird bei Benutzern, die als Kommunikationsprüfer eingetragen sind unter "Projekte prüfen" angezeigt.
- ✓ Der Prüfer wird über seine Aufgabe als Prüfer informiert.
- ✓ Der angezeigte Projektstatus lautet "Kommunikationsprüfung"
- ✓ Der Prüfer kann das Projekt bearbeiten und speichern.
- ✓ Der Prüfer kann das Projekt nicht löschen.
- ✓ Der Prüfer kann das Projekt ablehnen und zurückweisen.
 - ✓ Beim Ablehnen öffnet sich ein Dialog, in dem die Ablehnung begründet werden muss. Ohne Begründung kann das Projekt nicht abgelehnt werden.
 - ✓ Nach dem Zurückweisen sieht der Ersteller das Projekt wieder in der Projektverwaltung.
 - ✓ Der Ersteller sieht die Begründung der Ablehnung.
- ✓ Der Prüfer kann das Projekt Veröffentlichen.
 - ✓ Die Veröffentlichung muss mittels Dialog bestätigt werden.

Veröffentlichte Projekte verwalten

- ✓ Bei veröffentlichten Projekten werden 2 Schaltflächen angezeigt, eine um das Projekt zu archivieren und eine um eine Neufassung zu erstellen.
- ✓ Durch einen Klick auf "Neufassung erstellen" öffnet sich ein Dialog, wird dieser mit "Erstellen", öffnet sich eine neue Version.
- ✓ Die neue Version kann bearbeitet werden.
- ✓ Die neue Version kann durch den Benutzer wieder veröffentlicht werden (erfordert dieselben Prüfschritte wie ein neues Projekt).
- ✓ Durch das Veröffentlichen der neuen Version wird die alte Version für den Benutzer nicht mehr sichtbar und durch die neue Version ersetzt.
- ✓ Durch einen Klick auf "Archivieren" öffnet sich ein Dialog, wird dieser durch "Archivieren" bestätigt, ändert das Projekt seinen Status auf "Archiviert".
- ✓ Das archivierte Projekt ist in der Projektsuche (Hauptseite) nicht mehr auffindbar.
- ✓ Das archivierte Projekt verfügt über eine Schaltfläche "Projekt reaktivieren"
- ✓ Durch einen Klick auf "Projekt reaktivieren" öffnet sich ein Dialog, wird dieser mit "Reaktivieren" bestätigt, wechselt das Projekt den Status auf Entwurf.
- ✓ Das reaktivierte Projekt kann bearbeitet werden und anschliessend wieder veröffentlicht werden (erfordert dieselben Prüfschritte wie ein neues Projekt).

Startseite: Forschungsprojekte der OST

- ✓ Anzahl Treffer wird korrekt angezeigt und entspricht der Menge an gefundenen Projekten.
- ✓ Kacheln zur Darstellung der Ergebnisse werden korrekt angezeigt.

Filtern

Unauthorisierter Benutzer (Startseite)

- ✓ Filter nach Sichtbarkeit wird nicht angezeigt.
- ✓ Projekte mit Sichtbarkeitsstatus OST und intern werden nicht gelistet, es werden nur Projekte mit Status "Öffentlich" gefunden.
- ✓ Filterkombinationen ohne Treffer zeigt Text "Keine Treffer gefunden." an.

Mit Login

- ✓ Filtern nach Arbeitsergebnis liefert korrekte Resultate.
- ✓ Filtern nach Schlagwörtern liefert korrekte Resultate.
- ✓ Filtern nach Orte liefert korrekte Resultate.
- ✓ Filtern nach Sponsor liefert korrekte Resultate.
- ✓ Filtern nach Durchführungszeitraum liefert korrekte Resultate.
- ✓ Filtern nach Datum der Veröffentlichung liefert korrekte Resultate.
- ✓ Filtern nach Organisationseinheiten liefert korrekte Resultate.
- ✓ Filtern nach Departemente liefert korrekte Resultate.
- ✓ Filtern nach Sichtbarkeit liefert korrekte Resultate.
- ✓ Anzahl gesetzte Filter werden bei ausgeklapptem Filter als Badge und bei eingeklapptem Filter als Tags angezeigt.
- ✓ Sind bei einer einzigen Filterleiste mehr als 10 Filtern gesetzt, werden keine Tags mehr angezeigt.
- ✓ Durch einen Klick auf das x lassen sich die Tags und die damit verbundenen Filtereinstellungen entfernen.
- ✓ Gesetzte Filter lassen sich über den Button "Filter zurücksetzen" zurücksetzen.
- ✓ Filterleiste lässt sich über den Button "Alle Einklappen" einklappen.

Suchen

Unauthorisierter Benutzer (Startseite)

- ✓ Suche nach Titel, Autor, Schlagwörter, Arbeitsergebnis, Sponsor, Region, Organisationseinheiten, Metadaten und Abstract und Kurzbeschreibung liefert korrekte Resultate.
- ✓ Suche nach Text in angehängtem PDF-Dokument funktioniert.
- ✓ Suche nach mehreren Wörtern schränkt die Suche weiter ein.
- ✓ Suche funktioniert auch in kombination mit setzen von Filtern.
- ✓ Suche nach nicht existierendem Begriff zeigt Text "Keine Treffer gefunden" an.
- ✓ Suche ist nicht case sensitiv.
- ✓ Suche mit Wortanfängen funktioniert korrekt (z.B. "Simo" findet "Simon").
- ✓ Gesuchte Begriffe werden in der Projektleiste farbig hervorgehoben, sofern sie darin enthalten sind.

Projekt-Detailansicht

- ✓ Die Seitennavigation wird korrekt dargestellt.
- ✓ Die Seitennavigation bringt den Benutzer zu den entsprechenden Abschnitten.
- ✓ Im Bereich "Überblick" werden Daten zu Durchführungszeitraum, Arbeitsergebnis, Schlagwörter, Projektdateien und Mitwirkenden korrekt angezeigt.
- ✓ Das Thumbnail (sofern vorhanden) wird korrekt dargestellt.
- ✓ Projektdateien (sofern vorhanden) lassen sich über einen Klick öffnen.
- ✓ Im Bereich "Beschreibung" wird die Beschreibung korrekt dargestellt.
- ✓ Hochgeladene Medien werden auf der Projektdetailseite korrekt angezeigt.
- ✓ Bilder (sofern vorhanden) werden als Slideshow dargestellt.
- ✓ Videos (sofern vorhanden) können durch einen Videoplayer abgespielt werden.
- ✓ Im Bereich "Details" werden Beteiligte Organisationseinheiten, Sponsor, Sprache und Veröffentlichungsdatum dargestellt.
- ✓ Sofern das Projekt über weitere optionale Inhalte verfügt, werden auch diese korrekt dargestellt.
- ✓ Nach Klick auf einen Link (optionales Element) wird der Benutzer auf die entsprechende Seite weitergeleitet.

- ✓ Mit einem Klick auf “Als PDF exportieren” lässt sich ein PDF der Beschreibung generieren.

Dashboard

- ✓ Jedes Diagramm im Dashboard wird korrekt dargestellt.
- ✓ Das Anwenden der Filter führt zu einer korrekten Aktualisierung der Diagramme.
- ✓ Sind durch das Anwenden eines Filters keine Daten zur Anzeige vorhanden, wird dies dem Benutzer entsprechend mitgeteilt.
- ✓ Aktuelle Filtereinstellung lassen sich als Lesezeichen speichern.
- ✓ Ohne die Angabe eines Namens können Lesezeichen nicht gespeichert werden.
- ✓ Der Benutzer kann in der Ansicht aller Lesezeichen durch einen Klick zwischen den einzelnen Lesezeichen wechseln.
- ✓ Lesezeichen können umbenannt werden.
- ✓ Lesezeichen können gelöscht werden.
- ✓ Wird eine Filtereinstellung gewählt, zu der es ein gespeichertes Lesezeichen gibt, so wird dieses Lesezeichen automatisch als aktiv angezeigt.

Admin

- ✓ Durch einen Klick auf “Adminbereich” gelangt der Benutzer auf eine Übersichtsseite, die alle administrativen Funktionen auflistet.

Stammdatenverwaltung

- ✓ Arbeitsergebnisse können angelegt und bearbeitet werden, entsprechende Änderungen sind in der Filterleiste, sowie auch in der Projekterfassung sichtbar.
- ✓ Arbeitsergebnisse können gelöscht werden, sofern diese noch in keinem Projekt verwendet wurden.
- ✓ Arbeitsergebnisse, die bereits in einem Projekt verwendet werden, werfen eine Fehlermeldung beim Versuch diese zu löschen.
- ✓ Arbeitsergebnisse können archiviert werden, dadurch sind diese in der Filterleiste und in der Projekterfassung nicht mehr auswählbar, werden aber in bestehenden Projekten weiterhin angezeigt.

- ✓ Metadaten können angelegt und bearbeitet werden, entsprechende Änderungen sind in der Projekterfassung sichtbar.
- ✓ Metadaten können gelöscht werden, sofern diese noch in keinem Projekt verwendet wurden.
- ✓ Metadaten, die bereits in einem Projekt verwendet werden, werfen eine Fehlermeldung beim Versuch diese zu löschen.
- ✓ Metadaten können archiviert werden, dadurch sind diese in der Projekterfassung nicht mehr ausfüllbar.
- ✓ Organisationseinheiten vom Typ Departement können angelegt und bearbeitet werden, entsprechende Änderungen sind in der Filterleiste sichtbar.
- ✓ Organisationseinheiten vom Typ Institut können angelegt und bearbeitet werden, entsprechende Änderungen sind in der Filterleiste, sowie auch in der Projekterfassung sichtbar.
- ✓ Organisationseinheiten können gelöscht werden, sofern diese noch in keinem Projekt verwendet wurden.
- ✓ Organisationseinheiten, die bereits in einem Projekt verwendet werden, werfen eine Fehlermeldung beim Versuch diese zu löschen.
- ✓ Organisationseinheiten können archiviert werden, dadurch sind diese in der Filterleiste und in der Projekterfassung nicht mehr auswählbar, werden aber in bestehenden Projekten weiterhin angezeigt.
- ✓ Regionen können angelegt und bearbeitet werden, entsprechende Änderungen sind in der Filterleiste, sowie auch in der Projekterfassung sichtbar.
- ✓ Regionen können gelöscht werden, sofern diese noch in keinem Projekt verwendet wurden.
- ✓ Regionen, die bereits in einem Projekt verwendet werden, werfen eine Fehlermeldung beim Versuch diese zu löschen.
- ✓ Regionen können archiviert werden, dadurch sind diese in der Filterleiste und in der Projekterfassung nicht mehr auswählbar, werden aber in bestehenden Projekten weiterhin angezeigt.
- ✓ Schlagwörter können angelegt und bearbeitet werden, entsprechende Änderungen sind in der Filterleiste, sowie auch in der Projekterfassung sichtbar.
- ✓ Schlagwörter können gelöscht werden, sofern diese noch in keinem Projekt verwendet wurden.
- ✓ Schlagwörter, die bereits in einem Projekt verwendet werden, werfen eine Fehlermeldung beim Versuch diese zu löschen.
- ✓ Schlagwörter können archiviert werden, dadurch sind diese in der Filterleiste und in der Projekterfassung nicht mehr auswählbar, werden aber in bestehenden Projekten weiterhin

angezeigt.

- ✓ Sponsoren können angelegt und bearbeitet werden, entsprechende Änderungen sind in der Filterleiste, sowie auch in der Projekterfassung sichtbar.
- ✓ Sponsoren können gelöscht werden, sofern diese noch in keinem Projekt verwendet wurden.
- ✓ Sponsoren, die bereits in einem Projekt verwendet werden, werfen eine Fehlermeldung beim Versuch diese zu löschen.
- ✓ Sponsoren können archiviert werden, dadurch sind diese in der Filterleiste und in der Projekterfassung nicht mehr auswählbar, werden aber in bestehenden Projekten weiterhin angezeigt.
- ✓ Zu einem Sponsor kann ein Folgesponsor angelegt werden.
- ✓ Nach dem Anlegen eines Folgesponsors kann der alte, sowie auch der neue Sponsor weiterhin gesucht werden. Für neue Projekte kann jedoch nur der neue Folgesponsor verwendet werden.
- ✓ Beim Suchen oder Filtern nach dem Folgesponsor werden nur Treffer gefunden die explizit diesen neuen Folgesponsor verwenden.
- ✓ Beim Suche oder Filtern nach dem Vorgängersponsor werden sowohl Treffer gefunden die diesen Vorgängersponsor haben oder den Folgesponsor verwenden.

Benutzerverwaltung

- ✓ Im Reiter “Neue Konten Aktivieren” kann ein neues Benutzerkonto geprüft, verändert und aktiviert werden.
- ✓ Im Reiter “Änderungsanfragen prüfen” werden neue Änderungsanfragen angezeigt.
- ✓ Wird eine Änderungsanfrage angenommen werden die vom Benutzer gemachten Änderungen in sein Profil übernommen.
- ✓ Wird eine Änderungsanfrage abgelehnt, so bleibt das Profil des Benutzers unverändert.
- ✓ Administratoren können selbst Änderungen an der Änderungsanfrage vornehmen, sofern diese angenommen wird.
- ✓ Im Reiter “Benutzer verwalten” können alle aktiven Benutzerkonten bearbeitet werden.
- ✓ Wird ein Benutzerkonto beim Bearbeiten gesperrt, so wird dieses neu im Reiter “Gesperrte Benutzerkonten” angezeigt.
- ✓ Das Sperren eines Benutzerkontos ist nur unter Angabe einer Begründung möglich.
- ✓ Das Sperren eines Benutzerkontos entzieht dem gesperrten Benutzer seine jeweiligen Berechtigungen.

- ✓ Hat ein gesperrter Benutzer an einem Projekt mitwirkt, wird er in diesem Projekt weiterhin erwähnt, es sei denn er wird explizit aus diesem Projekt entfernt.
- ✓ Im Reiter "Gesperrte Benutzerkonten" können gesperrte Benutzerkonten reaktiviert werden.
- ✓ Beim reaktivieren eines Kontos kann das Benutzerprofil vom Administrator bearbeitet werden. (z.B. um die passende Rolle oder Organisationseinheit zu vergeben)
- ✓ Wird einem Benutzer durch eine Änderung eine neue Rolle zugewiesen, so wird diese beim Benutzer nach einem Neuladen der Seite wirksam.

Benutzerprofil

- ✓ Auf der Profilseite kann unter "Profil bearbeiten" eine Änderungsanfrage eingereicht werden.
- ✓ Wird eine Änderungsanfrage eingereicht, so wird der Benutzer darauf hingewiesen, dass diese erst noch von einem Administrator bestätigt werden muss.
- ✓ Solange eine Änderungsanfrage noch nicht angenommen wurde, weist eine Meldung darauf hin, dass sich diese Anfrage noch in Bearbeitung befindet.
- ✓ Die Änderungsanfrage kann bearbeitet werden.
- ✓ Die Änderungen werden erst im Benutzerkonto gespeichert sobald, die Änderungen von einem Administrator bestätigt wurden.

Login

- ✓ Navigationselement um sich einzuloggen wird dargestellt und funktioniert.
- ✓ Button "Einloggen mit Gitlab" wird korrekt dargestellt und funktioniert.

Registrieren

- ✓ Information "Wer darf ein Konto erstellen" wird korrekt dargestellt.
- ✓ Durch einen Klick auf "Einloggen mit Gitlab" kann sich ein Benutzer mittels Gitlab einloggen.
- ✓ Nach dem initialen Login und nach jedem Login bis das Benutzerkonto von einem Administrator aktiviert wurde, wird der Benutzer auf die Profilseite weitergeleitet.

9.1.4 Load Tests

In diesem Abschnitt sind die Screenshots der Konfigurationen, Tests und Resultate in Apache JMeter aufgelistet.

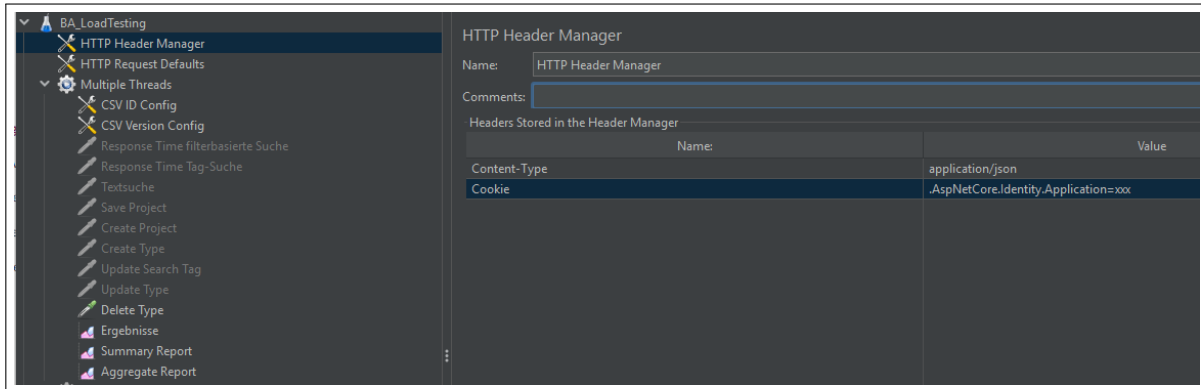


Abbildung 9.1: Identity Cookie und Content Headers für alle Requests

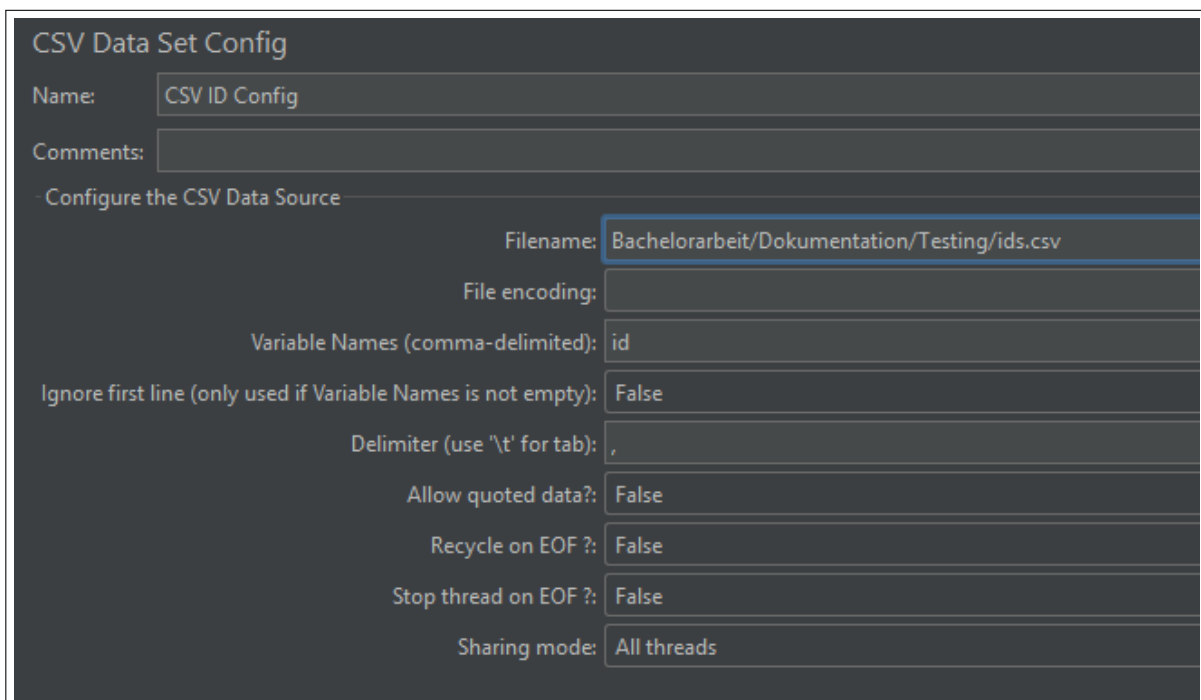


Abbildung 9.2: CSV Data Set Config mit IDs für die Delete-Tests

The screenshot shows an HTTP Request client interface. At the top, the title is "HTTP Request". Below it, there are fields for "Name:" (containing "Create Type") and "Comments:". A tabbed interface below has "Basic" and "Advanced" tabs, with "Basic" selected. Under the "Basic" tab, there is a "Web Server" section with fields for "Protocol [http:]", "Server Name or IP:", and "Port Number:". Below this is the "HTTP Request" section, where the method is set to "POST" and the "Path:" is "/api/admin/core-data/project-types". There are several checkboxes: "Redirect Automatically" (unchecked), "Follow Redirects" (checked), "Use KeepAlive" (checked), "Use multipart/form-data" (unchecked), and "Browser-compatible headers" (unchecked). At the bottom, there are tabs for "Parameters", "Body Data", and "Files Upload", with "Body Data" selected. The "Body Data" tab shows a single entry with index "1" and a JSON body: `{ "isArchived": false, "name": "multimedia" }`.

Abbildung 9.3: Request für die Erstellung eines ProjectTypes

The screenshot shows an HTTP Request client interface. At the top, the title is "HTTP Request". Below it, there are fields for "Name:" (containing "Delete Type") and "Comments:". A tabbed interface below has "Basic" and "Advanced" tabs, with "Basic" selected. Under the "Basic" tab, there is a "Web Server" section with fields for "Protocol [http:]", "Server Name or IP:", and "Port Number:". Below this is the "HTTP Request" section, where the method is set to "DELETE" and the "Path:" is "/api/admin/core-data/project-types/\${id}". There is a "Content encoding:" field. There are several checkboxes: "Redirect Automatically" (unchecked), "Follow Redirects" (checked), "Use KeepAlive" (checked), "Use multipart/form-data" (unchecked), and "Browser-compatible headers" (unchecked).

Abbildung 9.4: Request für das Löschen eines ProjectTypes mit dynamisch eingefügter ID in der URL

The screenshot shows an HTTP Request tool interface. The 'Name' field is 'Update Type'. The 'Comments' field is empty. The 'Basic' tab is selected, showing 'Web Server' settings (Protocol [http], Server Name or IP) and 'HTTP Request' settings (Method: PUT, Path: /api/admin/core-data/project-types). Checkboxes for 'Follow Redirects' and 'Use KeepAlive' are checked. The 'Body Data' tab is selected, showing a single parameter with the following JSON body:

```
1 [{"id":9,"name":"Multimedia","isArchived":true,"version":${version}]}
```

Abbildung 9.5: Request für die Bearbeitung eines ProjectTypes mit ID und dynamisch eingefügter Version (aus CSV) im Body.

The screenshot shows an HTTP Request tool interface. The 'Name' field is 'Update Search Tag'. The 'Comments' field is empty. The 'Basic' tab is selected, showing 'Web Server' settings (Protocol [http], Server Name or IP) and 'HTTP Request' settings (Method: PUT, Path: /api/admin/core-data/search-tags). Checkboxes for 'Follow Redirects' and 'Use KeepAlive' are checked. The 'Body Data' tab is selected, showing a single parameter with the following JSON body:

```
1 [{"id":10,"description":"Artificial Intelligence","parentId":1,"isArchived":false,"version":${version}]}
```

Abbildung 9.6: Request für die Bearbeitung eines verwendeten SearchTags mit ID, ParentID und dynamisch eingefügter Version (aus CSV) im Body.

Summary Report

Name:

Summary Report

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only:

☐ Errors

☐ Successes

Configure

Label	# Samples	Average ↑	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
Load Searchtags	50	23	21	52	4.35	100.00%	25.3/sec	4.15	5.69	168.0
Create Type	50	36	29	90	8.85	0.00%	24.9/sec	28.64	27.98	1176.0
Update Type	29	38	28	62	8.57	3.45%	23.3/sec	26.84	26.57	1181.0
Delete Type	100	40	21	105	10.36	0.00%	4.7/min	0.09	0.08	1171.0
Update Search Tag	100	40	30	91	8.96	48.00%	45.2/min	0.96	0.88	1310.7
Textsuche	80	59	27	96	11.97	0.00%	40.0/min	0.82	0.76	1265.0
Response Time filterbasierte Suche	86	84	45	136	17.19	0.00%	43.0/min	4.29	1.47	6119.0
Response Time Tag-Suche	84	92	62	131	14.98	0.00%	42.0/min	8.13	1.04	11880.0
TOTAL	579	55	21	136	26.01	17.10%	19.6/min	1.09	0.41	3411.0

Aggregate Report

Name:

Comments:

Write results to file / Read from file

Filename:

Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received K...	Sent KB/sec
Response Time filterbasierte Suche	86	84	81	106	114	127	45	136	0.00%	43.0/min	4.29	1.47
Response Time Tag-Suche	84	92	94	111	114	129	62	131	0.00%	42.0/min	8.13	1.04
Textsuche	80	59	57	75	80	86	27	96	0.00%	40.0/min	0.82	0.76
Update Type	29	38	35	48	58	62	28	62	3.45%	23.3/sec	26.84	26.57
Delete Type	100	40	38	53	55	66	21	105	0.00%	4.7/min	0.09	0.08
Create Type	50	36	35	40	42	90	29	90	0.00%	24.9/sec	28.64	27.98
Update Search Tag	100	40	39	48	51	74	30	91	48.00%	45.2/min	0.96	0.88
Load Searchtags	50	23	23	25	25	52	21	52	100.00%	25.3/sec	4.15	5.69
TOTAL	579	55	45	96	103	117	21	136	17.10%	19.6/min	1.09	0.41

9.1.5 Durchführung der Usability Tests

Usability Testing - Administratives Personal

Anmerkungen [↗](#)

- Gross-/Kleinschreibung beachten
- Benutzerverwaltung sehr intuitiv
- Besser evtl.: "Beschreibung aus angehängtem Dokument generieren", Link weiter oben, Tooltip auf Link, Generieren-Button beschriften
- Hierarchische Filter sehr praktisch
- Suche nach Filteroptionen auch
- Oberfläche grosszügig und übersichtlich
- Schön gestaltet und intuitiv

Vorbedingungen Authentifizierung [↗](#)

Es gilt als Vorbedingung, dass der Benutzer im Besitz eines Gitlab Accounts ist oder die nötigen Berechtigungen hat um einen Gitlab Account für das Gitlab der OST zu erstellen.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Als neuer Benutzer des Portals möchten Sie erst einmal einen Benutzer-Account anlegen. Erstellen Sie ein Benutzerkonto.	1. Der Benutzer findet das Navigationselement im Header, welches zur "Login"-Seite führt. 2. Der Benutzer klick t auf die Schaltfläche "Einloggen mit Gitlab" 3. Der Benutzer meldet sich mit seinem Gitlab-Konto an, sofern er bereits ein Konto hat. Sollte er noch kein Konto haben, versteht er anhand des Informationstextes auf der Anmeldeseite von Gitlab ob er befugt ist, ein Konto zu erstellen und wie er dafür vorgehen soll.	• Geht zu Login • Klickt auf "Wer darf ein Konto erstellen" • "OST SSO" nicht direkt klar (Beschreibung auf GitLab hilft) • Fazit: Nach Lesen der Beschreibung Login erfolgreich abgeschlossen
2	Da Sie im Institut für Software als Mitarbeiter angestellt sind, möchten Sie dies auch in Ihrem Profil konfigurieren. Passen Sie das Benutzerprofil entsprechend an.	1. Der Benutzer bemerkt das Navigationselement "Profil" 2. Der Benutzer verwendet die Felder "Organisationseinheit" und "Rolle" um sich als Mitarbeiter beim Institut für Software einzutragen.	• Wählt Rolle • Wählt zufällige Organisationseinheit • Gibt dann Titel ein • Findet es schliesslich doch unter Organisationseinheit • Schliesst dann Änderungsanfrage ab

			Anmerkung: Aktuell noch Aus- und Einloggen für Aktivieren der Berechtigung erforderlich
--	--	--	---

Vorbedingungen Stammdatenverwaltung [🔗](#)

Für alle Aufgaben gilt als Vorbedingung, dass der Benutzer eingeloggt ist und die für den Test erforderlichen Berechtigungen hat.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Als Administrator der Anwendung möchten Sie die Arbeitsergebnisse bearbeiten. Dazu möchten Sie im ersten Schritt herausfinden, welche Arbeitsergebnisse es aktuell gibt.	- Der Benutzer bemerkt das Navigationsfeld "Administration" und verwendet dieses, um zur Admin-Seite zu gelangen. - Der Benutzer findet anhand der Seitennavigation oder der Übersichtsseite die Tabelle, in welcher alle Arbeitsergebnisse abgebildet werden.	Findet alles sofort
2	Die Hochschule verlangt nach dem neuen Arbeitsergebnis-Typ "Bachelorarbeit". Erstellen Sie dieses neue Arbeitsergebnis.	- Der Benutzer sieht und verwendet die Schaltfläche "Neues Arbeitsergebnis anlegen" - Im Dialog "Neues Arbeitsergebnis anlegen" gibt er den Namen im entsprechenden Feld ein und lässt den Status auf "aktiv".	Erstellt es ohne Probleme
3	Leider kam es zu einem Missverständnis und der Arbeitsergebnis-Typ "Bachelorarbeit" wird doch nicht benötigt. Löschen Sie das vorhin erstellte Arbeitsergebnis wieder.	- Der Benutzer klickt auf das Mülleimer-Icon, um das entsprechende Element zu löschen. - Es öffnet sich ein Dialogfeld, welches vom Benutzer gelesen wird. Der Benutzer entschliesst sich anschliessend dazu, das Element zu löschen.	Löscht es ohne Probleme

4	Anscheinend kam es auch beim Arbeitsergebnis-Typ "Software" zu einem Missverständnis. Löschen Sie auch diesen Typ.	1. Der Benutzer löscht das Arbeitsergebnis vom Typ Software (gleiches Vorgehen wie im letzten Schritt) 2. Der Benutzer erhält eine Fehlermeldung, die ihn darauf hinweist, dass dieses Arbeitsergebnis verwendet wird und daher nicht gelöscht werden kann. 3. Fragt man den Benutzer, wie er weiter vorgehen würde, wäre das Archivieren der nächste vorgeschlagene Schritt.	• Gleiches vorgehen • Setzt Status dann auf Archiviert • Alles korrekt gemacht
5	Da das Arbeitsergebnis bereits verwendet wird, ist das Löschen leider nicht mehr möglich. Daher ist es erforderlich, das Arbeitsergebnis zu archivieren. Archivieren Sie das Arbeitsergebnis.	1. Der Benutzer klickt auf die "Bearbeiten"-Schaltfläche 2. Im Dialog "Arbeitsergebnis bearbeiten" ändert er den Status von "Aktiv" auf "Archiviert"	• Archiviert das Arbeitsergebnis
6	Nun möchten Sie kurz nachsehen, wie viele und welche Arbeitsergebnisse bereits archiviert wurden. Filtern Sie nach denjenigen Arbeitsergebnissen, die den Status "Archiviert" haben.	1. Der Benutzer findet das Filter-Dropdown mit dem nach einem Status gefiltert werden kann. 2. Der Benutzer wählt den Status "Archiviert" aus.	• Findet den Filter
7	Es interessiert Sie, ob es bereits ein Arbeitsergebnis mit dem Namen "Datensatz" gibt. Suchen Sie nach diesem Arbeitsergebnis.	1. Der Benutzer bemerkt das globale Suchfeld oder das Suchfeld nach Namen und verwendet dieses, um das gesuchte Arbeitsergebnis zu finden.	• Sucht erfolgreich
8	Ein verärgerter Mitarbeiter hat ihnen mitgeteilt, dass sein Lieblingskanton "Thurgau" nicht vorhanden ist. Erstellen Sie die neue Region "Thurgau".	1. Der Benutzer navigiert zur Verwaltungsansicht der Regionen. 2. Der Benutzer verwendet die Schaltfläche "Neue Region anlegen". Es öffnet sich ein Dialog "Neue Ortschaft anlegen".	• Findet das Formular • Füllt alles korrekt aus • Feedback: Sehr intuitiv

		3. Der Benutzer füllt den Dialog mit den entsprechenden Angaben ab. 4. Als übergeordnete Region wählt der Benutzer intuitiv die Region "Schweiz".	
9	Der Kanton Thurgau soll neu ein eigenständiges Land sein. Passen Sie die Stammdaten (Regionen) des Programms entsprechend an.	1. Der Benutzer sucht nach dem Kanton Thurgau mit der Suchleiste oder findet diesen, indem er die Kind-Elemente der Region "Schweiz" begutachtet. 2. Der Benutzer betätigt die Bearbeiten-Schaltfläche und ändert die übergeordnete Region auf "Keine übergeordnete Region" ab.	• Öffnet Formular • Wählt "Keine übergeordnete Region" aus
10	Vorbereitung: Es wird auf einem anderen Rechner (durch den Fragesteller) das Programm als Administrator ausgeführt. Während die Testperson den Datensatz in der Bearbeitungsansicht geöffnet hat, wird dieser insgeheim abgeändert. Öffnen Sie die Bearbeitungsansicht der Region "Schweiz" und warten Sie anschliessend auf weitere Anweisungen. (Schweiz insgeheim bearbeiten). In einer Volksabstimmung wurde beschlossen, dass die Schweiz neu in "Helvetische Republik" umbenannt werden sollte, da dieser Name von einer Volksmehrheit als seriöser und geopolitisch vorteilhafter empfunden wurde.	1. Der Benutzer erhält eine Concurrency-Fehlermeldung und bemerkt dadurch, dass eine zeitgleiche Bearbeitung stattgefunden hat. 2. Der Benutzer folgt der Anweisung der Fehlermeldung, lädt die Daten erneut und sieht die Änderungen, die der Kollege (Fragesteller) gemacht hat.	• Fehler wird erzeugt • Erst etwas Verwirrung vorhanden • Versucht erneut zu Bearbeiten • Lädt dann Seite neu •

Vorbedingungen veröffentlichtes Projekt verwalten & Projekt archivieren [↗](#)

Für alle Aufgaben gilt als Vorbedingung, dass der Benutzer eingeloggt ist und die für den Test erforderlichen Berechtigungen hat. Zudem existiert bereits ein veröffentlichtes Projekt, welches der Benutzer verwalten und archivieren kann.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Vor einiger Zeit haben Sie ein Forschungsprojekt zum Thema "KI" mit dem Titel "KI in der Softwareentwicklung" veröffentlicht. Das Abstract weist jedoch massive inhaltliche Fehler auf. Erstellen Sie eine neue Version des Projekts, um darin den Fehler zu korrigieren.	1. Der Benutzer sucht und findet das Projekt in der Sparte "Meine Projekte" 2. Der Benutzer klickt auf den Button "Neufassung erstellen" 3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird. 4. Der Benutzer nimmt in der neuen Version des Projekts die entsprechenden Änderungen vor.	• Erstellt Neufassung erfolgreich
2	Veröffentlichen Sie diese neue Version organisationsintern.	1. Der Benutzer wählt die Projektsichtbarkeit "Organisationseinheit", falls diese noch nicht gewählt ist. 2. Der Benutzer betätigt den Button "Organisationseinheitsintern veröffentlichen" und bestätigt den Dialog, der sich anschliessend öffnet. 3. Der Benutzer versteht, dass das vorherige Projekt nun überarbeitet wurde.	• Wählt alles korrekt aus
3	Ihr Chef teilt Ihnen mit, dass Ihr Forschungsprojekt ein grober Fehler aufweist und in diesem Zustand nicht präsentiert werden darf. Archivieren Sie das Projekt.	1. Der Benutzer sucht und findet das Projekt in der Sparte "Meine Projekte" 2. Der Benutzer klickt auf den Button "Archivieren" 3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird.	• Archiviert das Projekt erfolgreich
4	Es hat sich herausgestellt, dass der vermeintlich grobe Fehler nicht in Ihrem	1. Der Benutzer sucht und findet das Projekt in der Sparte "Meine Projekte"	• Schliesst alles erfolgreich ab

	Forschungsprojekt, sondern im Projekt eines Kollegen vorlag. Ihr Chef hat sich bei Ihnen entschuldigt und meinte, man solle das Projekt umgehend wieder veröffentlichen. Sorgen Sie dafür, dass das Projekt wieder veröffentlicht wird.	2. Der Benutzer klickt auf den Button "Projekt reaktivieren" 3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird. 4. Der Benutzer startet den Kontroll- / Freigabeprozess.	
--	--	--	--

Suche (Suchoptimierungen + Suche in PDF + Filter-Usability-Optimierungen) [🔗](#)

Als Vorbedingung gilt, dass bereits Projekte mit den gesuchten Ergebnissen existieren.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Sie möchten mehr über die Projekte an der OST erfahren. Suchen Sie nach Projekten mit dem Arbeitsergebnis „Software“.	Der Benutzer sucht nach „Software“ oder wählt im Filter „Arbeitsergebnis: Software“ aus.	• Sucht per "Suche" • Setzt ausserdem Filter
2	Suchen Sie nun nach den Softwareprojekten mit dem Sponsor „Eureka“.	Der Benutzer setzt zusätzlich den Filter „Sponsor: Eureka“ oder fügt den Suchbegriff „Eureka“ hinzu.	• Sucht erst in "Schlagwörter" • Findet dann Sponsoren • Findet gleiche Suchergebnisse per Textsuche
3	Sie haben von Ihrer Kollegin Julia Weber erfahren, dass Sie erst kürzlich Projekte veröffentlicht hat. Versuchen Sie, diese zu finden.	Der Benutzer sucht nach „Julia Weber“.	• Sucht nach Julia Weber
4	Setzen Sie Ihre Suche zurück und suchen Sie nach Projekten des Instituts für Architektur und des Institut für Software, die „künstliche Intelligenz“ erwähnen.	• Der Benutzer klickt auf „Suche zurücksetzen“ unterhalb der Filterleiste • Er filtert dann nach den Departementen „Informatik“ und „Gesundheit“ • Benutzer sucht nach „künstliche Intelligenz“ oder verwendet das Schlagwort „Künstliche Intelligenz“.	• verwendet Textsuche • Sehr gutes Feedback • Besonders Schnelligkeit positiv • Optimierungen aus BA kommen sehr gut an

Verwenden Sie nun die Suche eigenständig und probieren Sie verschiedene Filter sowie Suchabfragen aus.

Benutzerverwaltung

Als Vorbedingung gilt, dass der Benutzer eingeloggt ist und die Rolle „**Administrator**“ besitzt. Ausserdem liegen eine neue Benutzeranfrage sowie eine Änderungsanfrage eines bestehenden Benutzers vor.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Navigieren Sie zur Adminseite und ermitteln Sie, welche Aufgaben aktuell anstehen.	Der Benutzer gelangt zur Admin-Seite und sieht anhand der Badges, dass sowohl eine Kontoaktivierung als auch eine Änderungsanfrage bearbeitet werden sollten.	• Erreicht die Seite • Seite nicht ganz korrekt geladen
2	Der Benutzer „Benutzer Neu“ möchte ein Konto anlegen. Prüfen Sie seine Anfrage, weisen Sie ihm das Institut Architektur zu und vergeben Sie die Rolle „Mitarbeiter“.	• Der Benutzer öffnet die Seite „Neue Konten aktivieren“ • Er findet den Benutzer “Benutzer Neu” und klickt auf “Prüfen” • Er legt als Organisationseinheit “Institut Architektur” und als Rolle “Mitarbeiter” fest und aktiviert das Konto	• Erfolgreich abgeschlossen
3	Der Benutzer „Livio Mauchle“ möchte Änderungen an seinem Konto vornehmen. Finden Sie die Anfrage und nehmen Sie sie an.	• Der Benutzer öffnet die Seite „Änderungsanfragen prüfen“ • Er findet den Benutzer “Livio Mauchle” und klickt auf “Prüfen” • Er nimmt die Änderungsanfrage an	• Sofort gefunden • Gross-/Kleinschreibung beachten
4	Der Benutzer „Thomas Bauer“ hat bislang keine Berechtigungen. Vergeben Sie ihm die Rolle „Mitarbeiter“.	• Der Benutzer öffnet die Seite „Benutzer verwalten“ • Er findet den Benutzer “Thomas Bauer” und klickt auf “Bearbeiten” • Er weist ihm die Rolle Mitarbeiter zu und speichert die Änderungen	• Sofort gefunden

5	Die ehemalige Mitarbeiterin „Julia Weber“ hat die OST verlassen und soll keine Änderungen mehr vornehmen können. Sperren Sie ihr Benutzerkonto.	Der Benutzer öffnet die Seite „Benutzer verwalten“, findet „Julia Weber“, klickt auf „Bearbeiten“, wählt „Benutzerkonto sperren“ und gibt eine sinnvolle Begründung ein.	- Erst versucht ohne Begründung abzuschliessen - Dann erfolgreich
6	Sechs Monate nach ihrem Austritt kehrt „Julia Weber“ als normale Mitarbeiterin ans Institut für Software zurück. Reaktivieren Sie ihr Konto.	Der Benutzer öffnet die Seite „Gesperrte Benutzer“, findet „Julia Weber“, klickt auf „Bearbeiten“, weist erneut die Organisationseinheit Institut für Software und die Rolle Mitarbeiter zu und klickt auf „Benutzerkonto reaktivieren“.	Sofort gefunden

KI-Abstract Generator + Editor Erweiterung [🔗](#)

Als Vorbedingung gilt, dass der Benutzer eingeloggt ist und die Rolle „**Mitarbeiter**“ besitzt.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Legen Sie ein neues Projekt an. Verschaffen Sie sich zunächst einen kurzen Überblick und erfassen Sie dann eine Projektbeschreibung. Äussern Sie dabei laut Ihre Gedanken.	Der Benutzer legt über „Projektverwaltung → Neues Projekt anlegen“ ein neues Projekt an und bemerkt den Hinweis „Aus angehängtem PDF-Dokument generieren“.	- Erstellt Projekt - Link könnte besser platziert werden - Unklar, was bei Klick darauf passiert
2	Versuchen Sie nun, eine Beschreibung anhand eines PDF-Dokuments automatisch zu generieren.	Der Benutzer klickt auf den Link „Aus angehängtem PDF-Dokument generieren“, lädt ein PDF hoch und verwendet die Schaltfläche zum Erzeugen der Beschreibung aus dem Dokument.	- Klickt auf Datei wählen - Lädt dann Datei hoch “Generieren“-Button nach etwas Suchen gefunden - Beschreibung hinzufügen - Zeichenlimit bei Beschreibung überschritten → erst unklar, ob diese bearbeitet werden darf
3	Die Beschreibung wurde erstellt. Sehen Sie, wie das fertige Projekt später aussehen wird?	Der Benutzer findet die Schaltfläche „Zur Vorschau“ und betrachtet das Projekt in der Vorschauansicht.	- Findet Vorschau - Versucht, mit “Zurück“-Button zurückzukehren → “Zurück zu den Projekten”

			• Findet "Zur Bearbeitungsansicht erst nach etwas suchen"
4	Die Beschreibung wirkt noch unstrukturiert. Unterteilen Sie sie in die Abschnitte Problemstellung, Lösung und Fazit und formatieren Sie die Überschriften entsprechend.	Der Benutzer wendet die Formatierungsoptionen für Überschriften korrekt an und gliedert den Text in die drei geforderten Abschnitte.	• Findet Formatierungsoptionen • Überschriften sind klar
5	Fügen Sie im Abschnitt Lösung eine Aufzählung aller durchgeführten Schritte hinzu und heben Sie wichtige Wörter (fett, kursiv oder unterstrichen) hervor.	Der Benutzer nutzt die Listenfunktion für die Schrittauflistung und versteht wie er Hervorhebungen (Fettschrift, Kursivschrift, Unterstreichung) anwenden kann.	• Verwendet Listenfunktion • Findet Hervorhebungen etc.

Dashboard-Erweiterung

Für die unten stehenden Aufgaben gilt als Vorbedingung, dass bereits Projekte existieren, damit sinnvolle Statistiken im Dashboard angezeigt werden.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Finden Sie heraus, in welchen Regionen Projekte mit dem Sponsor „Eureka“ durchgeführt wurden und welches Arbeitsergebnis bei diesem Sponsor am häufigsten erzielt wurde.	Der Benutzer setzt den Filter Sponsor = Eureka und sieht die Statistiken Projekte nach Region sowie Projekte nach Arbeitsergebnis. Anhand diese Statistiken lassen sich die gesuchten Antworten finden.	• Setzt Filter • Findet entsprechende Grafik •
2	Speichern Sie die aktuellen Filtereinstellungen, da diese für zukünftige Abfragen erneut benötigt werden.	Der Benutzer klickt auf „Lesezeichen hinzufügen“, vergibt einen aussagekräftigen Namen und sichert die aktuellen Filtereinstellungen.	• Klickt auf Lesezeichen hinzufügen • Speichert es ab
3	Sie möchten nun herausfinden welche Organisationseinheiten bereits Projekte im Raum St. Gallen durchgeführt haben. Setzen sie erst alle Filter zurück und führen Sie anschliessend die Abfrage durch.	Der Benutzer klickt auf „Filter zurücksetzen“ und stellt anschliessend im Filter Region den Wert St. Gallen ein.	• Setzt Filter korrekt

4	Rufen Sie nun Ihre zuvor gespeicherte Filterkombination wieder auf.	Der Benutzer öffnet die Lesezeichen-Anzeige und wählt das entsprechende Lesezeichen aus.	• Findet das alte Lesezeichen sofort wieder
5	Sie sind mit dem aktuellen Namen des Lesezeichens nicht ganz zufrieden. Ändern Sie den Namen des gespeicherten Lesezeichens.	Der Benutzer klickt auf das Stiftsymbol in der Lesezeichenliste, trägt im Dialog einen neuen Namen ein und bestätigt diesen.	• Bearbeitet das Lesezeichen erfolgreich
6	Das Lesezeichen wird nicht weiter benötigt. Entfernen Sie das Lesezeichen aus der Lesezeichenleiste.	Der Benutzer klickt auf das Mülleimer-Symbol (oder den entsprechenden Button), bestätigt im Dialog das Löschen und entfernt das Lesezeichen.	• Löscht das Lesezeichen erfolgreich

Verwenden Sie das Dashboard nun eigenständig und probieren Sie verschiedene Filterkombinationen aus.

Usability Testing - Wissenschaftlicher Mitarbeiter 1

Vorbedingungen Authentifizierung [🔗](#)

Es gilt als Vorbedingung, dass der Benutzer im Besitz eines Gitlab Accounts ist oder die nötigen Berechtigungen hat um einen Gitlab Account für das Gitlab der OST zu erstellen.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Als neuer Benutzer des Portals möchten Sie erst einmal einen Benutzer-Account anlegen. Erstellen Sie ein Benutzerkonto.	1. Der Benutzer findet das Navigationselement im Header, welches zur "Login"-Seite führt. 2. Der Benutzer klickt auf die Schaltfläche "Einloggen mit Gitlab" 3. Der Benutzer meldet sich mit seinem Gitlab-Konto an, sofern er bereits ein Konto hat. Sollte er noch kein Konto haben, versteht er anhand des Informationstextes auf der Anmeldeseite von Gitlab ob er befugt ist, ein Konto zu erstellen und wie er dafür vorgehen soll.	• Sucht erst, ob es andere Möglichkeit zum Registrieren gibt
2	Da Sie im Institut für Software als Mitarbeiter angestellt sind, möchten Sie dies auch in Ihrem Profil konfigurieren. Passen Sie das Benutzerprofil entsprechend an.	1. Der Benutzer bemerkt das Navigationselement "Profil" 2. Der Benutzer verwendet die Felder "Organisationseinheit" und "Rolle" um sich als Mitarbeiter beim Institut für Software einzutragen.	• Findet Institut für Software • Anmerkung: INScheinbar noch nicht vorhanden

Suche (Suchoptimierungen + Suche in PDF + Filter-Usability-Optimierungen) [🔗](#)

Als Vorbedingung gilt, dass bereits Projekte mit den gesuchten Ergebnissen existieren.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Sie möchten mehr über die Projekte an der OST erfahren. Suchen Sie nach	Der Benutzer sucht nach „Software“ oder wählt im Filter „Arbeitsergebnis: Software“ aus.	• Sucht mit Suchleiste • Findet Software unter Arbeitsergebnis

	Projekten mit dem Arbeitsergebnis „Software“.		
2	Suchen Sie nun nach den Softwareprojekten mit dem Sponsor „Eureka“.	Der Benutzer setzt zusätzlich den Filter „Sponsor: Eureka“ oder fügt den Suchbegriff „Eureka“ hinzu.	• Findet Sponsor-Filter sofort
3	Sie haben von Ihrer Kollegin Julia Weber erfahren, dass Sie erst kürzlich Projekte veröffentlicht hat. Versuchen Sie, diese zu finden.	Der Benutzer sucht nach „Julia Weber“.	• Sucht in Suchleiste • Findet Projekte • Versucht dann auf den Namen zu klicken (Idee: „Profilseite“ mit allen Projekten einer Person) • Durchsucht auch Filter
4	Setzen Sie Ihre Suche zurück und suchen Sie nach Projekten des Instituts für Architektur und des Institut für Software, die „künstliche Intelligenz“ erwähnen.	• Der Benutzer klickt auf „Suche zurücksetzen“ unterhalb der Filterleiste • Er filtert dann nach den Organisationseinheiten „Informatik“ und „künstliche Intelligenz“ • Benutzer sucht nach „künstliche Intelligenz“ oder verwendet das Schlagwort „Künstliche Intelligenz“.	• Setzt Filter korrekt • Findet Ergebnisse

Verwenden Sie nun die Suche eigenständig und probieren Sie verschiedene Filter sowie Suchabfragen aus.

Anmerkungen:

- Sehr flüssig
- Besonders Suche nach Optionen sticht positiv heraus
- Sortierfunktion nicht vorhanden
- Bei Datum etwas Verwirrung: Veröffentlichungs- vs. Durchführungsdatum

Benutzerverwaltung [🔗](#)

Als Vorbedingung gilt, dass der Benutzer eingeloggt ist und die Rolle „**Administrator**“ besitzt. Ausserdem liegen eine neue Benutzeranfrage sowie eine Änderungsanfrage eines bestehenden Benutzers vor.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Navigieren Sie zur Adminseite und ermitteln Sie, welche Aufgaben aktuell anstehen.	Der Benutzer gelangt zur Admin-Seite und sieht anhand der Badges, dass sowohl eine Kontoaktivierung als auch eine Änderungsanfrage bearbeitet werden sollten.	• Findet Admin-Navigation nach etwas suchen • Evtl. in „Admin“ umbenennen

2	Der Benutzer „Benutzer Neu“ möchte ein Konto anlegen. Prüfen Sie seine Anfrage, weisen Sie ihm das Institut Architektur zu und vergeben Sie die Rolle „Mitarbeiter“.	• Der Benutzer öffnet die Seite „Neue Konten aktivieren“ • Er findet den Benutzer „Benutzer Neu“ und klickt auf „Prüfen“ • Er legt als Organisationseinheit „Institut Architektur“ und als Rolle „Mitarbeiter“ fest und aktiviert das Konto	• Findet Tab für neue Konten • Aktuell kein neues Konto vorhanden • Offene Aufgaben gut hervorgehoben
3	Der Benutzer „Livio Mauchle“ möchte Änderungen an seinem Konto vornehmen. Finden Sie die Anfrage und nehmen Sie sie an.	• Der Benutzer öffnet die Seite „Änderungsanfragen prüfen“ • Er findet den Benutzer „Livio Mauchle“ und klickt auf „Prüfen“ • Er nimmt die Änderungsanfrage an	• Findet Änderungsanfrage dank Aufgaben-Anzeige • Hervorhebung geänderter Felder?
4	Der Benutzer „Thomas Bauer“ hat bislang keine Berechtigungen. Vergeben Sie ihm die Rolle „Mitarbeiter“.	• Der Benutzer öffnet die Seite „Benutzer verwalten“ • Er findet den Benutzer „Thomas Bauer“ und klickt auf „Bearbeiten“ • Er weist ihm die Rolle Mitarbeiter zu und speichert die Änderungen	• Geht auf Benutzerverwaltung • Sieht den Benutzer • Macht die geforderte Änderung
5	Die ehemalige Mitarbeiterin „Julia Weber“ hat die OST verlassen und soll keine Änderungen mehr vornehmen können. Sperren Sie ihr Benutzerkonto.	• Der Benutzer öffnet die Seite „Benutzer verwalten“, findet „Julia Weber“, klickt auf „Bearbeiten“, wählt „Benutzerkonto sperren“ und gibt eine sinnvolle Begründung ein.	• Geht auf Benutzerverwaltung • Standard „5“ ist etwas wenig bei Anzahl angezeigter Mitarbeiter • Sperrt diese • Gibt Begründung ein • Toast etwas schnell weg
6	Sechs Monate nach ihrem Austritt kehrt „Julia Weber“ als normale Mitarbeiterin ans Institut für Software zurück. Reaktivieren Sie ihr Konto.	• Der Benutzer öffnet die Seite „Gesperrte Benutzer“, findet „Julia Weber“, klickt auf „Bearbeiten“, weist erneut die Organisationseinheit Institut für Software und die Rolle Mitarbeiter zu und klickt auf „Benutzerkonto reaktivieren“.	• Sucht erst unter „Benutzer verwalten“ • Findet dann schnell „Gesperrte Benutzerkonten“ • Reaktiviert sie erfolgreich • Toast verschwindet, wenn Benutzer Maus darauf bewegt

Anmerkungen

- Man kann nicht viel falsch machen
- Toasts etwas kurz/verschwinden
- Zu wenige angezeigte Benutzer standardmässig
- Titel auf Admin-Startseite

KI-Generator + Editor Erweiterung [🔗](#)

Als Vorbedingung gilt, dass der Benutzer eingeloggt ist und die Rolle „Mitarbeiter“ besitzt.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Legen Sie ein neues Projekt an. Verschaffen Sie sich zunächst einen kurzen Überblick und erfassen Sie dann eine Projektbeschreibung. Äussern Sie dabei laut Ihre Gedanken.	Der Benutzer legt über „Projektverwaltung → Neues Projekt anlegen“ ein neues Projekt an und bemerkt den Hinweis „Aus angehängtem PDF-Dokument generieren“.	• Findet “Neues Projekt anlegen” in Menü • Bemerkt Hinweis nicht direkt • Probiert Formatierungen aus
2	Versuchen Sie nun, eine Beschreibung anhand eines PDF-Dokuments automatisch zu generieren.	Der Benutzer klickt auf den Link „Aus angehängtem PDF-Dokument generieren“, lädt ein PDF hoch und verwendet die Schaltfläche zum Erzeugen der Beschreibung aus dem Dokument.	• Findet beim Runterscrollen den Link • Zieht Datei mit Drag und Drop in das Feld • Anmerkung: Feld könnte grösser sein • Klickt Datei hochladen • Sieht Button für Generieren nicht direkt • Beschreibung für Generieren auch bei Dateien platzieren • Findet Button nach etwas suchen • Beschreibung könnte länger sein • Link zu Dateien besser Beschreiben, z.B. Titel “Beschreibung aus Datei generieren” + Link “Zu den Dateien”
3	Die Beschreibung wurde erstellt. Sehen Sie, wie das fertige Projekt später aussehen wird?	Der Benutzer findet die Schaltfläche „Zur Vorschau“ und betrachtet das Projekt in der Vorschauansicht.	• Findet Vorschau (schon vor Aufgabe)
4	Die Beschreibung wirkt noch unstrukturiert. Unterteilen Sie sie in die Abschnitte Problemstellung, Lösung	Der Benutzer wendet die Formatierungsoptionen für Überschriften korrekt an und	• Benutzer macht alles korrekt mit Formatierungsoptionen

	und Fazit und formatieren Sie die Überschriften entsprechend.	gliedert den Text in die drei geforderten Abschnitte.	
5	Fügen Sie im Abschnitt Lösung eine Aufzählung aller durchgeführten Schritte hinzu und heben Sie wichtige Wörter (fett, kursiv oder unterstrichen) hervor.	Der Benutzer nutzt die Listenfunktion für die Schrittauflistung und versteht wie er Hervorhebungen (Fettschrift, Kursivschrift, Unterstreichung) anwenden kann.	• Benutzer verwendet Listenfunktion etc. korrekt

Dashboard-Erweiterung [↗](#)

Für die unten stehenden Aufgaben gilt als Vorbedingung, dass bereits Projekte existieren, damit sinnvolle Statistiken im Dashboard angezeigt werden.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Finden Sie heraus, in welchen Regionen Projekte mit dem Sponsor „Eureka“ durchgeführt wurden und welches Arbeitsergebnis bei diesem Sponsor am häufigsten erzielt wurde.	Der Benutzer setzt den Filter Sponsor = Eureka und sieht die Statistiken Projekte nach Region sowie Projekte nach Arbeitsergebnis. Anhand diese Statistiken lassen sich die gesuchten Antworten finden.	• Geht zuerst auf “Projekte prüfen” • Dann zum Dashboard • Setzt Filter korrekt • Findet Antworten heraus
2	Speichern Sie die aktuellen Filtereinstellungen, da diese für zukünftige Abfragen erneut benötigt werden.	Der Benutzer klickt auf „Lesezeichen hinzufügen“, vergibt einen aussagekräftigen Namen und sichert die aktuellen Filtereinstellungen.	• Sucht erst bei Filteroptionen • Findet dann “Lesezeichen hinzufügen” • Benennt es sinnvoll
3	Sie möchten nun herausfinden welche Organisationseinheiten bereits Projekte im Raum St. Gallen durchgeführt haben. Setzen sie erst alle Filter zurück und führen Sie anschliessend die Abfrage durch.	Der Benutzer klickt auf „Filter zurücksetzen“ und stellt anschliessend im Filter Region den Wert St. Gallen ein.	• Setzt Filter korrekt • Verwendet Browser-Suche um korrekte Statistik zu finden
4	Rufen Sie nun Ihre zuvor gespeicherte Filterkombination wieder auf.	Der Benutzer öffnet die Lesezeichen-Anzeige und wählt das entsprechende Lesezeichen aus.	• Findet Lesezeichen Anzeige • Wählt Lesezeichen korrekt aus
5	Sie sind mit dem aktuellen Namen des Lesezeichens nicht ganz zufrieden.	Der Benutzer klickt auf das Stiftsymbol in der Lesezeichenliste, trägt im	• Findet Bearbeitung sofort ohne Probleme • Gibt neuen Namen

	Ändern Sie den Namen des gespeicherten Lesezeichens.	Dialog einen neuen Namen ein und bestätigt diesen.	
6	Das Lesezeichen wird nicht weiter benötigt. Entfernen Sie das Lesezeichen aus der Lesezeichenleiste.	Der Benutzer klickt auf das Mülleimer-Symbol (oder den entsprechenden Button), bestätigt im Dialog das Löschen und entfernt das Lesezeichen.	• Löscht und entfernt das Lesezeichen • Keine Probleme

Verwenden Sie das Dashboard nun eigenständig und probieren Sie verschiedene Filterkombinationen aus.

Anmerkungen

- Word Cloud evtl. nicht schräg

Usability Testing - Wissenschaftlicher Mitarbeiter 2

Allgemein [🔗](#)

- Statistiken sind gut
- Styl von ost sehr cool
- Feedback eventuell Filter (erste Filter) von Anfang an aufklappen
- Stammdatenverwaltung eventuell die Suche in ganzer Tabelle entfernen

Vorbedingungen Authentifizierung [🔗](#)

Es gilt als Vorbedingung, dass der Benutzer im Besitz eines Gitlab Accounts ist oder die nötigen Berechtigungen hat um einen Gitlab Account für das Gitlab der OST zu erstellen.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Als neuer Benutzer des Portals möchten Sie erst einmal einen Benutzer-Account anlegen. Erstellen Sie ein Benutzerkonto.	1. Der Benutzer findet das Navigationselement im Header, welches zur "Login"-Seite führt. 2. Der Benutzer klickt auf die Schaltfläche "Einloggen mit Gitlab" 3. Der Benutzer meldet sich mit seinem Gitlab-Konto an, sofern er bereits ein Konto hat. Sollte er noch kein Konto haben, versteht er anhand des Informationstextes auf der Anmeldeseite von Gitlab ob er befugt ist, ein Konto zu erstellen und wie er dafür vorgehen soll.	• Benutzer findet Login • Klickt einloggen mit GitLab • Funktioniert alles
2	Da Sie im Institut für Software als Mitarbeiter angestellt sind, möchten Sie dies auch in Ihrem Profil konfigurieren. Passen Sie das Benutzerprofil entsprechend an.	1. Der Benutzer bemerkt das Navigationselement "Profil" 2. Der Benutzer verwendet die Felder "Organisationseinheit" und "Rolle" um sich als Mitarbeiter beim Institut für Software einzutragen.	• Eventuell mit Abkürzung → für Suche (IFS) • Rest funktioniert gut

Vorbedingungen Stammdatenverwaltung [🔗](#)

Für alle Aufgaben gilt als Vorbedingung, dass der Benutzer eingeloggt ist und die für den Test erforderlichen Berechtigungen hat.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Als Administrator der Anwendung möchten Sie die Arbeitsergebnisse bearbeiten. Dazu möchten Sie im ersten Schritt herausfinden, welche Arbeitsergebnisse es aktuell gibt.	1. Der Benutzer bemerkt das Navigationsfeld "Administration" und verwendet dieses, um zur Admin-Seite zu gelangen. 2. Der Benutzer findet anhand der Seitennavigation oder der Übersichtsseite die Tabelle, in welcher alle Arbeitsergebnisse abgebildet werden.	• Findet Arbeitsergebnisse sofort
2	Die Hochschule verlangt nach dem neuen Arbeitsergebnis-Typ "Bachelorarbeit". Erstellen Sie dieses neue Arbeitsergebnis.	1. Der Benutzer sieht und verwendet die Schaltfläche "Neues Arbeitsergebnis anlegen" 2. Im Dialog "Neues Arbeitsergebnis anlegen" gibt er den Namen im entsprechenden Feld ein und lässt den Status auf "aktiv".	• Findet "Neues Arbeitsergebnis Anlegen" • Ist klar
3	Leider kam es zu einem Missverständnis und der Arbeitsergebnis-Typ "Bachelorarbeit" wird doch nicht benötigt. Löschen Sie das vorhin erstellte Arbeitsergebnis wieder.	1. Der Benutzer klickt auf das Mülleimer-Icon, um das entsprechende Element zu löschen. 2. Es öffnet sich ein Dialogfeld, welches vom Benutzer gelesen wird. Der Benutzer entschliesst sich anschliessend dazu, das Element zu löschen.	• Funktioniert intuitiv
4	Anscheinend kam es auch beim Arbeitsergebnis-Typ "Software" zu einem Missverständnis. Löschen Sie auch diesen Typ.	1. Der Benutzer löscht das Arbeitsergebnis vom Typ Software (gleiches Vorgehen wie im letzten Schritt) 2. Der Benutzer erhält eine Fehlermeldung, die ihn darauf hinweist, dass dieses Arbeitsergebnis verwendet wird und daher nicht gelöscht werden kann. 3. Fragt man den Benutzer, wie er weiter vorgehen würde, wäre das	• Versucht zu löschen • Meldung Conflict erscheint • Eventuell in Fehlermeldung erwähnen dass man es Archivieren kann "please Archive it" • Fehlermeldung länger anzeigen oder sogar mittig näher beim Löschen anzeigen

		Archivieren der nächste vorgeschlagene Schritt.	
5	Da das Arbeitsergebnis bereits verwendet wird, ist das Löschen leider nicht mehr möglich. Daher ist es erforderlich, das Arbeitsergebnis zu archivieren. Archivieren Sie das Arbeitsergebnis.	1. Der Benutzer klickt auf die "Bearbeiten"-Schaltfläche 2. Im Dialog "Arbeitsergebnis bearbeiten" ändert er den Status von "Aktiv" auf "Archiviert"	• Diese Anweisung macht die vorherige Problematik klar • Archivieren funktioniert
6	Nun möchten Sie kurz nachsehen, wie viele und welche Arbeitsergebnisse bereits archiviert wurden. Filtern Sie nach denjenigen Arbeitsergebnissen, die den Status "Archiviert" haben.	1. Der Benutzer findet das Filter-Dropdown mit dem nach einem Status gefiltert werden kann. 2. Der Benutzer wählt den Status "Archiviert" aus.	• Verwendet Filter, funktioniert
7	Es interessiert Sie, ob es bereits ein Arbeitsergebnis mit dem Namen "Datensatz" gibt. Suchen Sie nach diesem Arbeitsergebnis.	1. Der Benutzer bemerkt das globale Suchfeld oder das Suchfeld nach Namen und verwendet dieses, um das gesuchte Arbeitsergebnis zu finden.	• Verwendet Suchfeld • Globales Suchfeld fällt nicht auf und ist nicht unbedingt nötig
8	Ein verärgerter Mitarbeiter hat ihnen mitgeteilt, dass sein Lieblingskanton "Thurgau" nicht vorhanden ist. Erstellen Sie die neue Region "Thurgau".	1. Der Benutzer navigiert zur Verwaltungsansicht der Regionen. 2. Der Benutzer verwendet die Schaltfläche "Neue Region anlegen". Es öffnet sich ein Dialog "Neue Ortschaft anlegen". 3. Der Benutzer füllt den Dialog mit den entsprechenden Angaben ab. 4. Als übergeordnete Region wählt der Benutzer intuitiv die Region "Schweiz".	• Funktioniert intuitiv
9	Der Kanton Thurgau soll neu ein eigenständiges Land sein. Passen Sie die Stammdaten (Regionen) des Programms entsprechend an.	1. Der Benutzer sucht nach dem Kanton Thurgau mit der Suchleiste oder findet diesen, indem er die Kind-Elemente der Region "Schweiz" begutachtet.	• Funktioniert ohne Probleme • Vorschlag: eventuell erstes Element "Keine Übergeordnete Region anders hervorheben"

		2. Der Benutzer betätigt die Bearbeiten-Schaltfläche und ändert die übergeordnete Region auf "Keine übergeordnete Region" ab.	
10	Vorbereitung: Es wird auf einem anderen Rechner (durch den Fragesteller) das Programm als Administrator ausgeführt. Während die Testperson den Datensatz in der Bearbeitungsansicht geöffnet hat, wird dieser insgeheim abgeändert. Öffnen Sie die Bearbeitungsansicht der Region "Schweiz" und warten Sie anschliessend auf weitere Anweisungen. (Schweiz insgeheim bearbeiten). In einer Volksabstimmung wurde beschlossen, dass die Schweiz neu in "Helvetische Republik" umbenannt werden sollte, da dieser Name von einer Volksmehrheit als seriöser und geopolitisch vorteilhafter empfunden wurde.	1. Der Benutzer erhält eine Concurrency-Fehlermeldung und bemerkt dadurch, dass eine zeitgleiche Bearbeitung stattgefunden hat. 2. Der Benutzer folgt der Anweisung der Fehlermeldung, lädt die Daten erneut und sieht die Änderungen, die der Kollege (Fragesteller) gemacht hat.	• Sieht Fehlermeldung • Lädt die Seite neu • Sieht die Änderungen, versteht den Fehler

Suche (Suchoptimierungen + Suche in PDF + Filter-Usability-Optimierungen) [↗](#)

Als Vorbedingung gilt, dass bereits Projekte mit den gesuchten Ergebnissen existieren.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Sie möchten mehr über die Projekte an der OST erfahren. Suchen Sie nach Projekten mit dem Arbeitsergebnis „Software“.	Der Benutzer sucht nach „Software“ oder wählt im Filter „Arbeitsergebnis: Software“ aus.	• Verwendet suche, findet Softwareprojekte

2	Suchen Sie nun nach den Softwareprojekten mit dem Sponsor „Eureka“.	Der Benutzer setzt zusätzlich den Filter „Sponsor: Eureka“ oder fügt den Suchbegriff „Eureka“ hinzu.	• Verwendet die Suche um die Projekte zu finden • Eventuell Keywords auch anzeigen die gefunden wurden
3	Sie haben von Ihrer Kollegin Julia Weber erfahren, dass Sie erst kürzlich Projekte veröffentlicht hat. Versuchen Sie, diese zu finden.	Der Benutzer sucht nach „Julia Weber“.	• Verwendet Suche Funktion, findet Projekte direkt
4	Setzen Sie Ihre Suche zurück und suchen Sie nach Projekten des Instituts für Architektur und des Institut für Software, die „künstliche Intelligenz“ erwähnen.	• Der Benutzer klickt auf „Suche zurücksetzen“ unterhalb der Filterleiste • Er filtert dann nach den Departementen „Informatik“ und „Gesundheit“ • Benutzer sucht nach „künstliche Intelligenz“ oder verwendet das Schlagwort „Künstliche Intelligenz“.	• Verwendet Suchfunktion für Organisationseinheiten und “Künstliche Intelligenz”

Verwenden Sie nun die Suche eigenständig und probieren Sie verschiedene Filter sowie Suchabfragen aus.

Dashboard-Erweiterung [🔗](#)

Für die unten stehenden Aufgaben gilt als Vorbedingung, dass bereits Projekte existieren, damit sinnvolle Statistiken im Dashboard angezeigt werden.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Finden Sie heraus, in welchen Regionen Projekte mit dem Sponsor „Eureka“ durchgeführt wurden und welches Arbeitsergebnis bei diesem Sponsor am häufigsten erzielt wurde.	Der Benutzer setzt den Filter Sponsor = Eureka und sieht die Statistiken Projekte nach Region sowie Projekte nach Arbeitsergebnis. Anhand diese Statistiken lassen sich die gesuchten Antworten finden.	• Erst Probleme die Statistiken zu finden • Findet Regionen und korrekte Filtereinstellungen sofort
2	Speichern Sie die aktuellen Filtereinstellungen, da diese für zukünftige Abfragen erneut benötigt werden.	Der Benutzer klickt auf „Lesezeichen hinzufügen“, vergibt einen aussagekräftigen Namen und sichert die aktuellen Filtereinstellungen.	• Findet Lesezeichen hinzufügen • Funktioniert einwandfrei

3	Sie möchten nun herausfinden welche Organisationseinheiten bereits Projekte im Raum St. Gallen durchgeführt haben. Setzen sie erst alle Filter zurück und führen Sie anschliessend die Abfrage durch.	Der Benutzer klickt auf „Filter zurücksetzen“ und stellt anschliessend im Filter Region den Wert St. Gallen ein.	• Findet korrekte Filtereinstellung • Findet die Statistik
4	Rufen Sie nun Ihre zuvor gespeicherte Filterkombination wieder auf.	Der Benutzer öffnet die Lesezeichen-Anzeige und wählt das entsprechende Lesezeichen aus.	• Navigiert zu Lesezeichen • Wählt Lesezeichen • Keine Probleme
5	Sie sind mit dem aktuellen Namen des Lesezeichens nicht ganz zufrieden. Ändern Sie den Namen des gespeicherten Lesezeichens.	Der Benutzer klickt auf das Stiftsymbol in der Lesezeichenliste, trägt im Dialog einen neuen Namen ein und bestätigt diesen.	• Klickt auf Lesezeichen ändern • Alles klar
6	Das Lesezeichen wird nicht weiter benötigt. Entfernen Sie das Lesezeichen aus der Lesezeichenleiste.	Der Benutzer klickt auf das Mülleimer-Symbol (oder den entsprechenden Button), bestätigt im Dialog das Löschen und entfernt das Lesezeichen.	• Alles klar

Verwenden Sie das Dashboard nun eigenständig und probieren Sie verschiedene Filterkombinationen aus.

Usability Testing - Student 1

Vorbedingungen Authentifizierung [🔗](#)

Es gilt als Vorbedingung, dass der Benutzer im Besitz eines Gitlab Accounts ist oder die nötigen Berechtigungen hat um einen Gitlab Account für das Gitlab der OST zu erstellen.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Als neuer Benutzer des Portals möchten Sie erst einmal einen Benutzer-Account anlegen. Erstellen Sie ein Benutzerkonto.	1. Der Benutzer findet das Navigationselement im Header, welches zur "Login"-Seite führt. 2. Der Benutzer klickt auf die Schaltfläche "Einloggen mit Gitlab" 3. Der Benutzer meldet sich mit seinem Gitlab-Konto an, sofern er bereits ein Konto hat. Sollte er noch kein Konto haben, versteht er anhand des Informationstextes auf der Anmeldeseite von Gitlab ob er befugt ist, ein Konto zu erstellen und wie er dafür vorgehen soll.	Benutzer fand sehr schnell heraus wo und wie er ein Benutzerkonto anlegen konnte, die Anmeldung war erfolgreich.
2	Da Sie im Institut für Software als Mitarbeiter angestellt sind, möchten Sie dies auch in Ihrem Profil konfigurieren. Passen Sie das Benutzerprofil entsprechend an.	1. Der Benutzer bemerkt das Navigationselement "Profil" 2. Der Benutzer verwendet die Felder "Organisationseinheit" und "Rolle" um sich als Mitarbeiter beim Institut für Software einzutragen.	Profil wurde angepasst, Titel evtl. missverständlich; Hinweis zu Rollenklärung empfohlen.

Vorbedingungen Admin Page [🔗](#)

Für alle Aufgaben gilt als Vorbedingung, dass der Benutzer eingeloggt ist und die für den Test erforderlichen Berechtigungen hat.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Als Administrator der Anwendung möchten Sie die Arbeitsergebnisse	1. Der Benutzer bemerkt das Navigationsfeld "Administration" und	Navigation und Anzeige funktionierten problemlos.

	bearbeiten. Dazu möchten Sie im ersten Schritt herausfinden, welche Arbeitsergebnisse es aktuell gibt.	verwendet dieses, um zur Admin-Seite zu gelangen. 2. Der Benutzer findet anhand der Seitennavigation oder der Übersichtsseite die Tabelle, in welcher alle Arbeitsergebnisse abgebildet werden.	
2	Die Hochschule verlangt nach dem neuen Arbeitsergebnis-Typ "Bachelorarbeit". Erstellen Sie dieses neue Arbeitsergebnis.	1. Der Benutzer sieht und verwendet die Schaltfläche "Neues Arbeitsergebnis anlegen" 2. Im Dialog "Neues Arbeitsergebnis anlegen" gibt er den Namen im entsprechenden Feld ein und lässt den Status auf "aktiv".	Neuer Typ erfolgreich angelegt.
3	Leider kam es zu einem Missverständnis und der Arbeitsergebnis-Typ "Bachelorarbeit" wird doch nicht benötigt. Löschen Sie das vorhin erstellte Arbeitsergebnis wieder.	1. Der Benutzer klickt auf das Mülleimer-Icon, um das entsprechende Element zu löschen. 2. Es öffnet sich ein Dialogfeld, welches vom Benutzer gelesen wird. Der Benutzer entschliesst sich anschliessend dazu, das Element zu löschen.	Löschen funktionierte wie erwartet.
4	Anscheinend kam es auch beim Arbeitsergebnis-Typ "Software" zu einem Missverständnis. Löschen Sie auch diesen Typ.	1. Der Benutzer löscht das Arbeitsergebnis vom Typ Software (gleiches Vorgehen wie im letzten Schritt) 2. Der Benutzer erhält eine Fehlermeldung, die ihn darauf hinweist, dass dieses Arbeitsergebnis verwendet wird und daher nicht gelöscht werden kann. 3. Fragt man den Benutzer, wie er weiter vorgehen würde, wäre das Archivieren der nächste vorgeschlagene Schritt.	Fehlermeldung wurde angezeigt, Archivieren als logischer nächster Schritt erkannt.
5	Da das Arbeitsergebnis bereits verwendet wird, ist	1. Der Benutzer klickt auf die "Bearbeiten"-	Archivierung problemlos möglich.

	das Löschen leider nicht mehr möglich. Daher ist es erforderlich, das Arbeitsergebnis zu archivieren. Archivieren Sie das Arbeitsergebnis.	Schaltfläche 2. Im Dialog "Arbeitsergebnis bearbeiten" ändert er den Status von "Aktiv" auf "Archiviert"	
6	Nun möchten Sie kurz nachsehen, wie viele und welche Arbeitsergebnisse bereits archiviert wurden. Filtern Sie nach denjenigen Arbeitsergebnissen, die den Status "Archiviert" haben.	1. Der Benutzer findet das Filter-Dropdown mit dem nach einem Status gefiltert werden kann. 2. Der Benutzer wählt den Status "Archiviert" aus.	Testperson versuchte es erst mit Filter, dann mit Sortierfunktion. Beide funktionierten korrekt.
7	Es interessiert Sie, ob es bereits ein Arbeitsergebnis mit dem Namen "Datensatz" gibt. Suchen Sie nach diesem Arbeitsergebnis.	1. Der Benutzer bemerkt das globale Suchfeld oder das Suchfeld nach Namen und verwendet dieses, um das gesuchte Arbeitsergebnis zu finden.	Suche funktionierte wie erwartet, evtl. globales Suchfeld deutlicher beschriften.
8	Ein verärgerter Mitarbeiter hat ihnen mitgeteilt, dass sein Lieblingskanton "Thurgau" nicht vorhanden ist. Erstellen Sie die neue Region "Thurgau".	1. Der Benutzer navigiert zur Verwaltungsansicht der Regionen. 2. Der Benutzer verwendet die Schaltfläche "Neue Region anlegen". Es öffnet sich ein Dialog "Neue Ortschaft anlegen". 3. Der Benutzer füllt den Dialog mit den entsprechenden Angaben ab. 4. Als übergeordnete Region wählt der Benutzer intuitiv die Region "Schweiz".	Neue Region korrekt angelegt. Hat als übergeordnete Region intuitiv die Schweiz gewählt.
9	Das Bundesgericht hat entschieden, dass der Kanton Thurgau mit der jährlichen Ernennung der "Apfelkönigin" monarchistische Züge aufweist, welche verfassungswidrig sind. Aus diesem Grund wurde beschlossen, dass der Thurgau künftig nicht mehr zur Schweiz gehört. Der Kanton Thurgau ist neu ein	1. Der Benutzer sucht nach dem Kanton Thurgau mit der Suchleiste oder findet diesen, indem er die Kind-Elemente der Region "Schweiz" begutachtet. 2. Der Benutzer betätigt die Bearbeiten-Schaltfläche und ändert die übergeordnete Region auf "Keine	Änderung wurde korrekt vorgenommen. Testperson hat sofort verstanden, dass die übergeordnete Region bearbeitet werden soll.

	eigenständiges Land. Passen Sie die Stammdaten (Regionen) des Programms entsprechend an.	übergeordnete Region" ab.	
10	Vorbereitung: Es wird auf einem anderen Rechner (durch den Fragesteller) das Programm als Administrator ausgeführt. Während die Testperson den Datensatz in der Bearbeitungsansicht geöffnet hat, wird dieser insgeheim abgeändert. Öffnen Sie die Bearbeitungsansicht der Region "Schweiz" und warten Sie anschliessend auf weitere Anweisungen. (Schweiz insgeheim bearbeiten). In einer Volksabstimmung wurde beschlossen, dass die Schweiz neu in "Helvetische Republik" umbenannt werden sollte, da dieser Name von einer Volksmehrheit als seriöser und geopolitisch vorteilhafter empfunden wurde.	1. Der Benutzer erhält eine Concurrency-Fehlermeldung und bemerkt dadurch, dass eine zeitgleiche Bearbeitung stattgefunden hat. 2. Der Benutzer folgt der Anweisung der Fehlermeldung, lädt die Daten erneut und sieht die Änderungen, die der Kollege (Fragesteller) gemacht hat.	Die resultierende Fehlermeldung wurde beachtet und die Anleitung befolgt. Nach einem Refresh waren die Änderungen sichtbar. Verbesserungsvorschlag: Fehler-Tooltip sollte länger sichtbar sein.

Vorbedingungen veröffentlichtes Projekt verwalten & Projekt archivieren [🔗](#)

Für alle Aufgaben gilt als Vorbedingung, dass der Benutzer eingeloggt ist und die für den Test erforderlichen Berechtigungen hat. Zudem existiert bereits ein veröffentlichtes Projekt, welches der Benutzer verwalten und archivieren kann.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Vor einiger Zeit haben Sie ein Forschungsprojekt zum Thema "KI" mit dem Titel "KI in der Softwareentwicklung" veröffentlicht. Das Abstract weist jedoch massive	1. Der Benutzer sucht und findet das Projekt in der Sparte "Meine Projekte" 2. Der Benutzer klickt auf den Button "Neufassung erstellen"	Vorgang war gut verständlich, die Testperson verstand den Unterschied zwischen Draft und Published und konnte diesen erklären.

	inhaltliche Fehler auf. Erstellen Sie eine neue Version des Projekts, um darin den Fehler zu korrigieren.	3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird. 4. Der Benutzer nimmt in der neuen Version des Projekts die entsprechenden Änderungen vor.	
2	Veröffentlichen Sie diese neue Version organisationsintern.	1. Der Benutzer wählt die Projektsichtbarkeit "Organisationseinheit", falls diese noch nicht gewählt ist. 2. Der Benutzer betätigt den Button "Organisationseinheitsintern veröffentlichen" und bestätigt den Dialog, der sich anschließend öffnet. 3. Der Benutzer versteht, dass das vorherige Projekt nun überarbeitet wurde.	Veröffentlichung hat funktioniert. Vorschlag: UI-Optimierung (Sidenav mit Shortlinks, Statusanzeige)
3	Ihr Chef teilt Ihnen mit, dass Ihr Forschungsprojekt ein grober Fehler aufweist und in diesem Zustand nicht präsentiert werden darf. Archivieren Sie das Projekt.	1. Der Benutzer sucht und findet das Projekt in der Sparte "Meine Projekte" 2. Der Benutzer klickt auf den Button "Archivieren" 3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird.	Konnte erfolgreich ohne Probleme durchgeführt werden.
4	Es hat sich herausgestellt, dass der vermeintlich grobe Fehler nicht in Ihrem Forschungsprojekt, sondern im Projekt eines Kollegen vorlag. Ihr Chef hat sich bei Ihnen entschuldigt und meinte, man solle das Projekt umgehend wieder veröffentlichen. Sorgen Sie dafür, dass das Projekt wieder veröffentlicht wird.	1. Der Benutzer sucht und findet das Projekt in der Sparte "Meine Projekte" 2. Der Benutzer klickt auf den Button "Projekt reaktivieren" 3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird. 4. Der Benutzer startet den Kontroll- / Freigabeprozess.	Reaktivierung und Freigabeprozess wurden korrekt durchgeführt; Hinweis: automatische Aktualisierung der Liste fehlt.

Usability Testing - Student 2

Anmerkungen [↗](#)

- Sieht sehr clean aus
- Toasts auch schön

Vorbedingungen Authentifizierung [↗](#)

Es gilt als Vorbedingung, dass der Benutzer im Besitz eines Gitlab Accounts ist oder die nötigen Berechtigungen hat um einen Gitlab Account für das Gitlab der OST zu erstellen.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Als neuer Benutzer des Portals möchten Sie erst einmal einen Benutzer-Account anlegen. Erstellen Sie ein Benutzerkonto.	1. Der Benutzer findet das Navigationselement im Header, welches zur "Login"-Seite führt. 2. Der Benutzer klick t auf die Schaltfläche "Einloggen mit Gitlab" 3. Der Benutzer meldet sich mit seinem Gitlab-Konto an, sofern er bereits ein Konto hat. Sollte er noch kein Konto haben, versteht er anhand des Informationstextes auf der Anmeldeseite von Gitlab ob er befugt ist, ein Konto zu erstellen und wie er dafür vorgehen soll.	• Profil erfolgreich angelegt
2	Da Sie im Institut für Software als Mitarbeiter angestellt sind, möchten Sie dies auch in Ihrem Profil konfigurieren. Passen Sie das Benutzerprofil entsprechend an.	1. Der Benutzer bemerkt das Navigationselement "Profil" 2. Der Benutzer verwendet die Felder "Organisationseinheit" und "Rolle" um sich als Mitarbeiter beim Institut für Software einzutragen.	• Wählt direkt (ohne Aufforderung) Organisationseinheit und Rolle (Mitarbeiter) aus • Anmerkung: Hinweis für kommunikativer Prüfer gut

Vorbedingungen veröffentlichtes Projekt verwalten & Projekt archivieren [↗](#)

Für alle Aufgaben gilt als Vorbedingung, dass der Benutzer eingeloggt ist und die für den Test erforderlichen Berechtigungen hat. Zudem existiert bereits ein veröffentlichtes Projekt, welches der Benutzer verwalten und archivieren kann.

Nr	Aufgabe	Erwartetes Resultat	Resultat
----	---------	---------------------	----------

1	Vor einiger Zeit haben Sie ein Forschungsprojekt zum Thema "KI" mit dem Titel "KI in der Softwareentwicklung" veröffentlicht. Das Abstract weist jedoch massive inhaltliche Fehler auf. Erstellen Sie eine neue Version des Projekts, um darin den Fehler zu korrigieren.	1. Der Benutzer sucht und findet das Projekt in der Sparte "Meine Projekte" 2. Der Benutzer klickt auf den Button "Neufassung erstellen" 3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird. 4. Der Benutzer nimmt in der neuen Version des Projekts die entsprechenden Änderungen vor.	• Veröffentlicht Projekt zuerst • Erstellt neues Projekt • Altes wird archiviert • Anmerkung: Versionierung etwas verwirrend im ersten Moment
2	Veröffentlichen Sie diese neue Version organisationsintern.	1. Der Benutzer wählt die Projektsichtbarkeit "Organisationseinheit", falls diese noch nicht gewählt ist. 2. Der Benutzer betätigt den Button "Organisationseinheitsintern veröffentlichen" und bestätigt den Dialog, der sich anschliessend öffnet. 3. Der Benutzer versteht, dass das vorherige Projekt nun überarbeitet wurde.	• Führt alles erfolgreich durch
3	Ihr Chef teilt Ihnen mit, dass Ihr Forschungsprojekt ein grober Fehler aufweist und in diesem Zustand nicht präsentiert werden darf. Archivieren Sie das Projekt.	1. Der Benutzer sucht und findet das Projekt in der Sparte "Meine Projekte" 2. Der Benutzer klickt auf den Button "Archivieren" 3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird.	• Führt alles erfolgreich durch
4	Es hat sich herausgestellt, dass der vermeintlich grobe Fehler nicht in Ihrem Forschungsprojekt, sondern im Projekt eines Kollegen vorlag. Ihr Chef hat sich bei Ihnen entschuldigt und meinte, man solle das Projekt umgehend wieder veröffentlichen. Sorgen Sie	1. Der Benutzer sucht und findet das Projekt in der Sparte "Meine Projekte" 2. Der Benutzer klickt auf den Button "Projekt reaktivieren" 3. Es öffnet sich ein Dialog, welcher vom Benutzer gelesen und verstanden wird.	• Fehlendes Komma bei Popup

	dafür, dass das Projekt wieder veröffentlicht wird.	4. Der Benutzer startet den Kontroll- / Freigabeprozess.	
--	---	--	--

Suche (Suchoptimierungen + Suche in PDF + Filter-Usability-Optimierungen) [🔗](#)

Als Vorbedingung gilt, dass bereits Projekte mit den gesuchten Ergebnissen existieren.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Sie möchten mehr über die Projekte an der OST erfahren. Suchen Sie nach Projekten mit dem Arbeitsergebnis „Software“.	Der Benutzer sucht nach „Software“ oder wählt im Filter „Arbeitsergebnis: Software“ aus.	• Findet alles schnell
2	Suchen Sie nun nach den Softwareprojekten mit dem Sponsor „Eureka“.	Der Benutzer setzt zusätzlich den Filter „Sponsor: Eureka“ oder fügt den Suchbegriff „Eureka“ hinzu.	• Anmerkung: Tags sind sehr schön
3	Sie haben von Ihrer Kollegin Julia Weber erfahren, dass Sie erst kürzlich Projekte veröffentlicht hat. Versuchen Sie, diese zu finden.	Der Benutzer sucht nach „Julia Weber“.	• Sucht erst nach Filtereinstellung • Danach in Suchleiste • Findet nichts, wählt dann Filter ab und sucht erneut
4	Setzen Sie Ihre Suche zurück und suchen Sie nach Projekten des Instituts für Architektur und des Institut für Software, die „künstliche Intelligenz“ erwähnen.	• Der Benutzer klickt auf „Suche zurücksetzen“ unterhalb der Filterleiste • Er filtert dann nach den Departementen „Informatik“ und „Gesundheit“ • Benutzer sucht nach „künstliche Intelligenz“ oder verwendet das Schlagwort „Künstliche Intelligenz“.	• Filtert nach Departement • Sucht KI in Suchfeld • Übersieht die Institute in den Organisationseinheiten, da Liste sehr lang

Verwenden Sie nun die Suche eigenständig und probieren Sie verschiedene Filter sowie Suchabfragen aus.

Benutzerverwaltung [🔗](#)

Als Vorbedingung gilt, dass der Benutzer eingeloggt ist und die Rolle „**Administrator**“ besitzt. Ausserdem liegen eine neue Benutzeranfrage sowie eine Änderungsanfrage eines bestehenden Benutzers vor.

Nr	Aufgabe	Erwartetes Resultat	Resultat
1	Navigieren Sie zur Adminseite und ermitteln	Der Benutzer gelangt zur Admin-Seite und sieht	Benutzer findet Adminseite

	Sie, welche Aufgaben aktuell anstehen.	anhand der Badges, dass sowohl eine Kontoaktivierung als auch eine Änderungsanfrage bearbeitet werden sollten.	
2	Der Benutzer „Benutzer Neu“ möchte ein Konto anlegen. Prüfen Sie seine Anfrage, weisen Sie ihm das Institut Architektur zu und vergeben Sie die Rolle „Mitarbeiter“.	• Der Benutzer öffnet die Seite „Neue Konten aktivieren“ • Er findet den Benutzer “Benutzer Neu” und klickt auf “Prüfen” • Er legt als Organisationseinheit “Institut Architektur” und als Rolle “Mitarbeiter” fest und aktiviert das Konto	Benutzer findet Einstellungen und nimmt Änderungen erfolgreich vor
3	Der Benutzer „Livio Mauchle“ möchte Änderungen an seinem Konto vornehmen. Finden Sie die Anfrage und nehmen Sie sie an.	• Der Benutzer öffnet die Seite „Änderungsanfragen prüfen“ • Er findet den Benutzer “Livio Mauchle” und klickt auf “Prüfen” • Er nimmt die Änderungsanfrage an	Benutzer findet auch Änderungsanfrage
4	Der Benutzer „Thomas Bauer“ hat bislang keine Berechtigungen. Vergeben Sie ihm die Rolle „Mitarbeiter“.	• Der Benutzer öffnet die Seite „Benutzer verwalten“ • Er findet den Benutzer “Thomas Bauer” und klickt auf “Bearbeiten” • Er weist ihm die Rolle Mitarbeiter zu und speichert die Änderungen	Benutzer bearbeitet Thomas Bauer erfolgreich
5	Die ehemalige Mitarbeiterin „Julia Weber“ hat die OST verlassen und soll keine Änderungen mehr vornehmen können. Sperren Sie ihr Benutzerkonto.	• Der Benutzer öffnet die Seite „Benutzer verwalten“, findet „Julia Weber“, klickt auf „Bearbeiten“, wählt „Benutzerkonto sperren“ und gibt eine sinnvolle Begründung ein.	Benutzer sperrt Julia Weber mit Begründung
6	Sechs Monate nach ihrem Austritt kehrt „Julia Weber“ als normale Mitarbeiterin ans Institut für Software zurück. Reaktivieren Sie ihr Konto.	• Der Benutzer öffnet die Seite „Gespernte Benutzer“, findet „Julia Weber“, klickt auf „Bearbeiten“, weist erneut die Organisationseinheit Institut für Software und	Benutzer wählt Rolle aus und reaktiviert das Benutzerkonto Anmerkung: Alles intuitiv

		die Rolle Mitarbeiter zu und klickt auf „Benutzerkonto reaktivieren“.	
--	--	---	--

KI-Abstract Generator + Editor Erweiterung [🔗](#)

Als Vorbedingung gilt, dass der Benutzer eingeloggt ist und die Rolle „Mitarbeiter“ besitzt.

Nr	Aufgabe	Erwartetes Resultat	Resultat	
1	Legen Sie ein neues Projekt an. Verschaffen Sie sich zunächst einen kurzen Überblick und erfassen Sie dann eine Projektbeschreibung. Äussern Sie dabei laut Ihre Gedanken.	Der Benutzer legt über „Projektverwaltung → Neues Projekt anlegen“ ein neues Projekt an und bemerkt den Hinweis „Aus angehängtem PDF-Dokument generieren“.	• Erstellt Projekt • Sieht nach kurzer Zeit den Link	
2	Versuchen Sie nun, eine Beschreibung anhand eines PDF-Dokuments automatisch zu generieren.	Der Benutzer klickt auf den Link „Aus angehängtem PDF-Dokument generieren“, lädt ein PDF hoch und verwendet die Schaltfläche zum Erzeugen der Beschreibung aus dem Dokument.	• Etwas Probleme beim Datei hochladen • Danach findet er Button für Generierung von selber • (Generiert erneut nur Inhaltsverzeichnis → Prompt anpassen?)	
3	Die Beschreibung wurde erstellt. Sehen Sie, wie das fertige Projekt später aussehen wird?	Der Benutzer findet die Schaltfläche „Zur Vorschau“ und betrachtet das Projekt in der Vorschauansicht.	• Findet es schnell	
4	Die Beschreibung wirkt noch unstrukturiert. Unterteilen Sie sie in die Abschnitte Problemstellung, Lösung und Fazit und formatieren Sie die Überschriften entsprechend.	Der Benutzer wendet die Formatierungsoptionen für Überschriften korrekt an und gliedert den Text in die drei geforderten Abschnitte.	• Verwendet diese erfolgreich	
5	Fügen Sie im Abschnitt Lösung eine Aufzählung aller durchgeführten Schritte hinzu und heben Sie wichtige Wörter (fett, kursiv oder unterstrichen) hervor.	Der Benutzer nutzt die Listenfunktion für die Schrittauflistung und versteht wie er Hervorhebungen (Fettschrift, Kursivschrift, Unterstreichung) anwenden kann.	• Verwendet auch diese erfolgreich • Anmerkung: Sieht ähnlich aus wie Jira	

9.2 Ergänzende Inhalte

9.2.1 Auswertung der Anzahl Codezeilen

Die Ermittlung der Codezeilen erfolgte nach dem gleichen Verfahren wie in der Studienarbeit. Die folgenden Beschreibungen wurden daher aus dem Vorgängerprojekt übernommen und an den aktuellen Stand angepasst.

Um herauszufinden, wie viele Zeilen Code das Projekt enthält, wird das Tool "cloc" verwendet. Dies kann mit dem folgenden Kommando gemacht werden:

```
cloc --list-file=folders_to_check_for_loc.txt
```

Dabei ist "folders_to_check_for_loc.txt" ein File, welches diejenigen Ordner auflistet, die den von uns geschriebenen Code enthalten. In den Auswertungen in Kapitel 7 wurden vom Umfang her vernachlässigbare Sprachen wie JavaScript und **Cascading Style Sheets (CSS)** im Frontend, sowie Razor, SQL und HTML im Backend weggelassen. Die folgenden Abbildungen zeigen die von cloc generierte Auswertung der Anzahl an Codezeilen für das Frontend, Backend und Unit Tests.

Language	files	blank	comment	code
Vuejs Component	83	926	81	9290
TypeScript	56	298	59	2280
CSS	2	14	1	67
JavaScript	1	10	1	64
SUM:	142	1248	142	11701

Abbildung 9.9: Anzahl Codezeilen im Frontend

Language	files	blank	comment	code
TypeScript	13	186	26	1055
SUM:	13	186	26	1055

Abbildung 9.10: Anzahl Codezeilen der Unit Tests im Frontend

Language	files	blank	comment	code
C#	192	1331	986	8481
SQL	2	66	59	323
Razor	9	16	8	98
HTML	2	0	0	82
SUM:	205	1413	1053	8984

Abbildung 9.11: Anzahl Codezeilen im Backend

Language	files	blank	comment	code
C#	25	799	289	3610
C# Generated	2	8	18	18
SUM:	27	807	307	3628

Abbildung 9.12: Anzahl Codezeilen der Unit Tests im Backend

9.2.2 Setup Anleitung und Dokumentation für Entwickler

README.md

2025-06-12

OST ResearchNavigator

A website for searching and managing information about the projects of all institutes of the OST University of Applied Sciences Eastern Switzerland.

This project was initiated during the Studienarbeit OST ResearchNavigator (formerly "Publikationsapp für Forschungsprojekte") by Livio Mauchle and Simon Hefti and continued during the Bachelor Thesis of the same group.

Overview

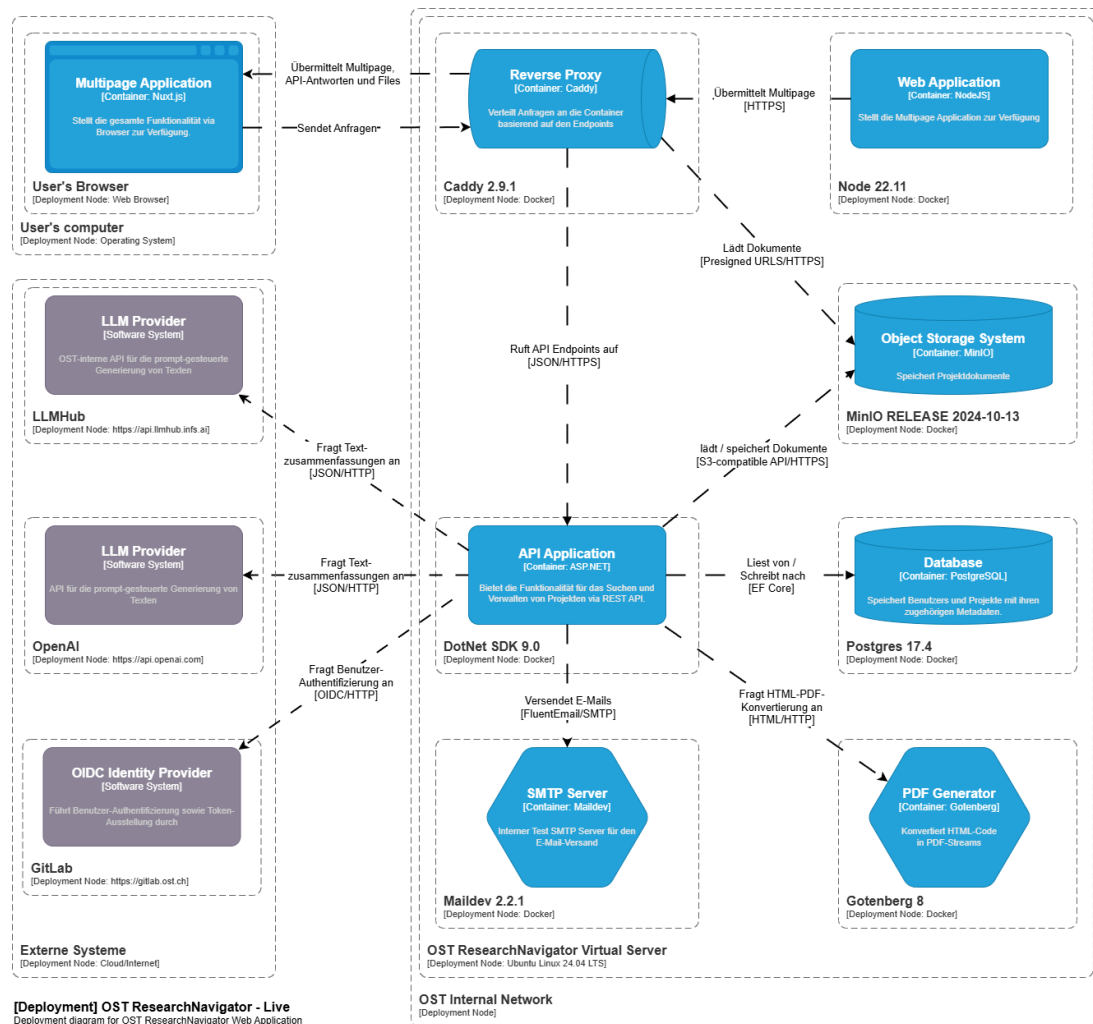
The software system is composed of the following components:

Core Components

- **ASP.NET Core Web API:** Provides functionality for searching, managing, and accessing projects via a REST API.
- **Nuxt.js Frontend:** A multipage frontend application offering the main user interface.
- **PostgreSQL Database:** Stores user data, project data, and associated metadata.
- **Minio Object Storage:** Used for storing project-related documents such as PDFs and images.
- **Gotenberg PDF Generator:** Converts HTML content into PDF streams to enable document export functionality.
- **MailDev SMTP Server:** An internal SMTP server used for testing email delivery during development.

External Services:

- **LLM Providers:** Two large language model providers are integrated, the provider and model can be selected through environment configuration (.env-File).
 - The internal OST-hosted LLMHub
 - OpenAI
- **OIDC Identity Provider:** GitLab is used as the identity provider for user authentication and authorization via OIDC.



The frontend and backend can be run individually as described in their respective README-Files [Backend README](#) and [Frontend README](#).

The whole system can also be run with docker compose (see below)

Known Issues

- **Formatting and Type Checking in Visual Studio:** We heard from other developers that there are issues with formatting and type checking when using Visual Studio. (We primarily develop with WebStorm and JetBrains Rider)

Development Links

- **Frontend Dev:** <http://localhost:3000>
- **Frontend Prod (with Caddy):** <http://localhost>
- **MinIO:** <http://localhost:9000> / <http://localhost:9001>
- **Backend:** <http://localhost:5050>
- **MailDev:** <http://localhost:1080/maildev>

Setup

Most of the configuration is included in the compose.yaml and the Dockerfiles. Some environment variables have to be provided. The root directory must contain a file ".env" with the following variables required by the compose-file. A sample file named ".env.example" has been provided and the setup needs only little reconfiguration (e.g. the Minio Keys).

Environment Variables

Database

Variables used for the database setup and to build the connection string for EF Core

- Server: Name of the postgres container.
- Port: Default 5432
- Database: Created when starting the postgres container the first time.
- Username & Password: Admin user for the postgres database

```
POSTGRES_SERVER=...  
POSTGRES_PORT=5432  
POSTGRES_DB=...  
POSTGRES_USER=...  
POSTGRES_PASSWORD=...
```

GitLab

The GitLab Client ID and Secret have to be created on gitlab.ost.ch under [Preferences -> Applications -> Add new Application](#). The Redirect URI must match the address of the API including the CallbackPath included in the appsettings.json, e.g. http://localhost/api/signin-gitlab. The required scopes are 'openid' , 'profile' and 'email'.

```
GITLAB_CLIENT_ID=...  
GITLAB_CLIENT_SECRET=...
```

Minio

The password, endpoint and keys of the minio fileserver

- Password: The password for the admin user (Default username ROOTNAME). Required for logging into the Minio Console and creating the access keys.
- Bucket Name: Default bucket that contains all project files. Created when starting Minio the first time.
- Endpoint: Host:Port of the Fileserver. The port is set in the docker compose minio configuration (Default localhost:9000).
- Access & Secret Key: Credentials used by the backend to connect to MinIO.They have to be created in the Minio Console first. See below for more information

```
MINIO_ROOT_PASSWORD=...  
MINIO_BUCKET_NAME=...  
MINIO_ENDPOINT=<host>:9000  
MINIO_ACCESS_KEY=...  
MINIO_SECRET_KEY=...
```

CORS

The allowed Cross-Origin-Hosts (without port), separated by comma. Only relevant without caddy, i.e. in the development environment.

```
ALLOWED_ORIGINS=localhost,127.0.0.1
```

Frontend / Caddy

The API base Scheme, Host and IP for configuration of the caddy reverse proxy. For communication with the backend, the API base url is used.

```
API_BASE_SCHEME=http(s)  
API_BASE_HOST=... (e.g. localhost)  
API_BASE_IP=... (e.g. 127.0.0.1)  
API_BASE_URL=.... (e.g. http://localhost:5050)
```

For the development-containers, the UI URL including the port is required

```
UI_DEV_URL=... (e.g. http://localhost:3000)
```

LLM

Keys (and model) for the LLMs used to generate the abstract from a PDF-File. The LLMHub-Token can be created on <https://llmhub.infs.ch/>. Their available models are listed under <https://llmhub.infs.ch/models/>.

By default, the LLMHub-API is used. To change this, adjust the LLM_PROVIDER setting.

```
OPEN_AI_KEY=sk-...  
OPEN_AI_MODEL=gpt-3.5-turbo-0125  
LLMHUB_KEY=...  
LLMHUB_MODEL=Qwen/QwQ-32B-AWQ  
LLM_PROVIDER=LLMHub | OpenAI
```

The URLs for both OpenAI and LLMHub can be configured in the appsettings.json file which can be found [here](#).

E-Mail

SMTP-Server, Port and credentials used for sending notification emails. The settings for the maildev-server, which runs in a docker container, are given below.

```
MAIL_SERVER=localhost
MAIL_PORT=1025
MAIL_USER=user
MAIL_PASSWORD=password
MAIL_SSL=false
```

Additionally, two email addresses have to be specified:

- Admin-Inbox: Email-Address, where emails for user management purposes are sent, i.e. a new user change request.
- Comval-Inbox: Email-Address, where notifications for every project that enters the comval-status are sent

```
ADMIN_INBOX=... (e.g. admin@ost.ch)
COMVAL_INBOX=... (e.g. comval@ost.ch)
```

Features

Configure whether contributor statistics should be displayed on the dashboard

```
SHOW_CONTRIBUTOR_STATISTICS=true/false
```

The email address of the user who is created with the role Administrator and activated when first starting the application. The user still needs to sign in with GitLab, so make sure the email is valid for OST GitLab.

```
DEFAULT_USER_EMAIL=x@ost.ch
```

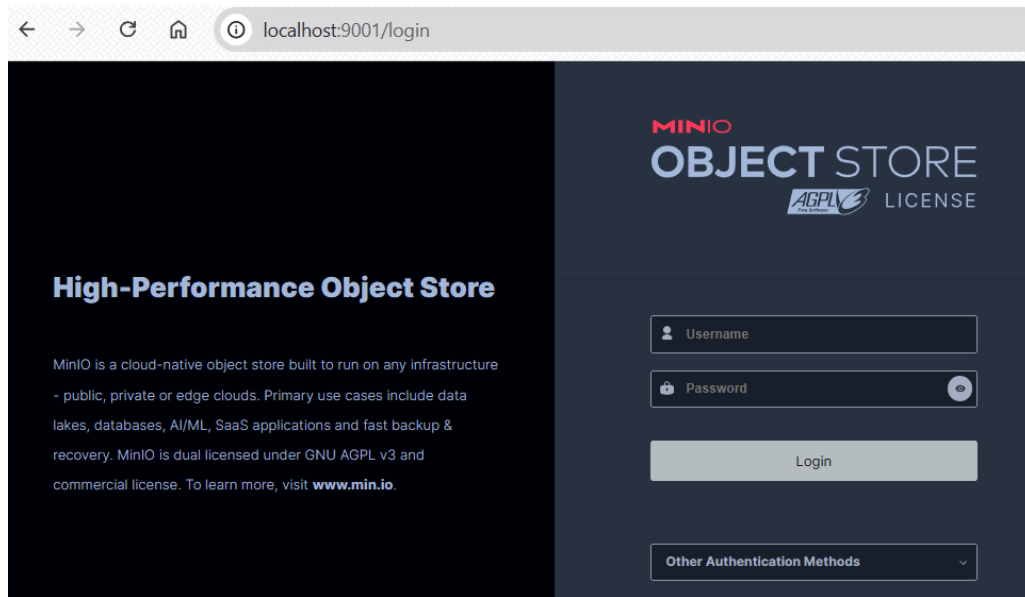
Configure whether test data should be inserted into the database when starting the application. Only works if not projects have been created yet.

```
INSERT_TEST_DATA=false
```

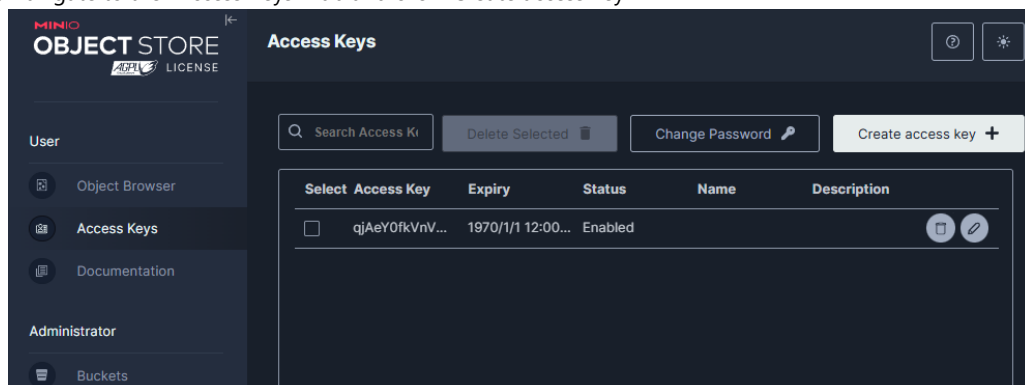
Minio

Creating the Minio Keys

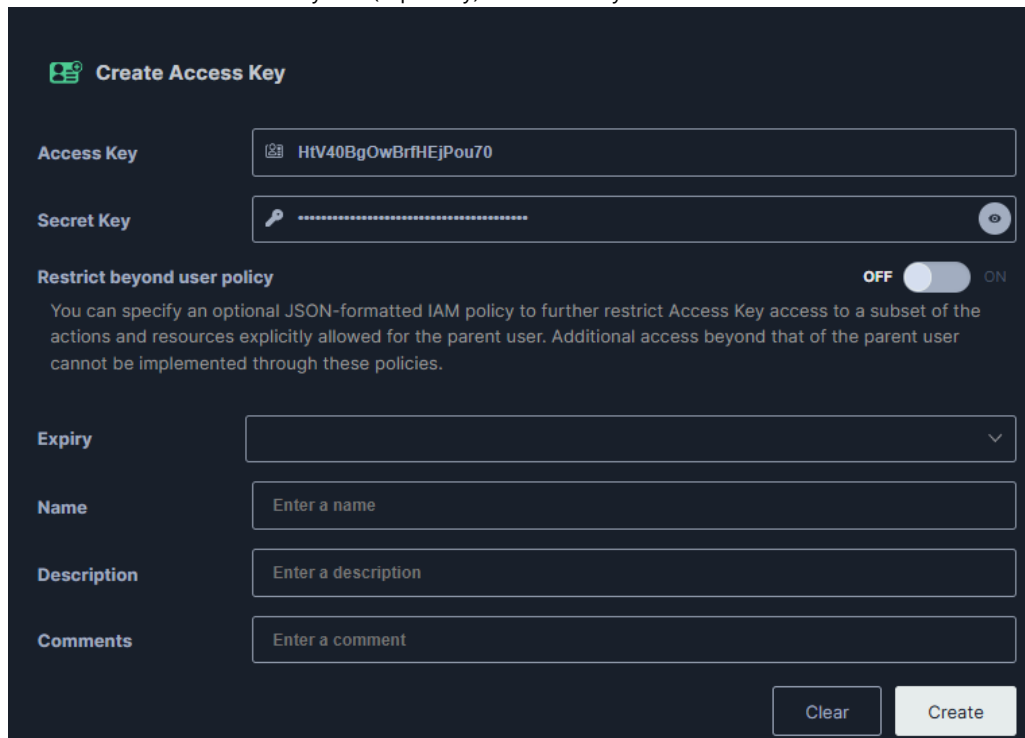
1. Open the console in a webbrowser. The host is the same as for the fileserver, the port is the second one in the docker compose configuration (Default localhost:9001).
2. Sign in: The username is also set in the docker compose file (Default ROOTNAME), the password inside the .env file in the root dir (MINIO_ROOT_PASSWORD).



3. Navigate to the "Access Keys"-Tab and click "Create access key"



4. Write down both the access key and (especially) the secret key



The screenshot shows the 'Create Access Key' page in the AWS IAM console. It features a dark blue header with the title 'Create Access Key' and a green icon. Below the header, there are two input fields: 'Access Key' containing the text 'HtV40BgOwBrfHEjPou70' and 'Secret Key' containing a masked string of dots. To the right of the 'Secret Key' field is a toggle switch labeled 'Restrict beyond user policy', which is currently set to 'OFF'. Below this, there is a paragraph of text explaining that an optional JSON-formatted IAM policy can be specified to restrict access. Further down, there are four more input fields: 'Expiry' (a dropdown menu), 'Name' (placeholder 'Enter a name'), 'Description' (placeholder 'Enter a description'), and 'Comments' (placeholder 'Enter a comment'). At the bottom right, there are two buttons: 'Clear' and 'Create'.

5. Click the "Create"-Button and set the keys in your configuration

Starting the application

All containers can be built and run with the following command

```
docker compose up --build
```

To enable Hot Reloading for the frontend during development, use the following command:

```
docker compose -f compose.dev.yaml up --build
```

Important: Starting the application — especially for the first time — may take a little while. You might see the Nuxt startup screen and then a white page for some time while the Vite server and client finish warming up.

Once the application has finished starting, the console output will look like this:

```
frontend-1 | [nuxt] [i] Compiled router.options.mjs in 675.71ms
frontend-1 | [nuxt] [i] Compiled types/plugins.d.ts in 763.64ms
frontend-1 | [nuxt] [i] Compiled plugins.client.mjs in 777.15ms
frontend-1 | ✓ Vite client built in 593ms
frontend-1 | ✓ Vite server built in 12498ms
frontend-1 | [nitro] ✓ Nuxt Nitro server built in 200265ms
frontend-1 | [i] Vite client warmed up in 49ms
postgres   | 2025-03-21 14:46:58.802 UTC [55] LOG:  checkpoint starting: time
postgres   | 2025-03-21 14:47:25.806 UTC [55] LOG:  checkpoint complete: wrote 241 buffers (1.5%);
sync=2.275 s, total=27.004 s; sync files=159, longest=0.083 s, average=0.015 s; distance=1159 kB, e
frontend-1 | [i] Vite server warmed up in 62922ms
frontend-1 | [i] ✨ new dependencies optimized: primevue/divider, primevue/inputgroup, primevue/icon
dion, primevue/datepicker, primevue/tree, primevue/accordionpanel, primevue/accordioncontent, prime
rimevue/usetoast, primevue/useconfirm, primevue/checkbox
frontend-1 | [i] ✨ optimized dependencies changed. reloading
frontend-1 | [i] ✨ new dependencies optimized: primevue/menubar, primevue/badge
frontend-1 | [i] ✨ optimized dependencies changed. reloading
frontend-1 |
```

Database Backups

A full backup is created via cron job at 02:00 every day. The backups are deleted after 7 days.

To restore the latest backup, run the following command:

```
docker exec -i pg_backup /bin/bash /restore.sh
```

To list all available backups, run:

```
docker exec -i pg_backup ls backups
```

To restore a specific backup, specify its name in the first parameter:

```
docker exec -i pg_backup /bin/bash /restore.sh <db_backup_xxx.sql.gz>
```

Development

Autogenerated Typescript Enums and Interfaces

Types used in the backend controllers are automatically generated from the C# models. The frontend uses these types to ensure type safety. They can be found in types/generatedApiTypes.ts. The types can be generated, using the "Generate Typescript ClientApi" runconfiguration in the backend. Or simply running the command

```
nswag run nswag.json
```

in the same folder where the nswag.json configuration in the backend project lies.

Backend Run Configurations

Run configurations (Rider) that developers should know about include:

- **Different .NET Launch Configurations:** Start the backend project.
- **Run:** Runs the linter on the backend project
- **Generate and Open Report (All in one):** Runs all tests and generates a coverage report. The report is then opened in the browser.
- **Generate Typescript ClientApi:** Generates the typescript types from the C# models. The types are generated in the frontend project.
- **Open Docs:** Opens the generated documentation in the browser.

Backend

Our backend is an ASP.NET Core Web API built with .Net 9 and C#

It can be run locally (without Minio filesystem, MailDev and Gotenberg PDF Generator) or with docker (and the previously mentioned services) as described in the [root README](#). Alternatively, you can also run Minio, MailDev and Gotenberg in Docker and the Backend locally as described below.

Local Setup

1. Installations

The following programs and installations are required / recommended:

- [.NET 9.0 SDK](#)
- Postgres: postgres server (Required) & pgadmin (Recommended)
- JetBrains Rider (Recommended)

2. Clone the Project

Clone the project onto your device and open it in your IDE (preferably Rider)

3. Install Dependencies

Run the following command in your CLI to install all required packages and tools:

```
dotnet restore
```

4. Configuration

The default configuration is set inside the appsettings.json file. For local development, those settings can be overwritten in a new file named appsettings.Development.json, which is ignored by git.

5. Setup your Postgres Database locally

During setup, it is important that the DB connection matches the connection string in your appsettings.*.json. The default user in postgres has the username "postgres" with the password "admin". You are free to create a new postgres user and use it for the database connection, however keep in mind that you have to adjust the configuration accordingly.

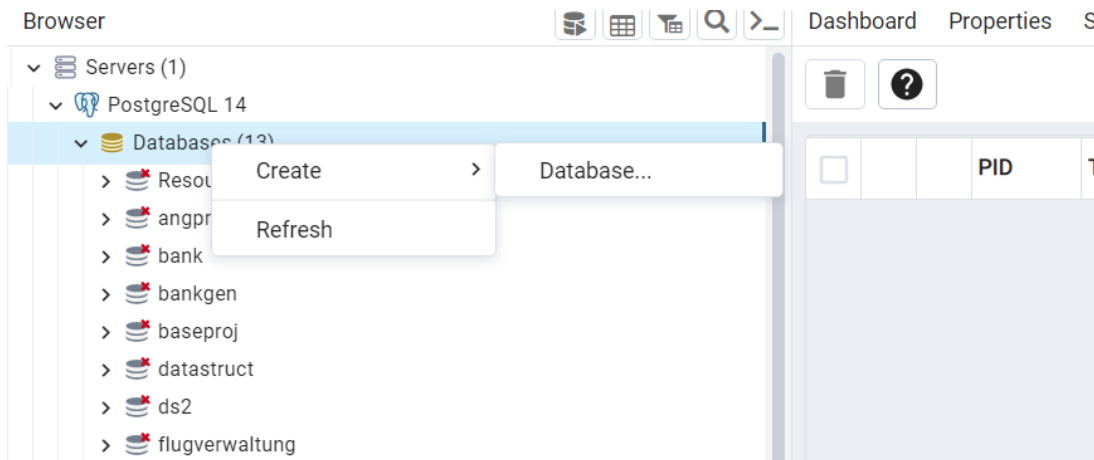
More details about the configuration are described below.

To run the project locally and not in docker, a Postgres database must be created locally. This can be done with PG Admin 4 as follows:

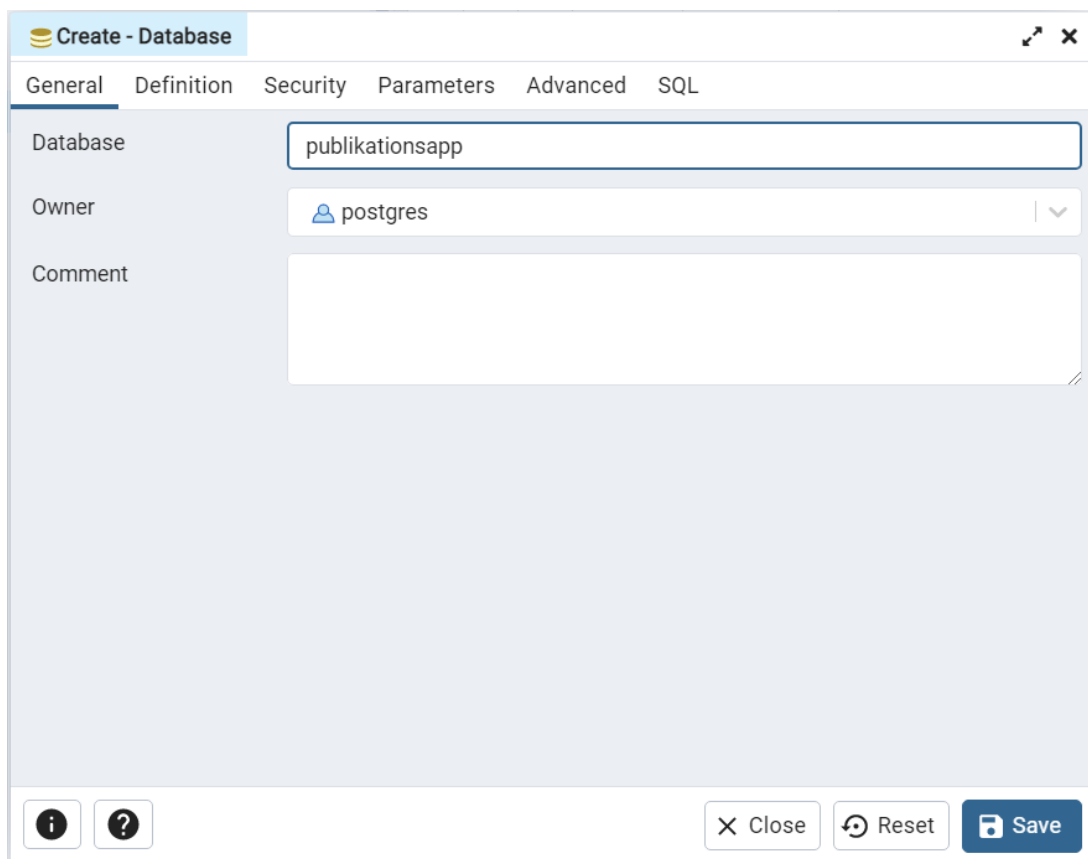
1. navigate to databases -> rightclick -> create -> database

README.md

2025-06-13



2. Give it the name "publikationsapp"
3. Select the user "postgres" (or whatever user you have set in your configuration)
4. save it



5. Create Database and Execute Migrations

To create the database, run the following command:

2 / 4

README.md

2025-06-13

```
cd OST.ResearchNavigator.Infrastructure
dotnet ef --startup-project ../OST.ResearchNavigator.Api/ database update 0
```

This creates the database **without applying** migrations. Migrations are applied automatically when the application starts. We avoid applying them manually at first because a script handles the creation of a database trigger used for the project search vector. Since this trigger prevents updates to certain fields while active, the script temporarily removes and reapplies it during migrations. Automating this ensures smoother development and fewer manual steps.

6. Starting the application

```
cd OST.ResearchNavigator.Api
dotnet run
```

Run Configurations

Predefined run configurations are available in Rider for nearly all commonly used commands.

Tests

Use the following command to run all unit tests:

```
cd UnitTests
dotnet test
```

To generate a coverage report, run the following commands:

```
cd UnitTests

dotnet test -c Debug --results-directory ./TestResults/cobertura --collect:"XPlat
Code Coverage" --test-adapter-path:. --
logger:"junit;LogFilePath=./TestResults/junit/junit-test-
result.xml;MethodFormat=Class;FailureBodyFormat=Verbose"

dotnet reportgenerator -reports:TestResults/cobertura/*/coverage.cobertura.xml -
targetdir:TestResults/coveragereport -reporttypes:Html

Invoke-Item .\TestResults\coveragereport\index.html
```

Coding Guidelines

We follow the Microsoft C# default code style. If necessary, new rules can be added by editing the .editorconfig file.

README.md

2025-06-13

- Prefer using extension method syntax over query syntax.
- Prefer async/await over blocking calls like `.Result` or `.Wait()`
- Enforce style rules via `.editorconfig`

Formatting

We rely on dotnet format to enforce consistency. Commands to format the projects:

```
cd <Project>
dotnet format <Project>.csproj
```

There are also run configurations for Rider to format the projects.

EF Migrations

Commands to add a new database migration:

```
cd OST.ResearchNavigator.Infrastructure
dotnet ef --startup-project ../OST.ResearchNavigator.Api/ migrations add
<migration-name>
```

Trigger (SQL Script)

To keep the `search_vector` column on the `Projects` table always in sync with any changes, we embed a PostgreSQL trigger and helper SQL in each migration:

Purpose: Automatically rebuild the full-text search vector whenever a project or any related data (tags, contributors, type, sponsor, region, metadata, etc.) changes.

This trigger runs at runtime in the database:

- BEFORE INSERT/UPDATE on `Projects`
- AFTER UPDATE on related tables (`SearchTags`, `AspNetUsers`, `ProjectTypes`, `Sponsors`, `Regions`, `OrganisationalUnits`, `Metadata`) to “touch” affected projects.

For in order to add this trigger to the database, Up and Down Migrations were written using SQL scripts. These are placed under: `OST.ResearchNavigator.Infrastructure\Migrations/SqlScripts\`

The Scripts are applied during an EF migration using the `MigrationSqlHelper` class. Using the following commands:

```
var upSql = MigrationSqlHelper.LoadScript("fts_project_up.sql");
migrationBuilder.Sql(upSql);
```

Frontend

Built with [Nuxt 3](#) to learn more.

Setup

Installing the dependencies:

```
# npm
npm install
```

For local development with API communication and without docker, you also need to specify the API Endpoint in a .env-File located in this directory

```
API_BASE_URL=... # e.g. http://localhost:5050 for local development
```

If the frontend service is started with docker compose, this is not required.

Development Server

Start the development server on <http://localhost:3000>:

```
# npm
npm run dev
```

Production

Build the application for production:

```
# npm
npm run build
```

Locally preview production build:

```
# npm
npm run preview
```

Lint

Display lint errors and warnings:

README.md

2025-06-13

```
npm run lint
```

Fix lint errors and warnings (where possible):

```
npm run lint:fix
```

Test

Run all tests:

```
npm run test
```

Run tests with code coverage:

```
npm run test:ci
```

9.2.3 Deployment Anleitung für OST VM

Deployment instructions

1. Open the console and connect to the server:

```
ssh ins@10.8.36.127
```

(then enter the server password)

2. Create a new entry in the deployment list (below):

You can check the status of the Docker containers with the following command:

```
sudo docker ps
```

(then enter the server password)

3. Pull the latest changes:

```
cd repos/publikationsapp-fuer-forschungsprojekte/
```

```
sudo git pull --rebase
```

No login required → Project Token with read_repository permission and reporter role is stored

https://gitlab.ost.ch/livio.mauchle/publikationsapp-fuer-forschungsprojekte/-/settings/access_tokens

(expires around 8th of December 2025)

4. If any of the files containing “localhost” have been changed, resolve the merge conflict. Keep the variant with “10.8.36.127”.

5. If any configuration has to be changed, modify the .env file under:

```
repos/publikationsapp-fuer-forschungsprojekte/.env
```

6. Shut down the container:

```
sudo docker compose down
```

7. Rebuild and start the container:

```
sudo docker compose up --build
```

9.2.4 Benutzeranleitung

user_instructions.md

2025-06-13

Benutzeranleitung

In diesem Dokument sind die Schritte zur Bedienung der wichtigsten Funktionen der Anwendung beschrieben.

Projektsuche mit Filtern

Die Startseite ist die Übersicht der veröffentlichten Projekte. Führen Sie die folgenden Schritte durch, um diese zu filtern.

1. Um nach Wörtern im Titel und Abstract sowie textbasiert nach Einstellungen wie Schlagwörtern, Autoren, Organisationseinheiten und Sponsoren zu suchen, können Sie diese in die Textleiste oben in der Mitte eingeben und auf "Los" klicken oder die Enter-Taste drücken.
2. Um nach einer Auswahl von Kriterien wie den Organisationseinheiten oder nach bestimmten Schlagwörtern (Themen) zu filtern, verwenden Sie die Filterleiste auf der linken Seite. Die einzelnen Filterkriterien können aufgeklappt werden. Falls Sie die Website auf einem kleinen Gerät wie einem Handy aufrufen, müssen Sie die Filter erst mit einem Klick auf "Filter" anzeigen.
3. Ist bei einem Filterkriterium keine Auswahl getroffen, wird das Kriterium nicht verwendet und die Projekte somit nicht danach gefiltert. Falls mehrere Werte ausgewählt sind, werden alle Projekte angezeigt, die einem der Werte entsprechen (z.B. falls drei Organisationseinheiten ausgewählt sind, werden alle Projekte angezeigt, an denen mindestens eines davon beteiligt war).
4. Mit "Filter zurücksetzen" können Sie die Filterauswahl leeren.

Forschungsprojekte der OST

Forschungsprojekte der OSTDashboardLoginOST
Forschungstechnik

Suche nach Titel, Autor, Abstract

Suche...Los

10 Treffer

Arbeitsergebnis	▼	Software 2024 AI for Climate Themenbereiche: Umwelt und Energie; Innovation und Management Beteiligte Organisationseinheiten: Institut für Architektur; Institut für Computational Engineering; Mitwirkende Person: Simon Hefti This project aims to leverage artificial intelligence to analyze climate data and predict future trends in climate change. The tool supports researchers by providing a platform to run complex models and visualize climate impacts over time.
Schlagwörter	▼	
Orte	▼	
Sponsor	▼	
Durchführungszeitraum	▼	Datensatz 2024 Smart Agriculture Themenbereiche: Technologie; Gesund leben und altern Beteiligte Organisationseinheiten: Institut für Wissen, Energie und Rohstoffe Zug; Institut für Mikrotechnik und Photonik; Mitwirkende Person: Livio Mauchle This project focuses on using IoT and machine learning to improve agricultural practices by collecting and analyzing data from various sensors in the field.
Datum der Veröffentlichung	▼	
Organisationseinheiten	▼	

Filter zurücksetzenFilter anwenden

Software | 2024

5. Möchten Sie zu einem Projekt weitere Informationen erhalten, klicken Sie in der Liste auf das Projekt. Damit werden Sie zur Detailseite weitergeleitet.
6. Auf der Detailseite können Sie mit der Navigation auf der rechten Seite (Überblick, Beschreibung und Details) zu einem bestimmten Abschnitt springen. Alternativ können Sie auch herunterscrollen, um alle Details, Beschreibungen und Dateien zum Projekt zu sehen.

7. Mit einem Klick auf "Als PDF exportieren" rechts über der Beschreibung können Sie ein PDF mit Titel und Beschreibung generieren.
8. Mit einem Klick auf "Zurück zur Startseite" (oben links) können Sie zur Projektsuche zurückkehren

Forschungsprojekte der OST Forschungsprojekte der OST Dashboard Login OST Ostschweizer Fachhochschule

[← Zurück zur Projektsuche](#)

- Überblick
- Beschreibung
- Details

ResearchNavigator



Durchführungszeitraum
1.4.2025 - 8.4.2025

Arbeitsergebnis
Forschungsprojekt

Schlagwörter
Technologie
Gesundheit und Soziales

Projektdateien
Dokumentation als PDF

Mitwirkende
Dr. Thomas Bauer
Prof. Julia Weber

Beschreibung [Als PDF exportieren](#)

In der modernen Forschungswelt ist die effiziente Verwaltung und Verbreitung von wissenschaftlichen Publikationen entscheidend für den Erfolg von Forschungsprojekten und die Karriere von Wissenschaftlern. Der ResearchNavigator ist eine innovative Webanwendung, die entwickelt wurde, um Forschern eine integrierte Lösung für das Organisieren, Veröffentlichen und Teilen ihrer Forschungsergebnisse zu bieten. Die Plattform ermöglicht es Forschern, ihre Projekte zentral abzuliegen und bietet gleichzeitig Funktionen zur Zusammenarbeit mit Kollegen, zur Einholung von Feedback und zur Verwaltung Prüfprozessen. Durch eine intuitive Benutzeroberfläche können Wissenschaftler ihre Publikationen einfach kategorisieren und die Sichtbarkeit ihrer Arbeiten erhöhen, indem sie sie öffentlich zugänglich machen oder gezielt mit bestimmten Nutzergruppen teilen. Die Publikationsapp bietet damit eine umfassende Lösung, die den gesamten Publikationsprozess unterstützt und es Forschern ermöglicht, sich stärker auf ihre Forschung zu konzentrieren, während sie gleichzeitig ihre wissenschaftliche Sichtbarkeit und Vernetzung erhöhen.



Login

Um Projekte zu erfassen und um OST- oder Instituts-interne Projekte anzusehen, müssen Sie sich zuerst anmelden

1. Navigieren Sie mit einem Klick auf "Login" in der Titelleiste auf die Login-Seite.
2. Klicken Sie auf auf "Einloggen mit GitLab".

Login Forschungsprojekte der OST Dashboard Login OST Ostschweizer Fachhochschule



Willkommen

Wer darf ein Konto erstellen?

[Einloggen mit GitLab](#)

3. Sie werden auf die GitLab OST Seite weitergeleitet. Klicken Sie dort auf "OST SSO" und melden Sie sich an.

GitLab OST

<https://gitlab.ost.ch> ist aus dem Internet ohne VPN zugänglich.

Diese Ablage steht allen Angehörigen der OST (Studierende, Mitarbeitende) für sämtliche Aktivitäten in den Leistungsbereichen Forschung, Lehre, Weiterbildung und Dienstleistungen zur Verfügung. Externe, die in diesen Leistungsbereichen mit der OST zusammenarbeiten, können gitlab.ost.ch ebenfalls nutzen. Der Zugriff für Externe muss durch einen OST-Mitarbeitenden über das ICT Support Portal <https://ict-support.ost.ch> beantragt werden.

OST-Angehörige melden sich mit ihrem OST-Account über "OST SSO" an.

Externe melden sich mit den ihnen mitgeteilten Zugangsdaten ("Username or email" und "Password") an.



Username or primary email

Password

[Forgot your password?](#)

☐ Remember me

Sign in

By signing in you accept the [Terms of Use](#) and acknowledge the [Privacy Statement and Cookie Policy](#).

or sign in with

 OST SSO

☐ Remember me

4. Klicken Sie auf der folgenden Seite auf "Authorize OST ResearchNavigator".
5. Sie werden zur Profilbearbeitung weitergeleitet.

OST ResearchNavigator is requesting access to your account on GitLab OST.

 Simon Hefti · @simon.hefti

- > Authenticate using OpenID Connect
- > Allows read-only access to the user's personal information using OpenID Connect
- > Allows read-only access to the user's primary email address using OpenID Connect

 Make sure you trust **OST ResearchNavigator** before authorizing.

Simon Hefti added this OAuth application 3 months ago.

Authorize OST ResearchNavigator

Cancel

6. Wählen Sie dort Ihre Rolle und Ihre Organisationseinheit aus und Klicken Sie auf "Profiländerung anfordern".
7. Warten Sie, bis ein Administrator Ihre Kontoanforderung überprüft hat. Sie werden per E-Mail benachrichtigt.
8. Bei jeder Profiländerung wird eine neue Anfrage an die Administratoren gestellt, welche wiederum überprüft werden muss, bevor die Änderung in Kraft tritt.

Forschungsprojekte der OST | Dashboard | Profil | (+) Abmelden

Benutzerprofil
Profil bearbeiten

Profil bearbeiten

Aktuelle Version

Anrede
Divers

Titel
-

Vorname
Simon

Nachname
Heftl

Organisationseinheit
Bachelor Data Science

Rolle
Administrator

Prüfer Kommunikation
Nein

Änderungsanfrage

Anrede*
Divers

Titel
Titel

Vorname*
Simon

Nachname*
Heftl

Organisationseinheit*
Bitte wählen

Die Angabe der Organisationseinheit ist erforderlich.

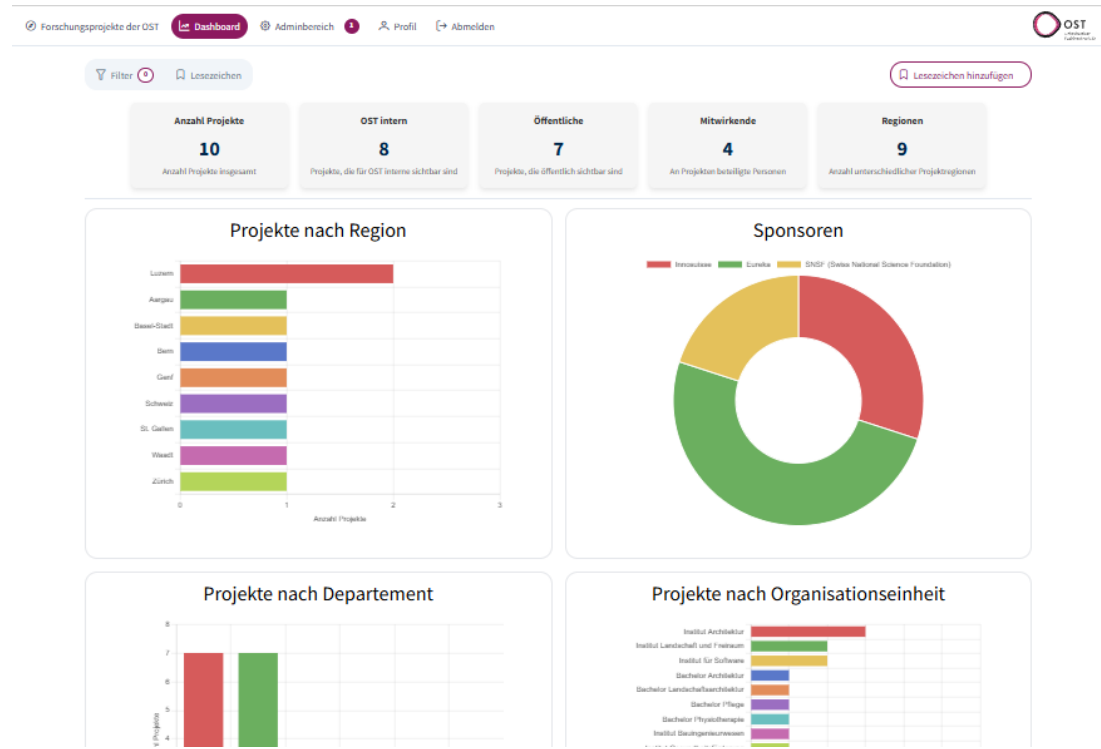
Rolle*
Ohne Berechtigungen

Prüfer Kommunikation ☐

Profiländerung anfordern

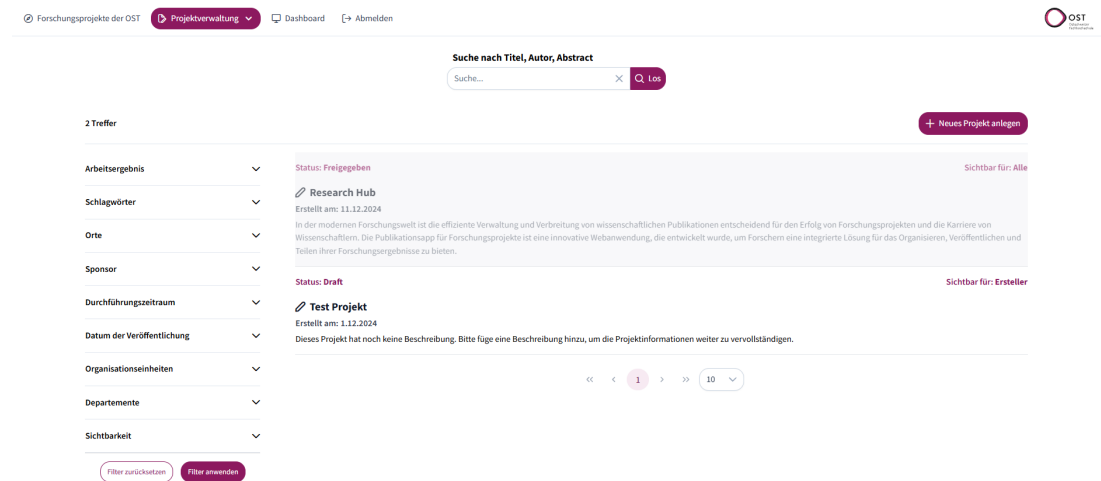
Dashboard

1. Das Dashboard erreichen Sie mit einem Klick auf "Dashboard" in der Titelleiste
2. Hier sehen Sie Statistiken zu den Projekten nach verschiedenen Kriterien
3. Um Filter auf die Projekte anzuwenden, aus denen die Statistiken berechnet werden, klicken Sie auf "Filter" (oben links).
4. Die einzelnen Filter und die Anwendung dieser funktioniert wie bei der Projektsuche. Nur die Filter "Schlagwörter" und "Sichtbarkeit" gibt es hier nicht.
5. Mit einem Klick auf "Lesezeichen hinzufügen" können die aktuellen Filtereinstellungen benannt und gespeichert werden.



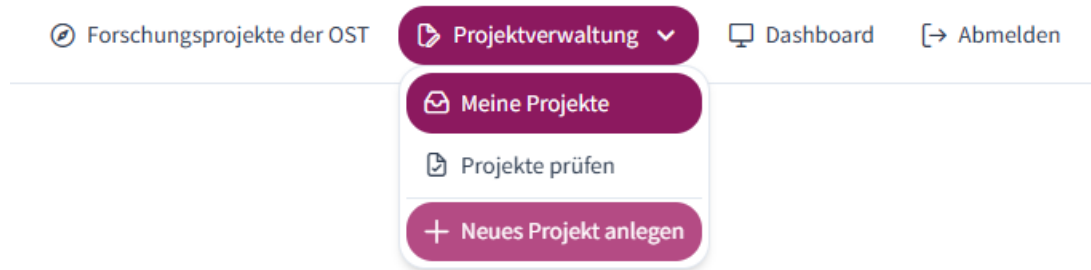
Projektverwaltung

1. Wenn Sie angemeldet sind, können Sie mit einem Klick auf "Projektverwaltung" in der Titelleiste die von Ihnen erfassten Projekte anzeigen.
2. Die Suche und Filter auf dieser Seite funktionieren wie auf der Startseite.
3. Um ein Projekt zu bearbeiten oder freizugeben, klicken Sie es in der Liste an.



Projekterstellung

1. Ein neues Projekt können Sie mit einem Klick auf "Neues Projekt anlegen" oben rechts in der Projektverwaltung oder alternativ in der Titelleiste unter Projektverwaltung -> Neues Projekt anlegen



2 Treffer

Projektbearbeitung

1. Nach der Erstellung eines neuen Projektes oder dem Klick auf ein Projekt in der Projektverwaltung öffnet sich die Projekterfassungsseite.
2. Hier können Sie alle Informationen zum Projekt in die dafür vorgesehenen Felder eintragen. Mit einem Stern (*) gekennzeichnete Felder sind Pflichtfelder.
3. Die Felder Kurzbeschreibung sowie Beschreibung lassen sich mithilfe von KI aus dem Inhalt einer PDF-Datei generiert werden (weitere Informationen im nächsten Abschnitt "File Upload").
4. Mit einem Klick auf "Speichern" können Sie jederzeit den aktuellen Zwischenstand speichern und später mit der Erfassung weiterfahren.
5. Falls Sie ein falsches Projekt erfasst haben, können Sie es mit einem Klick auf "Löschen" unwiderruflich entfernen.
6. Mit einem Klick auf "Vorschau" können Sie die Ansicht aktivieren, welche später auf der Detailseite angezeigt wird.

[← Zurück zu meinen Projekten](#)[📄 Zur Vorschau](#)[✓ Zur fachlichen Prüfung einreichen](#)[Löschen](#)[Speichern](#)

OST ResearchNavigator

Status: Entwurf

Titel & Beschreibung

Titel des Projekts*

OST ResearchNavigator



21 / 120

Beschreibung aus PDF-Dokument generieren ✎

Lade ein PDF-Dokument hoch und lass dir daraus ganz bequem die Beschreibung und Kurzbeschreibung des Projekts generieren.

[Zum Dateupload](#)

Kurzbeschreibung des Projekts*

Eine kurze Beschreibung des Projekts, die in der Liste der Suchergebnisse angezeigt wird.

OST ResearchNavigator: Zentrale Plattform für OST-Projekte mit Such- und Filterfunktionen. Nuxt.js, ASP.NET Core, PostgreSQL und MinIO. Verbessert Kommunikation und Transparenz.

177 / 300

Beschreibung des Projekts*

File Upload

1. Um Dateien hochzuladen, scrollen Sie in der Projekterfassung nach unten. Dort findest du den Abschnitt "Datei wählen und hochladen".
2. Folgen Sie den dort beschriebenen Schritten 1-3, um eine Datei hinzuzufügen.
3. Achten Sie auf den Dateityp! Für die Dateitypen "Bild", "Titelbild" und "Video" können nur entsprechende Dateiformate hochgeladen werden.
4. Die Dateien werden nach dem Hochladen in der Liste darunter angezeigt.
5. Mit einem Klick auf den Dateinamen können Sie die Datei im Browser öffnen oder herunterladen.
6. Um die Datei wieder zu löschen, klicken Sie auf das Mülleimer-Symbol rechts davon
7. Um die Kurzbeschreibung und Beschreibung mit KI zu generieren, klicken Sie auf "Beschreibungen generieren". Klicken Sie NICHT in das Textfeld, während der Vorgang läuft, sonst wird der generierte Text möglicherweise nicht korrekt angezeigt.

Medien & Dokumente**Datei wählen und hochladen (optional)**

1. Wähle eine Datei, die du hochladen möchtest.
 2. Wähle den passenden Dateityp aus, dieser bestimmt die Art und Weise wie der hochgeladene Inhalt angezeigt wird.
 3. Klicke auf "Hochladen" um die Datei dem Projekt hinzuzufügen.
 4. (optional) Verwende die Schaltfläche "Beschreibungen generieren" um die Beschreibung und Kurzbeschreibung des Projekts zu generieren.
- Die maximale Dateigröße beträgt 50 MB.

+ Datei auswählen Datei hochladen Abbrechen

Ziehen Sie eine Datei per Drag & Drop hierher, um sie hochzuladen

Hochgeladene Dateien

Dateiname	Dateityp	
SA_OST_ResearchNavigator_HS24.pdf	Dokument	Beschreibungen generieren

Projekt zur Prüfung einreichen

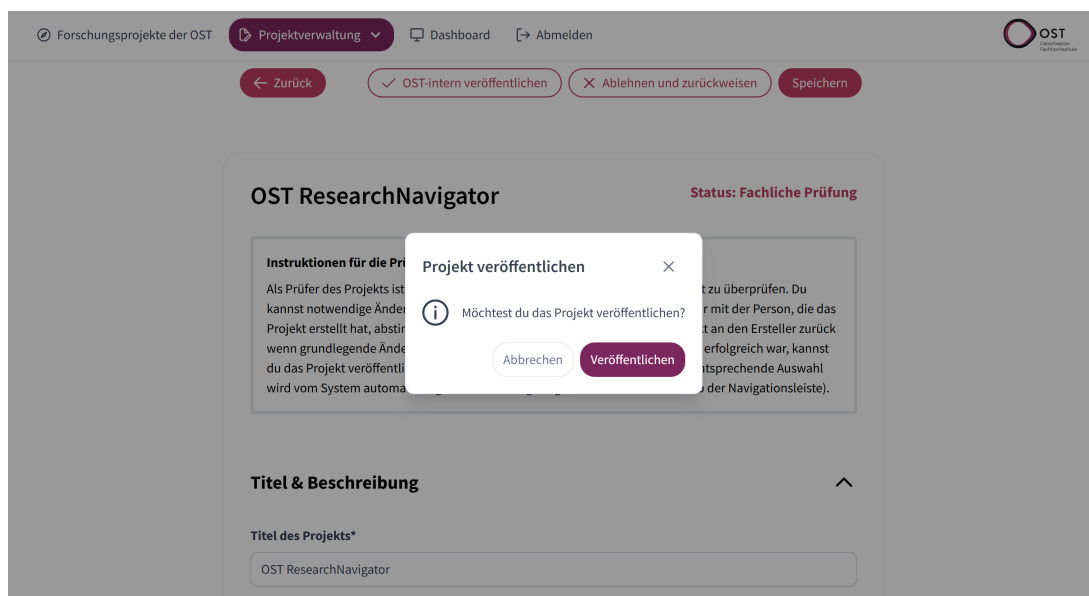
1. Sobald alle Pflichtfelder ausgefüllt sind, können Sie das Projekt für die fachliche Prüfung freigeben.
2. Klicken Sie dafür auf "Zur fachlichen Prüfung einreichen"
3. Im geöffneten Fenster wählen Sie die gewünschte fachliche Prüfperson und klicken dann auf "Zur Prüfung einreichen". VORSICHT: Sie können das Projekt danach NICHT mehr bearbeiten!
4. Sie werden auf Ihre Projektverwaltungsseite weitergeleitet. Das Projekt wird Ihnen noch angezeigt, ist aber ausgegraut und Sie können es nicht mehr anklicken.

The screenshot shows the 'Projekt zur Prüfung einreichen' modal dialog. The dialog has a title bar with a close button. The main content area contains the text: 'Wähle eine geeignete Person aus, die dein Projekt auf fachliche Kriterien prüfen soll.' Below this is a 'Prüfer' field with a dropdown menu showing 'Livio Mauchle'. At the bottom of the dialog are two buttons: 'Abbrechen' and 'Zur Prüfung einreichen'. The background shows the 'OST ResearchNavigator' project page with a 'Status: Draft' label and a 'Zur fachlichen Prüfung einreichen' button.

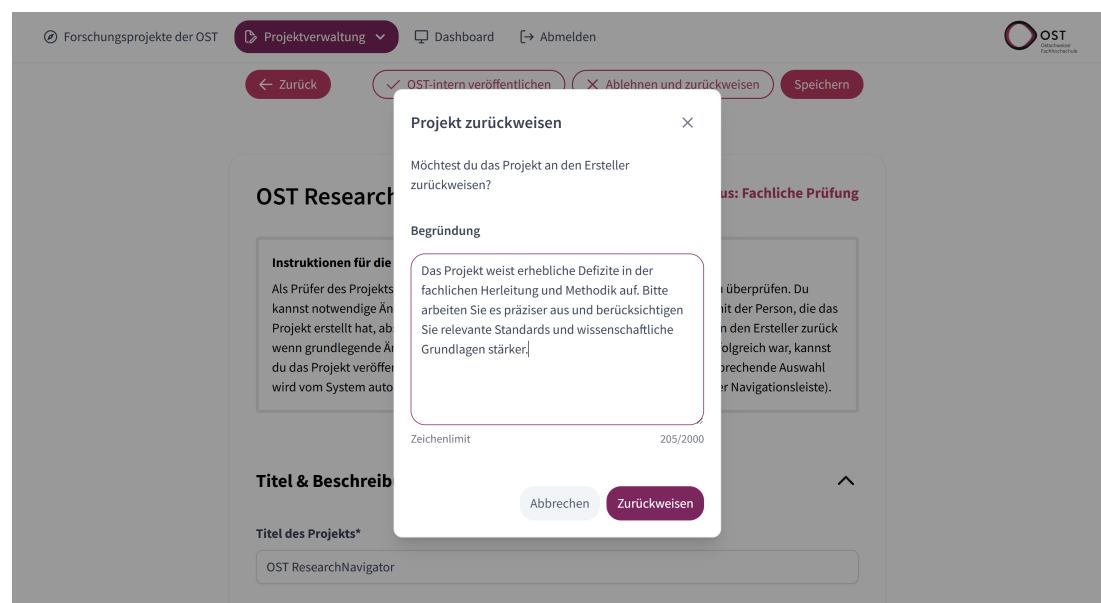
5. Sobald sowohl die fachliche als auch die kommunikative Prüfperson Ihr Projekt validiert hat, wird es freigeschaltet und ist per Suche auffindbar
6. Wird das Projekt von einer Prüfperson zurückgewiesen, kommt es für die Überarbeitung zurück in Ihre Projektverwaltung

Projekt validieren

1. Navigieren Sie über Projektverwaltung -> Projekte prüfen auf die Prüferübersicht.
2. Darauf sehen Sie alle Projekte, welche Ihnen für die fachliche Prüfung zugewiesen sind sowie alle Projekte, die aktuell für eine kommunikative Prüfung eingereicht sind.
3. Mit einem Klick auf eines der Projekte können Sie es in der Erfassungsansicht öffnen.
4. Sie können alle Informationen zum Projekt überarbeiten.
5. Wenn Sie mit dem Projekt zufrieden sind, können Sie es veröffentlichen bzw. weiterreichen. Im Falle einer fachlichen Prüfung klicken Sie dafür auf "OST-intern veröffentlichen" oder "Zur Kommunikationsprüfung einreichen", bei einer kommunikativen Prüfung auf "Veröffentlichen". Bestätigen Sie danach mit einem erneuten Klick auf "Veröffentlichen". VORSICHT: Die Veröffentlichung kann NICHT rückgängig gemacht werden.

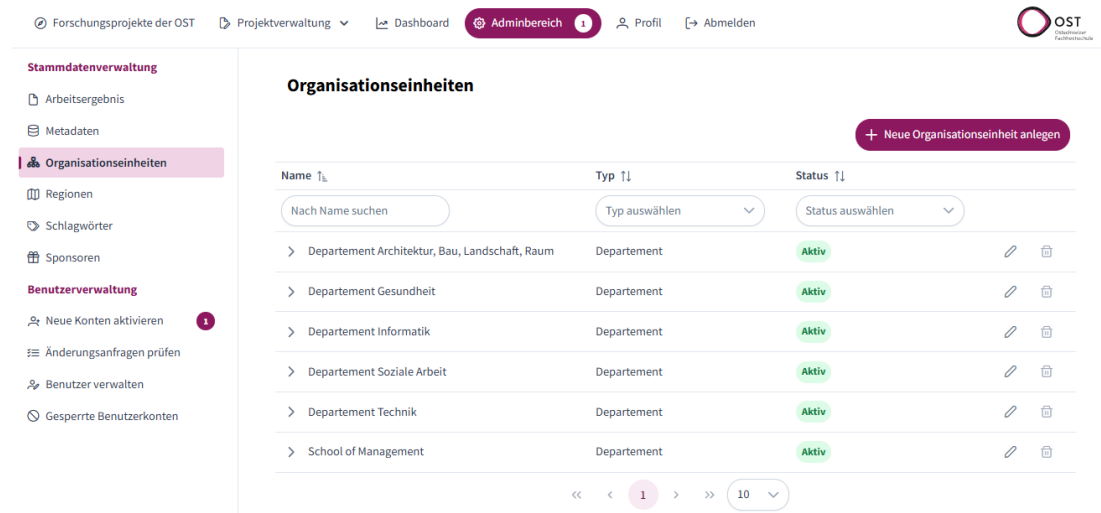


6. Sind Sie mit einem Projekt unzufrieden, können Sie es auch zurückweisen. Klicken Sie dafür auf "Ablehnen und zurückweisen"
7. Geben Sie eine Begründung ein, z.B. was fehlt oder was besser beschrieben werden soll.
8. Mit einem Klick auf "Zurückweisen" wird das Projekt an den Erfasser zurückgereicht, welcher es nochmals überarbeiten kann



Stammdaten verwalten (Administrator)

1. Navigieren Sie zum "Adminbereich"
2. Wählen Sie einen Stammdatentyp (z.B. Arbeitsergebnis, Metadaten oder Organisationseinheiten) aus. Sie sehen eine Auflistung aller vorhandenen Optionen, inklusive archivierten. (Die folgenden Beispiele handeln von den Organisationseinheiten, das Prinzip und die Ansicht sind aber bei allen Stammdaten dasselbe.)
3. Klicken Sie auf das Stift-Symbol, um einen Datensatz zu bearbeiten oder zu archivieren.
4. Klicken Sie auf das Mülleimer-Symbol, um einen Datensatz zu löschen. Achtung: Ein Datensatz kann nicht gelöscht werden, wenn er in Verwendung ist oder untergeordnete Datensätze besitzt. In diesem Fall können Sie ihn archivieren.
5. Klicken Sie auf "Neue x anlegen" um einen Datensatz hinzuzufügen.



5. Bei einem Klick auf den Stift oder hinzufügen öffnet sich ein Popup.

6. Felder, die mit einem * markiert sind, müssen ausgefüllt werden
7. Um den Datensatz zu archivieren, setzen Sie den Status aus "Archiviert".
8. Klicken Sie nach der Bearbeitung auf "Speichern"

Organisationseinheit bearbeiten ×

Name*

Typ*

Beschreibung

Abkürzung

Leiter der Organisationseinheit

Übergeordnete Organisationseinheit*

Status*

AbbrechenSpeichern

Benutzerverwaltung (Administrator)

1. Um neue Benutzeraccount zu aktivieren, klicken Sie in der Navigation auf der linken Seite oder auf der Startseite des Adminbereichs auf "Neue Konten aktivieren"
2. Sie sehen eine Auflistung aller beantragten, noch nicht aktivierten Benutzerkonten.
3. Mit einem Klick auf "Prüfen" öffnet sich ein Popup. Darin sehen Sie die Einstellungen, welche der Benutzer getroffen hat. Sie können das Konto aktivieren, die Einstellungen zuerst bearbeiten oder das Konto sperren, falls der Benutzer nicht zugelassen ist.

Neue Konten aktivieren

Suche: Tabelle durchsuchen

E-Mail ↑↓	Name ↑↓	Organisationseinheit ↑↓	Rolle ↑↓	Aktionen
Nach E-Mail suchen neu@ost.ch	Nach Name suchen Benutzer Neu	Nach Organisationseinheit Departement Architektur, Bau, Landschaft, Raum	Nach Rolle suchen Ohne Berechtigungen	Prüfen

Navigation: << < 1 > >> 5

4. Für die Überprüfung der Änderungsanfragen von bestehenden Benutzerkonten navigieren Sie zu "Änderungsanfragen prüfen".
5. Auch hier öffnet sich mit einem Klick auf "Prüfen" ein Popup.
6. Im Popup sehen Sie die aktuellen sowie die beantragten Werte nebeneinander.
7. Sie können die Werte auch hier bearbeiten, annehmen oder das Konto sperren. Zusätzlich können Sie hier auch einfach die Änderungsanfrage ablehnen, wodurch die aktuelle Version bestehen bleibt.

Änderungsanfrage prüfen

Aktuelle Version	Änderungsanfrage
Anrede Divers	Anrede* Divers
Titel -	Titel Titel
Vorname Simon	Vorname* Simon
Nachname Hefti	Nachname* Hefti
Organisationseinheit Bachelor Data Science	Organisationseinheit* Bachelor Betriebsökonomie
Rolle Hochschulleitung	Rolle* Hochschulleitung
Prüfer Kommunikation Nein	Prüfer Kommunikation ☐

Buttons: Abbrechen, Benutzerkonto sperren, Änderung ablehnen, Änderung annehmen

8. Unter "Benutzer verwalten" können Sie alle aktiven Benutzerkonten bearbeiten.
9. Öffnen Sie dafür mit einem Klick auf "Bearbeiten" die Einstellungen des Benutzers.
10. Im geöffneten Popup können Sie den Account sperren oder die Einstellungen bearbeiten.
11. Unter "Gesperrte Benutzerkonten" können Sie die gesperrten Benutzerkonten ansehen und reaktivieren.

9.2.5 API Dokumentation mittels Swagger UI

07.06.25, 10:39

Swagger UI



Select a definition

OST.ResearchNavigator.Api v1

My Title

1.0.0 OAS 3.0

<http://localhost:5050/swagger/v1/swagger.json>

Servers

http://localhost:5050

Account

GET /api/account/settings

GET /api/account/change-request

POST /api/account/change-request

Parameters

No parameters

Request body required

application/json

```
{
  "id": 0,
  "userId": 0,
  "requestedTitle": "string",
  "requestedGender": "string",
  "requestedFirstName": "string",
  "requestedLastName": "string",
  "requestedOrgUnitId": 0,
  "requestedRole": "OrgUnitEmployee",
  "requestedIsCommValidator": true,
  "status": "Pending",
  "comment": "string",
  "createdAt": "2025-06-07T08:37:55.214Z",
  "reviewedAt": "2025-06-07T08:37:55.214Z",
  "reviewedByUserId": 0
}
```

localhost:5050/swagger/index.html

1/10

184

Execute

Responses

Code	Description	Links
200	Media type application/octet-stream Controls Accept header. Example ValueSchemastring	No links

PUT/api/account/change-request

GET/api/account/oauth/{provider}

GET/api/account/logout

Bookmark

GET/api/bookmarks

POST/api/bookmarks

PATCH/api/bookmarks/{bookmarkId}

DELETE/api/bookmarks/{bookmarkId}

OrganisationalUnit

GET/api/orgunits/all

Project

GET/api/projects/{projectId}

POST	/api/projects/{projectId}	▼
DELETE	/api/projects/{projectId}	▼
GET	/api/projects/{projectId}/edit	▼
POST	/api/projects	▼
GET	/api/projects/statistics	▼
POST	/api/projects/{projectId}/successor	▼
PUT	/api/projects/{projectId}/archive	▼
POST	/api/projects/{projectId}/submit	▼
POST	/api/projects/{projectId}/reject	▼
GET	/api/projects/selectionValues	▼
POST	/api/projects/{projectId}/files	▼
DELETE	/api/projects/{projectId}/files/{fileId}	▼
GET	/api/projects/{projectId}/files/{fileId}/summary	▼
GET	/api/projects/{projectId}/files/summary	▼

Search



GET	/api/search/filterDetails	▼
GET	/api/search/projects	▼
GET	/api/search/projects/mine	▼
GET	/api/search/org-units/current/projects	▼

GET /api/search/projects/validation



MetadataAdmin



GET /api/admin/core-data/metadata



PUT /api/admin/core-data/metadata



POST /api/admin/core-data/metadata



DELETE /api/admin/core-data/metadata/{metadataId}



Metrics



GET /api/admin/metrics



OrganisationalUnitAdmin



GET /api/admin/core-data/organisational-units



PUT /api/admin/core-data/organisational-units



POST /api/admin/core-data/organisational-units



GET /api/admin/core-data/organisational-units/options



DELETE /api/admin/core-data/organisational-units/{orgUnitId}



ProjectTypeAdmin



GET /api/admin/core-data/project-types



PUT /api/admin/core-data/project-types



POST /api/admin/core-data/project-types



DELETE /api/admin/core-data/project-types/{projectTypeId}



RegionAdmin



GET	/api/admin/core-data/regions	▼
PUT	/api/admin/core-data/regions	▼
POST	/api/admin/core-data/regions	▼
DELETE	/api/admin/core-data/regions/{regionId}	▼

SearchTagAdmin



GET	/api/admin/core-data/search-tags	▼
PUT	/api/admin/core-data/search-tags	▼
POST	/api/admin/core-data/search-tags	▼
DELETE	/api/admin/core-data/search-tags/{searchTagId}	▼

SponsorAdmin



GET	/api/admin/core-data/sponsors	▼
PUT	/api/admin/core-data/sponsors	▼
POST	/api/admin/core-data/sponsors	▼
POST	/api/admin/core-data/sponsors/successor	▼
DELETE	/api/admin/core-data/sponsors/{sponsorId}	▼

UserManagement



GET	/api/admin/user-management/users	▼
PATCH	/api/admin/user-management/users/{id}	▼

PATCH	/api/admin/user-management/users/{id}/activate	▼
PATCH	/api/admin/user-management/users/{id}/disable	▼
GET	/api/admin/user-management/user-change-requests	▼
GET	/api/admin/user-management/user-change-requests/{id}	▼
POST	/api/admin/user-management/user-change-requests/accept	▼
POST	/api/admin/user-management/user-change-requests/reject	▼

Schemas



```
UserChangeRequestDto {
  id                [...]
  userId            [...]
  requestedTitle    [...]
  requestedGender*  [...]
  requestedFirstName*  [...]
  requestedLastName*  [...]
  requestedOrgUnitId  [...]
  requestedRole*    RoleEnum  [...]
  requestedIsCommValidator*  [...]
  status            ChangeRequestStatus  [...]
  comment           [...]
  createdAt         [...]
  reviewedAt        [...]
  reviewedByUserId  [...]
}
```

RoleEnum [...]

ChangeRequestStatus [...]

BookmarkDto

ProblemDetails

BookmarkUpdateDto

ProjectDetailsDto

ProjectVisibility

ProjectStatus

RegionDto

ContributorDto

ContributorType

UserDto

OrganisationalUnitDto

OrgUnitType

LinkDto

ProjectFileDto

FileType

StatisticsDto

ProjectEditDto

ProjectTypeDto

SponsorDto

VerificationDto

VerificationStatus

ValidatorType

SearchTagDto

ContributorEditDto

ProjectMetadataDto

EditSelectionValuesDto

MetadataDto

SummaryType

FilterDetailDto

TreeltemWithChildrenDto

TreeltemDto

ProjectSearchResponseDtoOfSearchResponseItemDto

SearchResponseItemDto

ProjectSearchResponseDtoOfSearchResponseEditViewItemDto

SearchResponseEditViewItemDto

MetadataCoreDataDto

AdminMetricsDto

OrganisationalUnitCoreDataDto

ProjectTypeCoreDataDto

SearchTagCoreDataDto

SponsorCoreDataDto

UserSettingsDto

UserAccountStatus

9.2.6 Auszug aus Dokumentation des Codes - Übersicht Service Interfaces

07.06.25, 11:12

My Project: OST.ResearchNavigator.Domain.Interfaces.Services Namespace Reference

OST.ResearchNavigator.Domain.Interfaces.Services Namespace Reference

Classes

interface	IAccessControlService	Defines methods for checking user access to resources. More...
interface	IAccountService	Defines methods for managing user accounts, including user data retrieval and managing change requests for user profile updates. More...
interface	IBookmarkService	Defines methods for managing user bookmarks. Bookmarks are used to save and retrieve specific filters that were set by the user. They are used in the project dashboard. More...
interface	ICoreDataService	Provides methods for managing core data such as sponsors, project types, metadata, organizational units, regions, and search tags. More...
interface	IFileService	Provides methods for managing project files, including upload, deletion, summaries, and access. More...
interface	IMetricsService	Provides methods for retrieving application metrics used by administrators. More...
interface	IOrganisationalUnitService	Defines operations for managing organisational units. More...
interface	IProjectService	Defines operations and business logic for creating, retrieving, updating, and managing projects. More...
interface	ISearchService	Provides search functionalities for projects, including filtering and retrieving filter details. More...
interface	IStatisticsService	Provides functionality to calculate and retrieve statistical data for projects. More...
interface	IUserManagementService	Provides functionality for managing user accounts and processing user change requests. More...
interface	IValidationService	Provides functionality for validating and managing project submissions and rejections, including transitions through technical and communication validations. More...

Generated by  1.9.7

9.2.7 Auszug aus Dokumentation des Codes - IAccountService

07.06.25, 11:14

My Project: OST.ResearchNavigator.Domain.Interfaces.Services.IAccountService Interface Reference

OST.ResearchNavigator.Domain.Interfaces.Services.IAccountService Interface Reference

Defines methods for managing user accounts, including user data retrieval and managing change requests for user profile updates. [More...](#)

Public Member Functions

Task< User? >	GetUserInformation (int id, CancellationToken cancellationToken=default)
	Retrieves detailed information about a user based on their unique identifier.
Task< UserChangeRequest? >	GetPendingUserChangeRequest (int userId, CancellationToken cancellationToken=default)
	Retrieves the pending change request associated with a specific user.
Task< UserChangeRequest? >	CreateUserChangeRequest (UserChangeRequest changeRequest, int userId, CancellationToken cancellationToken=default)
	Creates a new user change request if none is currently pending for the user.
Task< UserChangeRequest? >	UpdateUserChangeRequest (UserChangeRequest changeRequest, int userId, CancellationToken cancellationToken=default)
	Updates an existing pending user change request with new requested values.

Detailed Description

Defines methods for managing user accounts, including user data retrieval and managing change requests for user profile updates.

Member Function Documentation

◆ CreateUserChangeRequest()

Task< **UserChangeRequest?**>

```
OST.ResearchNavigator.Domain.Interfaces.Services.IAccountService.CreateUserChangeRequest ( UserChangeRequest changeRequest,
                                                                                          int userId,
                                                                                          CancellationToken cancellationToken = default
                                                                                          )
```

Creates a new user change request if none is currently pending for the user.

Parameters

- changeRequest** The user change request to create.
- userId** The unique identifier of the user.
- cancellationToken** A token to observe while waiting for the task to complete.

Returns

The newly created UserChangeRequest.

Exceptions

- BadRequestException** Thrown if a pending change request already exists or the user account is disabled.
- NotFoundException** Thrown if the user does not exist.

◆ GetPendingUserChangeRequest()

Task< **UserChangeRequest?**>

```
OST.ResearchNavigator.Domain.Interfaces.Services.IAccountService.GetPendingUserChangeRequest ( int          userId,
                                                                 CancellationToken cancellationToken = default
                                                                 )
```

Retrieves the pending change request associated with a specific user.

Parameters

userId The unique identifier of the user.
cancellationToken A token to observe while waiting for the task to complete.

Returns

A UserChangeRequest if one exists and is pending; otherwise, null.

◆ **GetUserInformation()**Task< **User?**>

```
OST.ResearchNavigator.Domain.Interfaces.Services.IAccountService.GetUserInformation ( int          id,
                                                                 CancellationToken cancellationToken = default
                                                                 )
```

Retrieves detailed information about a user based on their unique identifier.

Parameters

id The unique identifier of the user.
cancellationToken A token to observe while waiting for the task to complete.

Returns

A User entity if found; otherwise, null.

◆ **UpdateUserChangeRequest()**Task< **UserChangeRequest?**>

```
OST.ResearchNavigator.Domain.Interfaces.Services.IAccountService.UpdateUserChangeRequest ( UserChangeRequest changeRequest,
                                                                 int          userId,
                                                                 CancellationToken cancellationToken = default
                                                                 )
```

Updates an existing pending user change request with new requested values.

Parameters

changeRequest The updated change request data.
userId The unique identifier of the user.
cancellationToken A token to observe while waiting for the task to complete.

Returns

The updated UserChangeRequest if successful.

Exceptions

NotFoundException Thrown if no pending change request is found or the user does not exist.
BadRequestException Thrown if the user account is disabled.