

Unterstützung von Transactiv Memory Systems mittels der Verwendung von Tag Clouds

Bachelorarbeit

im Studiengang Wirtschaftsinformatik

Vorgelegt von

Manuel Votapek

Matr.-Nr.: 16561821



am

27.05.2020

an der ZHAW School of Management and Law

Betreut von

Dr. med. Philipp Stalder

Management Summary

Der Wandel zur Wissensgesellschaft hat dazu geführt, dass Wissen zu einem Erfolgsfaktor für Unternehmen geworden ist. Die beiden Kernprozesse Wissensidentifikation und Wissensverteilung des Wissensmanagements, nehmen eine wesentliche Rolle in dem von Wegner konzipierten Transactive Memory System (TMS) ein, bei diesem steht das Meta-Wissen im Zentrum. Ein ausgeprägtes TMS, also ein ausgeprägtes Wissen der einzelnen Mitglieder über das Wissen der anderen, hat einen positiven Einfluss auf die Leistung von Teams. Durch das Wachstum und die Globalisierung von Organisationen ist der Aufbau von TMS in Teams schwieriger geworden, weshalb der Einsatz von Informationssystemen in den Fokus rückt.

In dieser Bachelorarbeit wurde ein Prototyp konzipiert und umgesetzt, der den Austausch der für TMS relevanten Meta-Informationen «who does what» und «who is interested in what» in Teams fördern soll. Dabei wurde die Tag Cloud als Visualisierungsmittel verwendet. Aufgrund der einfachen und zugleich ansprechenden Darstellung, haben Tag Clouds in den letzten Jahren massiv an Popularität gewonnen. Für die Konzipierung wurden mittels Literaturrecherche Beiträge zu Informationssystemen zur Unterstützung von TMS und der Tag Cloud zusammengetragen, diese wurden zur Identifikation und Formulierung von Anforderungen an den Prototyp verwendet. Am fertiggestellten Prototyp wurde dann untersucht, ob dieser sich für die Erfassung und Auswertung der zuvor genannten Meta-Informationen eignet. Weiter wurde untersucht, ob die Tag Cloud von Benutzern akzeptiert und als nützlich wahrgenommen wird. Dazu wurde nebst einer technischen auch eine experimentelle Validierung mit elf Probanden durchgeführt.

Die Evaluation ergab, dass die Anforderungen erfüllt wurden und die Erfassung, Darstellung und Auswertung der Meta-Informationen im Prototyp grundsätzlich möglich sind. Von den Probanden wurde die Tag Cloud für interpretierbar und auswertbar befunden, auch zeigten die Resultate, dass die Tag Cloud ein akzeptiertes Darstellungsmittel ist. Zehn der elf Personen konnten sich vorstellen, den Prototyp oder eine ähnliche Applikation in einem Team einzusetzen. Während spezifische Auswertungen schwierig sind, hat die Tag Cloud Vorteile, wenn Benutzer sich einen Überblick verschaffen oder Trends erkennen wollen.

Es kann davon ausgegangen werden, dass die Tag Cloud sich für die Anwendung in einem Informationssystem zur Unterstützung von TMS eignet, sie sollte jedoch durch geeignete Mittel für die spezifische Datenauswertung ergänzt werden. Zudem ist der Nutzen sowohl von Tag Clouds als auch von Informationssystemen stark kontextabhängig, Tag Clouds eignen sich nicht für jede Art von Auswertung und Informationssysteme nicht für jedes Team. Um zu beantworten, wie Informationssysteme zur Unterstützung von TMS in Teams die Akzeptanz der Benutzer erlangen, müssten weiter Untersuchungen gemacht werden. Die vorliegende Arbeit macht mit dem Hilfsmittel der Tag Cloud einen kreativen Vorschlag zur Gestaltung eines solchen Systems, ob und wie ein solches in der Praxis durch echte Teams akzeptiert wird, wurde aber nicht beantwortet.

Inhaltsverzeichnis

Management Summary	II
Inhaltsverzeichnis	IV
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VI
Abkürzungsverzeichnis	VII
1 Einleitung	1
1.1 Problemstellung	1
1.2 Forschungslücke	2
1.3 Forschungsfragen	2
1.4 Zieldefinition	3
1.5 Abgrenzungen	4
1.6 Vorgehen und Methoden	4
2 Einführung in die Thematik	7
2.1 Grundlagen zu Wissen und Wissensmanagement	7
2.2 Grundlagen zum Transactive Memory System	8
2.3 Grundlagen zur Tag Cloud	10
3 Verwandte Systeme und Arbeiten	12
3.1 Informationssysteme zur Unterstützung von TMS	12
3.2 Nutzen von Tag Clouds	13
3.3 Anwendungsfelder von Tag Clouds	14
3.4 Gestaltung der Tag Cloud	15
3.5 Erkenntnisse	15
4 Anforderungen und Rahmenbedingungen	17
4.1 Rahmenbedingungen	17
4.2 Zieldefinition	17
4.3 Anforderungen	18
5 Technische Dokumentation	21
5.1 Applikationsarchitektur	21
5.1.1 REST-API	21
5.1.2 Single-Page-Application	22
5.1.3 Übersicht	23
5.2 Frontend Design	23

5.2.1	Login/Registrierung	24
5.2.2	Hauptseite	24
5.2.3	Administration	27
5.3	Backend Design	29
5.3.1	Datenmodell	29
5.3.2	API-Design	30
5.4	Backend Umsetzung	33
5.4.1	Datenbank / ORM Sequelize	35
5.4.2	REST-API	37
5.4.3	Testing	40
5.5	Frontend Umsetzung	40
5.5.1	React.js Konzepte	40
5.5.2	Erweiterungen	42
5.5.3	Aufbau Frontend	44
5.5.4	Komponenten	45
5.5.5	Testing	47
5.6	Entwicklungs- und Testumgebung	48
6	Validierung	51
6.1	Technische Validierung	51
6.1.1	Resultate	52
6.2	Validierung mittels User Acceptance Testing	54
6.2.1	Rahmenbedingungen	54
6.2.2	Resultate	58
7	Diskussion	63
7.1	Beurteilung der Resultate	63
7.2	Weiterentwicklung und alternative Lösungsmöglichkeiten	65
8	Schlussfolgerung	68
	Literatur- und Quellenverzeichnis	69
	Anhang	i
A	Testplan	i
B	Aufgabenstellung User-Acceptance-Testing	v
C	Fragebogen User-Acceptance-Testing	vi
D	Resultate User-Acceptance Testing	vii
E	Installations- und Betriebsanleitung	vii
F	Quellcode	xvii
G	Demo Video	xvii
H	PDF-Report	xvii

Abbildungsverzeichnis

Abbildung 1: Struktureller Zusammenhang zwischen Transactive-Memory-Systemen	9
Abbildung 2: Applikationsarchitektur	23
Abbildung 3: Mockup Login/Registrierung	24
Abbildung 4: Mockup Hauptseite 1	25
Abbildung 5: Mockup Hauptseite 2	26
Abbildung 6: Mockup Tag Details	27
Abbildung 7: Mockup Index Editor	28
Abbildung 8: Mockup Benutzerverwaltung	28
Abbildung 9: Mockup Gruppenverwaltung	29
Abbildung 10: ERD-Modell	29
Abbildung 11: OpenAPI Spezifikation	31
Abbildung 12: Frontend Komponenten Übersicht	45
Abbildung 13: Tag Cloud mit Testdaten in der Testumgebung	56
Abbildung 14: Durchschnittlich wahrgenommene Komplexität pro Aufgabe	59
Abbildung 15: Wie viele Tags sollen angezeigt werden?	60

Tabellenverzeichnis

Tabelle 1: Lieferobjekte	3
Tabelle 2: Funktionale Anforderungen	18
Tabelle 3: Anforderungen Erscheinungsbild Tag Cloud	19
Tabelle 4: Zusätzliche Anforderungen	20
Tabelle 5: Beschreibung Endpoints	32
Tabelle 6: Node.Js Erweiterungen	35
Tabelle 7: Diverse React Erweiterungen	43
Tabelle 8: User-Acceptance-Testing Aufgaben Übersicht	56
Tabelle 9: Aufgetretene Probleme pro Aufgabe und Proband	58

Abkürzungsverzeichnis

API	Application Programming Interface
CSV	Comma Seperated Value
JSON	JavaScript Object Notation
REST	Representationale State Transfer
SPA	Single-Page-Application
SQL	Structured Query Language
TMS	Transactive Memory System
UAT	User Acceptance Testing

1 Einleitung

In diesem Kapitel wird auf den wissenschaftlichen Bezug dieser Arbeit eingegangen, dabei wird der Wissenschaftskontext begründet und das weitere Vorgehen skizziert.

1.1 Problemstellung

Um als Organisation im globalen Wettbewerb zu bestehen, bedarf es der Fähigkeit, innovative Produkte und Dienstleistungen in kurzer Zeit zur Verfügung zu stellen. Damit das möglich ist, muss es Unternehmen gelingen, die benötigten Kompetenzen neu aufzubauen und vorhandenes Wissen effektiv und effizient zu transferieren (Nikodemus, 2017, S. 83). Dabei sind Teams ein wichtiger Aspekt der heute dominierenden wissensbasierten Organisationen. Eines der Hauptprobleme ist, dass Wissen in Teams ungleich zwischen Individuen und Artefakten verteilt ist (Boland, Tenkasi & Te'eni, 1994). In diversen Studien zum Thema Transactive Memory System (TMS) wurde die Fähigkeit von Teams, das Wissens sämtlicher Teammitglieder zu nutzen, in Bezug auf die Teamleistung untersucht. TMS setzt sich nach Wegner (1987, S. 186 ff.) aus den drei Komponenten individuelles Gedächtnis, externes Gedächtnis und transaktives Gedächtnis zusammen. Es beschreibt damit das wechselseitige Wissensnetzwerk in Gruppen. Der Fokus liegt dabei darauf, bestehendes Wissen zu finden und zu verwenden, der Kodifizierung kommt dabei eine kleinere Bedeutung zu. Wang, Huang, Davison und Yang (2018) belegten die positive Wirkung von TMS auf die Gruppenleistung. Sie empfehlen Teammitgliedern daher, den Wissensaustausch zwischen den Personen effizienter zu gestalten, indem diese die Kenntnisse aller Mitglieder über das Wissen («who knows what») und die Tätigkeiten («who does what») der einzelnen Personen einsetzen. Weiter legen sie den Managern von Organisation nahe, Informationssysteme zu verwenden, um die Entwicklung von TMS in den Gruppen zu fördern. Jackson (2011) teilt diese Ansicht und begründet den Einsatz von Informationssystemen zur Unterstützung von TMS zusätzlich mit dem Wachstum und die Globalisierung von Organisationen in der heutigen Zeit.

1.2 Forschungslücke

Jackson und Klobas (2008) haben die Charakteristiken von solchen Informationssystemen (IS), skizziert, lassen aber die detaillierte Untersuchung und Konzipierung solcher Systeme für die zukünftige Forschung offen. Die einfachste Form zur Beschreibung der Wissensobjekte einer Gruppe sind Labels (Mohammed & Dumville, 2001). Die Beschreibung von Informationssystemen, die solche Labels speichern, und wie diese visualisiert werden, ist weitgehend unerforscht. Eine Möglichkeit wäre, die Labels anhand von Wissenskarten darzustellen. Sie visualisieren das vorhandene Wissen (Labels) der Mitarbeiter und erleichtern damit den Zugriff bzw. ermöglichen einen grundsätzlichen Überblick (Lehner, 2019, S. 243). Die Tag Cloud ist eine Visualisierungsmethode um Tags zu einer oder mehreren Ressourcen zusammenzufassen und damit den Benutzern die Informationssuche zu erleichtern. Aufgrund der einfachen und zugleich ansprechenden Darstellung von Informationen, haben Tag Clouds in den letzten Jahren massiv an Popularität gewonnen, dabei dienen sie meist als Navigationsunterstützung im Internet. Durch die Charakteristik von TMS IS, welche in der einfachsten Form existierende Informationsobjekte innert einer Organisation oder Gruppe speichern und visualisieren, könnten Tag Clouds ein geeignetes Visualisierungstool sein, welches Benutzern ermöglicht, Informationen zu finden und dadurch TMS unterstützt.

1.3 Forschungsfragen

Kann die Tag Cloud als Visualisierungstool erfolgreich eingesetzt werden, um Teams bei TMS durch ein Informationssystem zu unterstützen?

1. Kann die Tag Cloud verwendet werden, um die Meta-Informationen «who does what» und «who is interested in what» in Teams zu erfassen und visualisieren?
2. Kann die Tag Cloud zur Auswertung dieser Meta-Informationen verwendet werden?
3. Wie kann ein Informationssystem zur Unterstützung von TMS in Teams die Akzeptanz bei Benutzern erlangen?
4. Wird die Tag Cloud bei Benutzern die ein Informationssystem zur Unterstützung von TMS verwendend als nützlich wahrgenommen?

1.4 Zieldefinition

Das Ziel dieser Arbeit ist die Konzipierung, Entwicklung und Validierung eines Prototyps, welcher es Benutzern die in Teams an verschiedenen Projekten arbeiten, ermöglicht, ihre aktuellen Tätigkeiten sowie ihr Interesse an Projekten mittels einer minimalen Eingabe in einer Webapplikation zu erfassen. Die erfassten Daten lassen sich mittels Tag Cloud sichtbar machen und auswerten. Der Prototyp soll TMS in Teams unterstützen und wird technisch sowie mittels User-Acceptance-Testing validiert. Die Resultate des User-Acceptance-Testing sollen Aufschluss über die Nützlichkeit und Akzeptanz eines solchen Informationssystems geben. In der nachfolgenden Tabelle 1 werden die einzelnen Lieferobjekte definiert:

Tabelle 1: Lieferobjekte

Beschreibung

Prototyp einer Webapplikation mit intuitivem Frontend und Visualisierung und Auswertung der erfassten Daten mittels Tag Cloud.

Lieferobjekt: Source Code des Prototyps (Anhang F)

Dokumentation zum Forschungsstand, Positionierung der eigenen Arbeit und der Vorgehensweise, sowie der technischen Dokumentation des Prototyps.

Lieferobjekt: Schriftliche Arbeit

Evaluation des Prototyps sowohl technisch als auch mittels User-Acceptance-Testing mit einer Testgruppe von mindestens 10 Personen.

Lieferobjekt: Dokumentation der Testresultate und der Ergebnisse aus dem User-Acceptance-Testing. (Anhang D)

Kurzes Video welches die Benutzung des Prototyps demonstriert.

Lieferobjekt: Video (Anhang G)

Installations- und Betriebsanleitung zum entwickelten Prototyp.

Lieferobjekt: Installations- und Betriebsanleitung (Anhang E)

1.5 Abgrenzungen

Im Rahmen dieser Arbeit wird ein Prototyp einer Webapplikation für die Erfassung von Meta-Informationen zu aktuellen Tätigkeiten und Interessen erstellt. Die Daten werden mittels Tag Cloud visualisiert. Das erstellte Artefakt ist dementsprechend nicht ein produktives System, das praktisch Anwendung findet. Es soll das Konzept TMS und die Visualisierungsmethode Tag Cloud kombiniert und validiert werden. Die Entwicklung von Algorithmen zur Generierung von Tag Clouds ist nicht Teil dieser Arbeit.

1.6 Vorgehen und Methoden

Um die Forschungsfragen zu beantworten, werden die folgenden Arbeitsschritte durchgeführt. Die Arbeit wird in drei Phasen gegliedert, wobei sich die erste mit der bestehenden Literatur befasst und die zweite mit der Entwicklung und Validierung des Prototyps. In der letzten Phase werden die Resultate dieser Arbeit diskutiert. Dabei wird nach dem konstruktionsorientierten Paradigma zur Erstellung und Bewertung von IT-Artefakten vorgegangen. Das von Hevner, March, Park und Ram (2004) entwickelte Framework liefert ein Regelwerk, das qualitativ hochwertige Forschung im Bereich von IS sicherstellt.

Für die drei Phasen wurden folgende Arbeitspakete (AP) identifiziert:

1. Literatur

AP 1.1 – Einführung in die Thematik

Einführung in die Forschungsgebiete TMS, TMS Informationssysteme und Tag Cloud mittels Literaturrecherche sowie Einordnung dieser Arbeit in Bezug auf den aktuellen Forschungsstand. Das Ergebnis dieses Arbeitspakets befindet sich in Kapitel 2.

AP 1.2 – Verwandte Systeme und Arbeiten

Ebenfalls mittels Literaturrecherche werden bereits durchgeführte Projekte und verwandte Systeme betrachtet. Darauf werden die Ergebnisse mit den Forschungsthemen dieser Arbeit verglichen. Die verwandten Systeme und Arbeiten werden in Kapitel 3 besprochen.

2. Prototyp

AP 2.1 – Anforderungen an Prototyp

Anhand der aus der Literaturrecherche erlangten Erkenntnisse sowie der Zielsetzung dieser Forschungsarbeit, werden die Anforderungen des Prototyps formuliert. Diese werden in Kapitel 4 beschrieben.

Die Arbeitspakete 2.2 bis 2.6 befassen sich mit der technischen Umsetzung des Prototyps, diese wird in Kapitel 5 beschrieben.

AP 2.2 – System Architektur

Die ermittelten Anforderungen werden genutzt, um die System Architektur zu definieren. Die dazugehörigen Konzepte werden beschrieben.

AP 2.3 Frontend Design

Es werden Mockups für die verschiedenen Benutzeroberflächen der Applikation erstellt. Diese sollen die zuvor definierten Anforderungen erfüllen.

AP 2.4 – Backend Design

Das Datenmodell für das Backend sowie die API-Endpoints werden anhand der Anforderungen und dem Frontend Design definiert.

AP 2.5 – Backend Umsetzung

Der Data-Layer und die API-Endpoints werden implementiert und beschrieben.

AP 2.6 – Frontend Umsetzung

Das Frontend Design wird implementiert, die API-Endpoints werden verwendet um Ressourcen aus dem Backend abzurufen. Wichtige Erkenntnisse und Lösungen im Sourcecode werden beschrieben.

AP 2.7 – Validierung Prototyp

Der fertige Prototyp wird in zwei Phasen validiert. Zuerst werden die Anforderungen als Testfälle übernommen, dann werden diese verwendet um den Prototyp zu testen. In einem

weiteren Schritt werden die Szenarien für das User-Acceptance-Testing definiert, ein Feedback-Fragebogen erstellt und letztendlich das Testing mit mindestens 10 Probanden durchgeführt. Beide Phasen sind auszuwerten und zu dokumentieren. Die Durchführung und Resultate der Validierung werden in Kapitel 6 beschrieben.

AP 2.8 – Anleitung Prototyp

Es ist eine schriftliche Installations- und Betriebsanleitung zu erstellen. Weiter wird ein Video erstellt, dass die Benutzung des Prototyps demonstriert.

3. Diskussion

AP-3.1 – Ergebnisse, Erkenntnisse und Fazit

Die in dieser Forschungsarbeit erstellten Ergebnisse werden reflektiert. Die eigene Arbeit wird kritisch besprochen und Aktivitäten für die zukünftige Forschung werden erfasst. Die Kapitel 7 und 8 befassen sich mit diesem Arbeitspaket.

2 Einführung in die Thematik

In diesem Kapitel werden die Grundlagen der Themengebiete, in welchen sich diese Arbeit befindet, erläutert. Dabei werden die Konzepte des Wissens und des Wissensmanagement, Transactive Memory System und der Tag Cloud vorgestellt.

2.1 Grundlagen zu Wissen und Wissensmanagement

Der Wandel von der Industrie- zur Wissensgesellschaft prägte die letzten Jahre. Wissen war bereits früher von Bedeutung, jedoch wird Wissen in der Wissensgesellschaft zu dem Produktionsfaktor, der andere Faktoren wie Land, Kapital und Arbeit dominiert (Willke, 2007, S. 21). Wissen und andere geistige Potenziale sind zu entscheidenden Faktoren für den Unternehmenserfolg geworden (Lehner, 2019, S. 37). Bevor die Aufgaben des Wissensmanagement beleuchtet werden können, gilt es ein Verständnis für den Wissensbegriff zu schaffen. Probst, Raub und Romhardt (2012, S. 23) definieren Wissen wie folgt:

«Wissen bezeichnet die Gesamtheit der Kenntnisse und Fähigkeiten, die Individuen zur Lösung von Problemen einsetzen. Dies umfasst sowohl theoretische Erkenntnisse als auch praktische Alltagsregeln und Handlungsanweisungen. Wissen stützt sich auf Daten und Informationen, ist im Gegensatz zu diesen jedoch immer an Personen gebunden. Es wird von Individuen konstruiert und repräsentiert deren Erwartungen über Ursache-Wirkungszusammenhänge.»

Die Definition beinhaltet bereits eine erste Abgrenzung des Wissensbegriffs von Daten und Informationen. Diese bilden zwar die Basis von Wissen, sind mit diesem jedoch nicht gleichzusetzen. Daraus folgt, dass Daten-, Informations- und Wissensmanagement stets zusammenspielen (Propst et al., 2012, S. 24). An dieser Stelle wird auf eine umfassende Auflistung der verschiedenen Wissenskategorien und Merkmale verzichtet. Es sei lediglich die in dieser Arbeit dominierende Wissensart vorgestellt: das sogenannte Meta-Wissen. Dabei handelt es sich um Wissen über Wissen. Dies schliesst auch der Aufbewahrungsort des Wissens ein. Ein Merkmal von Meta-Wissen ist, dass es nicht ohne die Beteiligung der Individuen in eine Meta-Wissensbasis überführt werden kann (Lehner, 2019, S. 84-85). Meta-Wissen über die Gruppenmitglieder ist ein erheblicher Bestandteil des Transactive Memory Systems welches im Kapitel 2.2 vorgestellt wird. Es beinhaltet

das Wissen über das Wissen und die Perspektiven der anderen Gruppenmitglieder in Bezug auf die Arbeits-, Kommunikations- und Koordinationsprozesse in der Gruppe (S. 93).

Dadurch, dass Wissen zu einem entscheidenden Faktor für Unternehmen wurde, ist es auch zur Managementaufgabe geworden. Da in dieser Arbeit primär auf das organisationale Wissensmanagement Bezug genommen wird, folgt die Definition der organisationalen Wissensbasis. Diese setzt sich nach Propst et al. (2012, S. 23) aus individuellen und kollektiven Wissensbeständen zusammen. Darüber hinaus umfasst sie die Daten und Informationsbestände, auf welchen individuelles und organisationales Wissen aufbaut. Die systematische und methodisch fundierte Wahrnehmung der Aufgaben, die damit zusammenhängen, erfolgt durch das Wissensmanagement (Lehner, 2019, S. 37). Propst et al. (2012, S. 34) nennen sechs Kernprozesse des Wissensmanagements, nämlich Wissensidentifikation, Wissenserwerb, Wissensentwicklung, Wissens(ver)teilung, Wissensnutzung und Wissensbewahrung. Da alle Kernprozesse enge Verbindungen aufweisen, empfehlen sie, dass von einer isolierten Optimierung in einzelnen Bereichen ohne Berücksichtigung der Auswirkungen auf andere Bereiche abgesehen wird. Interventionen können sowohl auf der individuellen und Gruppenebene ansetzen, als auch direkt auf die organisationale Ebene abzielen (Propst et al., 2012, S. 34).

Ursprünglich gab es zwei Grundausrichtungen des Wissensmanagements: es sind dies der humanorientierte und der technologische Ansatz (vgl. Schüppel 1996, S. 187-190). Der humanorientierte Ansatz sieht die Person als zentraler Wissensträger. Der technologische Ansatz geht vom Einsatz innovativer Technologien aus, bei dem das Wissen etwa erfasst und gespeichert wird (Lehner, 2019, S. 49). Lehner (2019, S. 49) sieht jedoch in den letzten Jahren verstärkte Versuche, die beiden Grundausrichtungen in einem ganzheitlichen Konzept zu vereinen. Er selbst propagiert ebenfalls einen integrativen Ansatz.

2.2 Grundlagen zum Transactive Memory System

Das Transactive Memory System (TMS) ist ein von Wegner (1987) konzipiertes System, dem Untersuchungen zum Gruppengedächtnis aus der Psychologie zugrunde liegen. Es besteht es aus den drei Komponenten externes Gedächtnis, individuelles Gedächtnis und transaktives Gedächtnis. Die Abbildung 1 (Lehner, 2019, S. 160) zeigt, dass ein Unternehmen eine Vielzahl von TMS umfasst.

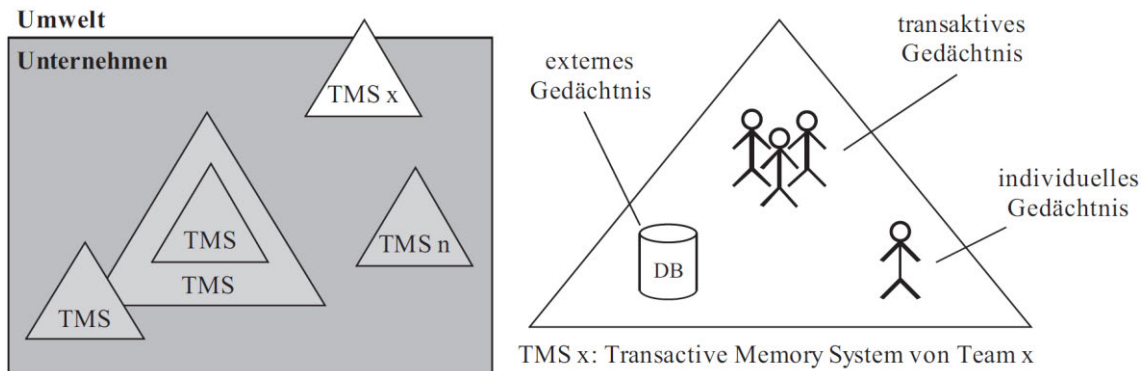


Abbildung 1: Struktureller Zusammenhang zwischen Transactive-Memory-Systemen

Jedes TMS besteht aus mehreren individuellen Gedächtnissen in Kombination mit der Kommunikation, die zwischen den Individuen stattfindet (Wegner, 1987). Wegner nutzt zur Charakterisierung des individuellen Gedächtnisses, die bekannte phasenorientierte Sicht auf die menschliche Informationsverarbeitung (Kodierung, Speicherung, Wiederauffinden) (Lehner, 2019, S. 161). Zudem umfasst das Gedächtnis auch ein Meta-Gedächtnis, welches es ermöglicht, Wissen überhaupt abzurufen – es speichert Wissen über das gespeicherte Wissen (Wegner, 1987). Im externen Gedächtnis werden Informationen ausserhalb des individuellen Gedächtnisses gespeichert, Wegner nennt etwa den Terminkalender als Beispiel. Damit die Informationen aus dem externen Gedächtnis abrufbar sind, muss im individuellen Gedächtnis zumindest gespeichert werden, wo sich diese befindet. Wegner geht davon aus, dass dieses Meta-Wissen über die Lokation von externen Informationen einen erheblichen Teil unseres individuellen Gedächtnisses belegt (Wegner, 1987). Die dritte Komponente stellte das transaktive Gedächtnis dar, dabei werden Personen wechselseitig als externes Gedächtnis herangezogen. Damit ist gemeint, dass wenn man weiss, dass eine andere Person etwas weiss, man über Kommunikation (Transaktion) auf deren individuelles Gedächtnis zugreifen kann. Damit entsteht ein grösseres und komplexeres Wissenssystem, als jedes einzelne individuelle Gedächtnis für sich. Jedoch kennt jede Person dieses Wissenssystem nur aus ihrer eigenen Perspektive, hat also ihr eigenes Meta-Wissen über das Wissen im System. Somit trägt jede Person mit ihrem individuellen Gedächtnis und dem darin enthaltenen Meta-Wissen zur Wiederauffindung von Informationen in der Gruppe bei.

Zahlreiche Studien belegen, dass ein ausgeprägtes TMS in Teams deren Leistung erhöhen kann (vgl. Dayan & Basarir, 2009; Dayan & Benedetto, 2009; Hsu, Shih, Chiang & Liu, 2012; Zhong et al., 2012). Lewis (2003) charakterisiert TMS wie folgt: «TMSs are

thought to facilitate quick and coordinated access to deep, specialized knowledge, so that a greater amount of task-relevant expertise can efficiently be brought to bear on team tasks.» Er befand ausserdem, dass TMS bei Teams, die für die Erledigung ihrer Aufgaben ihr Wissen bündeln und koordinieren müssen, die grösste Wirkung hat (Lewis, 2003). TMS ist daher besonders für Teams die wissensintensiven Arbeiten nachgehen, von hoher Bedeutung.

2.3 Grundlagen zur Tag Cloud

Die Tag Cloud ist eine Visualisierungsmethode, um Tags zu einer oder mehreren Ressourcen zusammenzufassen und damit den Benutzern die Informationssuche zu erleichtern. Im Gegensatz zu Word Clouds, bei denen die Häufigkeit von Wörtern in Texten visualisiert wird, werden Tags von Personen erstellt und referenzieren auf eine Ressource (Trattner, Helic & Strohmaier, 2014). Die Visualisierung basiert dabei auf dem Tag-Wort (Schlagwort) und einer weiteren Variablen, wie etwa Wichtigkeit oder Popularität, die durch die visuelle Repräsentation (z.B. Schriftgrösse, Farbe) abgebildet wird (Bateman, Nacenta & Gutwin, 2008). Typischerweise werden in Tag Clouds die N wichtigsten Tags angezeigt (Trattner et al., 2014). Ihren Bekanntheitsgrad erlangte die Tag Cloud, unter anderem durch die Foto-Plattform Flickr, diese erlaubte Benutzern ihre hochgeladenen Fotos mit Meta-Daten in der Form von Tags zu versehen, welche dann in Tag Clouds zur Navigation und der Suche von Fotos Verwendung fanden (Trattner et al., 2014). Die meist verbreitete Form der Tag Cloud im Internet verwendet die Popularität (wie oft ein Tag verwendet wurde) um die Schriftgrösse der Tags festzulegen (Bateman et al., 2008). Aufgrund der einfachen und zugleich ansprechenden Visualisierung von Informationen haben Tag Clouds in den letzten Jahren massiv an Popularität gewonnen. Rivadeneira, Gruen, Muller und Millen (2007) benennen vier Aufgaben, die Tag Clouds unterstützen:

1. **Suchen:** Einen spezifischen oder verwandten Begriff lokalisieren bzw. die Abwesenheit dessen feststellen. Die Suche dient als Mittel, um an detaillierte Information zu einem festgelegten Thema zu gelangen.
2. **Browsen:** Die Tag Cloud überfliegen, ohne ein spezifisches Ziel zu verfolgen. Dies führt dazu, dass Benutzer diese Tags untersuchen, die sie interessieren.

3. **Eindrucksbildung:** Sich einen Eindruck über den Inhalt der Datengrundlage machen. Der Eindruck umfasst dabei sowohl die herausragenden wie auch die weniger populären Begriffe. Etwa einen Überblick über den Inhalt eines Textes erlangen.
4. **Wiedererkennen:** Anhand einer Tag Cloud erkennen, welches Objekt/Thema damit beschrieben wird. Als Beispiel nennen sie, das Erkennen einer Person anhand einer Tag Cloud aufgrund Meta-Daten zu der Person.

Im Gegensatz zu anderen Visualisierungsmitteln kommt die Tag Cloud ohne zusätzliche Komponenten aus (z.B. die Balken im Balkendiagramm) (Bateman et al., 2008). Um die auszuwertende Variable (z.B. Popularität) zu visualisieren, wird die visuelle Repräsentation des Wortes angepasst. Dabei können Parameter in den zwei Kategorien Text Gestaltung und Wort Position verwendet werden. Text Gestaltung umfasst die Attribute der Schrift wie Grösse, Farbe oder fett/normal/kursiv Schreibweise. Die Position der einzelnen Tags, deren Sortierung oder Gruppierung, gehören zur zweiten Kategorie (Rivadeneira et al., 2007).

3 Verwandte Systeme und Arbeiten

In diesem Kapitel werden Systeme und Arbeiten besprochen, die in einer Beziehung zu der vorliegenden Arbeit stehen. Die Erkenntnisse und Konzepte haben einen erheblichen Einfluss auf die in Kapitel 4 formulierten Anforderungen.

3.1 Informationssysteme zur Unterstützung von TMS

Wang et al. (2018) belegten die positive Wirkung von TMS auf die Gruppenleistung. Sie empfehlen Teammitgliedern daher, den Wissensaustausch zwischen den Personen effizienter zu gestalten, indem diese die Kenntnisse aller Mitglieder über das Wissen («who knows what») und die Tätigkeiten («who does what») der einzelnen Personen einsetzen. Um dies umzusetzen, schlagen sie vor, Informationssysteme zu verwenden, um die Entwicklung von TMS in den Gruppen zu fördern. Auch Yuan, Fulk und Monge (2007) glauben, dass Informationssysteme Mitarbeiter dabei unterstützen können, die benötigte Expertise im Unternehmen zu finden und damit das Meta-Wissen betreffend «Wer weiss was» zu fördern. Ein weiterer Grund, der für die Verwendung von Informationstechnologie spricht, ist das Wachstum und die Globalisierung von Organisationen (Jackson, 2011). Moreland und Myaskovsky (2000) stellten in ihrem Versuch ausserdem fest, dass Teams, die nur in schriftlicher Form über die Expertise ihrer Kollegen informiert waren, gleich gute Ergebnisse erzielten, wie Teams, die sich persönlich austauschen konnten.

Jackson und Klobas (2008) haben die Charakteristiken von TMS Informationssystemen, skizziert. Dabei gehen sie von vier Basis-Komponenten aus.

1. Ein einfach zugängliches **Verzeichnis** bzw. mehrere Verzeichnisse. Die Verzeichnisse bestehen dabei aus einem Speicherort, an dem Labels und zugehörige Meta-Informationen, die es ermöglichen Wissen zu finden, gespeichert werden. Als Beispiele für Einträge im Verzeichnis, nennen sie Angaben zu Mitarbeitern und deren Wissensbereiche, Projekt-Reports oder Lebensläufe. Zudem soll koordiniertes Wissen, etwa in der Form von Dokumentensystemen, eingebunden werden.
2. Eine Methode um das Verzeichnis zu **verwalten**, so dass dieses den aktuellen Zustand im Unternehmen widerspiegelt. Es muss einen Prozess geben, um Erfahrungen und Wissensbereiche aktuell zu halten.

3. Funktionen um Information aus dem Verzeichnis **abrufen** und **suchen** zu können.
4. Eine Methode, die es erlaubt, die **Verantwortung für Wissensbereiche** zuzuweisen. Für die Zuweisung sollen Meta-Informationen über Rolle und Interessen der Mitarbeiter verwendet werden.

Jackson und Klobas (2008) machen damit einen sehr integrativen Vorschlag zur Gestaltung von TMS Informationssystemen. Dabei haben sie eine umfassende Liste mit Anforderungen aus der bestehenden TMS Literatur und ihren bisherigen Forschungsergebnissen zusammengetragen. Das von ihnen vorgeschlagene Informationssystem kombiniert dabei verschiedenste Ressourcen und Meta-Informationen.

In der Literatur werden jedoch auch weniger komplexe Informationssysteme genannt, die in Verbindung mit TMS stehen. Alavi und Tiwana (2002) nennen etwa Wissenskarten oder Gelbe Seiten Informationssysteme, die Auskunft über die Erfahrung und Wissensgebiete der Mitarbeiter geben. Während solche Systeme dem Vorschlag eines TMS Informationssystems von Jackson und Klobas (2008) wohl nicht entsprechen, ist ihr potenzieller Einfluss auf TMS trotzdem nicht abzuerkennen. TMS baut darauf auf, dass die Mitglieder wissen wer in ihrem System (Team/Organisation) was weiss. Dieses Wissen wird über die Kommunikation im Team aufgebaut. Somit können sämtliche Informationssysteme, die diese Kommunikation fördern oder unterstützen, potenziell einen positiven Einfluss auf die TMS Entwicklung haben. Dazu zählen etwa Groupware oder Social Software. Groupware steht im Zusammenhang mit dem Konzept des Computer Supported Cooperate Work (CSCW) (Lehner, 2019, S. 300). Dabei geht es um Informationssysteme, die die Kooperation und Koordination von Gruppen und Teams fördern. Lehner (S.301) nennt Kommunikationssysteme (z.B. Skype oder Zoom), Kollaborationssysteme (z.B. MS Teams) und Koordinationssysteme (z.B. Workflowmanagementsysteme wie JIRA) als Beispiele. Social Software steht für Web-Anwendungen, die Personen beim Informationsaustausch, dem Beziehungsaufbau und der Kommunikation in einem sozialen Kontext unterstützen (S. 307).

3.2 Nutzen von Tag Clouds

Sinclair und Cardew-Hall (2008) haben untersucht, für welche Art von Aufgaben Tag Clouds nützlich sind. Sie haben dafür Benutzern für verschiedene Aufgaben jeweils die

Wahl zwischen der Tag Cloud und der Eingabe eines Suchbegriffs gelassen. Dabei stellen sie fest, dass die Stärken der Tag Cloud beim Browsen ohne spezifisches Ziel liegen. Dies begründen sie dadurch, dass Tag Clouds besser geeignet sind, um sich einen Überblick zu verschaffen, als eine spezifische Charakterisierung des Inhalts zu erlangen. Zudem reduziert die Tag Cloud den Aufwand, um Informationen abzurufen, zumal dafür bereits ein Klick genügt, und der Benutzer dafür keine Sucheingabe vornehmen muss. Die Probanden haben ausserdem die Tag Cloud gehäuft dafür verwendet, um sich einen Überblick über das Thema und verfügbare Inhalte zu verschaffen. Dies diente ihnen dann als Einstiegspunkt für spezifische Suchen mittels Suchbegriffen. Sinclair und Cardew-Hall (2008) sehen daher das Potential der Tag Cloud unter anderem für einen zweiphasigen Suchprozess, bei dem die Tag Cloud den Benutzer an den verfügbaren Inhalt heranführt, und dieser dann in einem zweiten Schritt eine fokussierte Suche vornehmen kann. Als letzten Vorteil nennen sie, dass die Suche mittels Tag Cloud von den Benutzern weniger mentale Leistung erfordert, als das Definieren des Suchbegriffs. Damit könnte sich die Tag Cloud besonders für Benutzer eignen, die Informationen in einer Fremdsprache suchen (Sinclair & Cardew-Hall, 2008).

Während die Tag Cloud in gewissen Bereichen durchaus über Stärken verfügt, hat sie auch gewisse Limitationen. Dazu zählt unter anderem, dass sie sich nur im beschränkten Masse dazu eignet spezifische Informationen zu finden (Sinclair & Cardew-Hall, 2008). Zudem ist die Anzahl Tags die in einer Tag Cloud angezeigt werden können beschränkt, wodurch wenig populäre Tags allenfalls gar nicht angezeigt und somit nicht gefunden werden können (Sinclair & Cardew-Hall, 2008).

Bateman et al. (2008) sehen für die Tag Cloud eine Nische, die von vielen Suchmaschinen übersehen wird, nämlich da, wo Benutzer die Suchbegriffe nicht kennen oder sie ihnen nicht einfallen. Sinclair und Cardew-Hall (2008) kommen zum Schluss, dass Tag Clouds keinen Ersatz für die Begriffssuche darstellen, jedoch zur Unterstützung der Benutzer verwendet werden sollen. Inwiefern sich die Tag Cloud eignet, muss daher wohl von Anwendungsfall zu Anwendungsfall untersucht und abgewogen werden.

3.3 Anwendungsfelder von Tag Clouds

Die Tag Cloud wird für unterschiedliche Anwendungszwecke verwendet, Khusro, Jabeen und Khan (2018) nennen acht verschiedene Anwendungsfelder. Diese reichen von Social

Media Applikationen, bei denen die von Benutzern erstellten Tags zu Beiträgen und Fotos in der Tag Cloud visualisiert werden, bis hin zur Visualisierung von Text- oder Dokumentinhalten. Gottron (2009) hat in seiner Arbeit eine Methode entwickelt, die Tags aus Web-Dokumenten extrahiert und in einer Tag Cloud visualisiert. Die Testgruppe befand die generierten Tag Clouds seien zwar anfangs ungewohnt, sie könnten aber nach einiger Zeit durchaus hilfreich sein, um schneller einen Überblick über den Inhalt zu erhalten. De Spindler et al. (2011) haben in ihrer Arbeit ein Framework vorgestellt, das Browsen und Abfragen von Datensammlungen mittels mehreren synchronisierten Tag Clouds ermöglicht, womit auch komplexe Abfragen getätigt werden können.

3.4 Gestaltung der Tag Cloud

Bateman et al. (2008) untersuchten die Variablen Schriftgrösse, Farbe, Wortlänge und Positionierung in Relation zur Aufmerksamkeit der Benutzer, und stellen dabei fest, dass die Schriftgrösse und Fett-Schrift den grössten Einfluss haben. Deutsch, Schrammel und Tscheligi (2011) haben verschiedene Layouts der Tag Cloud verglichen und empfehlen, Clustering Algorithmen zu verwenden, da die Anzahl Tags, die ein Nutzer noch effizient wahrnimmt, beschränkt ist. Diese Erkenntnisse teilen Khusro et al. (2018). Sie empfehlen Tag Clouds mehrschichtig und dynamisch zu gestalten und dabei pro Schicht eine begrenzte Anzahl Tags anzuzeigen. Benutzer könnten dann von einer ersten oberflächlichen Tag Cloud, durch anklicken eines Tags in die darunter liegende Schicht navigieren, um spezifischere Tag Clouds zu betrachten.

3.5 Erkenntnisse

Die Recherche im Forschungsfeld dieser Arbeit hat ergeben, dass bereits Informationssysteme, die TMS unterstützen, existieren. Ein TMS Informationssystem gemäss dem Vorschlag von Jackson und Klobas scheint bis heute nicht umgesetzt. Die Forschungsarbeiten betreffend Tag Cloud fokussieren sich primär auf die Evaluierung ihres allgemeinen Nutzens und der Weiterentwicklung bzw. Optimierung. Tag Clouds werden bereits in verschiedenen Anwendungsfeldern eingesetzt, jedoch konnten keine existierenden Systeme oder Forschungsarbeiten gefunden werden, bei denen die Tag Cloud in den Fel-

dern TMS oder Wissensmanagement angewandt oder untersucht wird. Überraschenderweise finden sich nur wenige Case Studies, die die Tag Cloud als Lösungsansatz für ein spezifisches Problem untersuchen.

4 Anforderungen und Rahmenbedingungen

In diesem Kapitel werden die Anforderungen für die Entwicklung des Prototyps festgehalten. Dabei handelt es sich um eine Kombination der Vorgaben der fachlichen Betreuung dieser Bachelorarbeit und den erlangten Erkenntnissen aus dem letzten Kapitel. Zuerst werden Rahmenbedingungen für die Entwicklung des Prototyps vorgestellt, danach werden die Ziele und Anforderungen formuliert.

4.1 Rahmenbedingungen

Es wurden lediglich zwei Rahmendbedingungen vorgegeben, welche die Entwicklung des Prototyps beeinflussen. Als erste sei genannt, dass als Endprodukt eine Webapplikation (Website) erwartet wird. Weiter ist gefordert, dass der Prototyp möglichst einfach in Betrieb genommen werden kann. Es sollte daher auf Cloud-Provider (z.B. Amazon-Web-Services, Google Cloud) spezifische Implementationen möglichst verzichtet werden.

4.2 Zieldefinition

Die Ziele ergeben sich aus den Erkenntnissen aus Kapitel 3, den zuvor genannten Rahmenbedingungen und den Vorgaben der fachlichen Betreuung der Zürcher Fachhochschule dieser Arbeit. Die folgenden Ziele wurden erkannt und formuliert:

1. Der Prototyp soll es Benutzern, die in Teams an mehreren Projekten arbeiten, ermöglichen, ihre aktuellen Tätigkeiten sowie Interessen in einer Webapplikation zu erfassen. Die Erfassung soll in der Form von Tags geschehen, welche Meta-Wissen im Sinne des TMS darstellen («who does what» und «who is interested in what»).
2. Die erfassten Daten werden im Prototyp mittels Tag Cloud visualisiert und ausgewertet. Jeder Benutzer soll über eine persönliche Tag Cloud mit seinen erfassten Tags verfügen. Zudem soll auch eine Tag Cloud für sämtliche Benutzer verfügbar sein. Es sollen Auswertungen pro Tag möglich sein, die zeigen, welche Benutzer ein Tag wie oft hinzugefügt haben.

3. Es soll ein Tag Index verfügbar sein, von dem Benutzer bereits existierende Tags auswählen und posten können. Sie haben jedoch auch die Möglichkeit, neue Tags hinzuzufügen. Es soll möglich sein, Tags im Index zu bearbeiten, löschen oder hinzuzufügen.

Die zwei grundlegenden Funktionen sind die Erfassung der Interessen und Tätigkeiten als Tags und die Auswertung der Tags mittels Tag Clouds. Beide Funktionen sollen mittels sinnvoll und einfach gestaltetem Frontend intuitiv umgesetzt werden. Zusätzlich sollen administrative Funktionen, um den Tag Index zu bearbeiten und pflegen umgesetzt werden. Da die Webapplikation die Auswertung sowohl für einen einzelnen Benutzer als auch für sämtliche Benutzer ermöglicht, muss sie es Benutzern ermöglichen, sich zu registrieren und einzuloggen (Multi-User-Applikation).

4.3 Anforderungen

Anhand der Erkenntnisse der letzten Kapitel wurden funktionale Anforderungen an den Prototyp erhoben, welche in der Tabelle 2 dargestellt werden.

Tabelle 2: Funktionale Anforderungen

ID	Bezeichnung	Beschreibung
A1	Registrierung	Neue Benutzer können sich für die Applikation registrieren und damit neue Anmeldedaten erstellen.
A2	Einloggen	Benutzer können sich mit ihren Anmeldedaten einloggen.
A3	Ausloggen	Eingeloggte Benutzer können sich wieder ausloggen.
A4	Tags erfassen	Benutzer können Tags erfassen, dabei können sie zwischen den zwei Tag-Typen «Arbeite an» und «Interessiert an» auswählen. Dabei kann entweder ein bereits existierendes Tag aus dem Tag-Index gewählt, oder ein neues erstellt werden.

A5	Persönliche Tag Cloud anzeigen	Benutzer können eine Tag Cloud mit den von ihnen erfassten Tags anzeigen.
A6	Tag Cloud aller Benutzer anzeigen	Benutzer können eine Tag Cloud mit den von sämtlichen Benutzern erfassten Tags anzeigen.
A7	Ranking von Tags	Benutzer sollen ein Ranking sämtlicher erfassten Tags, die sie erfasst haben, erhalten. Dieselbe Auswertung soll für alle Benutzer möglich sein.
A8	Auswertung pro Tag	Benutzer haben die Möglichkeit Auswertungen zu einem gewählten Tag zu erhalten. Dabei erhalten sie Meta-Informationen zum Tag sowie welche Benutzer dasselbe Tag ebenfalls erfasst haben.
A9	Tag-Index bearbeiten	Es soll möglich sein, Tags aus dem Tag-Index zu entfernen, neue hinzuzufügen und Tags zu bearbeiten.

Im Kapitel 3.4 wurden Beiträge besprochen, die sich mit der visuellen Aspekten der Tag Cloud befassen, die von diversen Autoren erlangten Erkenntnisse sollen in die Entwicklung des Prototyps einfließen. In der Tabelle 3 werden Anforderungen an das Erscheinungsbild der Tag Cloud im Prototyp formuliert:

Tabelle 3: Anforderungen Erscheinungsbild Tag Cloud

ID	Bezeichnung	Beschreibung
T1	Einfärbung der Tags	Um populäre Tags (solche mit vielen Posts) hervorzuheben, sollen die Tags, ihrer Popularität im Verhältnis zu sämtlichen angezeigten Tags entsprechend, eingefärbt werden.
T2	Anzahl Tags einschränken	Die Anzahl Tags, die in der Tag Cloud angezeigt werden, soll vom Benutzer begrenzt sein.

T3	Stabile Tag Cloud	Beim hinzufügen / posten von Tags sollen sich die Positionen der Tags so wenig wie möglich verändern, auf keinen Fall dürfen dabei sämtliche Tags neu angeordnet werden.
----	-------------------	--

Während der Konzipierung des Systems und der Entwicklung des Prototyps haben sich weitere Anforderungen ergeben, welche in der Tabelle 4 ersichtlich sind.

Tabelle 4: Zusätzliche Anforderungen

ID	Bezeichnung	Beschreibung
B1	Autorisierung	Benutzer können die Rolle 'admin' oder 'user' haben. Benutzer mit der Rolle 'admin' sollen die Möglichkeit haben, die in Anforderung A9 formulierte Funktionalität zu nutzen.
B2	Gruppen	Benutzer sollen zu einer Gruppe gehören. Zusätzlich zu den Auswertungen, die in der Anforderung T3 formuliert wurden, können sie Auswertungen über die Tags ihrer und anderer Gruppen tätigen.
B3	Gruppenverwaltung	Benutzer mit der Rolle 'admin' sollen die Möglichkeit haben, neue Gruppen zu erstellen, bearbeiten und löschen.
B4	Benutzerverwaltung	Benutzer mit der Rolle 'admin' sollen die Möglichkeit haben, andere Benutzer einer Gruppe hinzuzufügen und ihre Rolle zu ändern.
B5	Reports	Es soll möglich sein, Reports aus der Applikation zu exportieren, so dass aus Auswertungen gewonnene Erkenntnisse auch ausserhalb der Applikation nutzbar gemacht werden.

5 Technische Dokumentation

In diesem Kapitel wird die Entwicklung des Prototyps beschrieben. Dabei wird zuerst die Applikationsarchitektur aufgezeigt und in der Folge technische Lösungsansätze der wichtigsten Funktionen beschrieben.

5.1 Applikationsarchitektur

Aufgrund der Rahmenbedingungen und den Anforderungen wurde eine passende Applikationsarchitektur gewählt und angewandt. Da es sich beim Prototyp um eine Webapplikation handelt, wurde eine Client-Server Architektur angewandt, die dem heutigen Standard entspricht. Die dabei verwendeten Konzepte zu REST-API und Single-Page-Applications werden nun vorgestellt.

5.1.1 REST-API

REST steht für Representational State Transfer und beschreibt einen Architektur-Stil und ist daher nicht mit einer Guideline oder einem Standard zu verwechseln (Doglio, 2015, S. 1). REST basiert auf dem HTTP Protokoll, und wurde entwickelt, um den Datenaustausch zwischen Client und Server zu vereinfachen (S.1). REST impliziert eine Client-Server Architektur, wobei der Server gewisse Services zur Verfügung stellt, sobald ein Client diese anfordert (S. 1). Es erlaubt dadurch, die Präsentationsschicht (Frontend) von der Verarbeitungsschicht (Backend) zu trennen, wobei diese Entkopplung wesentliche Vorteile bietet. REST ist statuslos, was bedeutet, dass jeder Request vom Client alle Informationen beinhalten muss, die der Server benötigt, um den Request zu verarbeiten (S.2). Die Serverseite hat also keine Kenntnisse über den Zustand des Clients und verarbeitet die erhaltenen Request isoliert. Die wichtigste Eigenschaft von REST ist jedoch, dass Schnittstellen einheitlich sein müssen, sprich es wird clientunabhängig immer dieselbe Schnittstelle verwendet (S. 5). Der Client kann somit eine Mobile-Applikation oder eine Webseite sein, das Format des Requests ist dasselbe.

Bei REST wird die Repräsentation des Zustands von Ressourcen ausgetauscht, wobei Ressourcen dabei nahezu alles sein können (z.B. eine Webseite, ein Bild, eine Person, ein

Report) (S. 8). Die Ressource beschreibt dabei, welche Information vom Server übermittelt und welche Aktionen zur Verfügung stehen. Die Repräsentation kann beliebig sein (Binär, XML, BASE64, JSON), jedoch hat sich JSON für viele Anwendungsfälle als Standard etabliert (S.28). Der Zugriff auf Ressourcen erfolgt über HTTP/S Requests, dabei stehen die HTTP Methoden zur Verfügung, die dem Server anzeigen, welche Art von Service der Client anfordern (S.11). Die wichtigsten Methoden sind dabei GET (Ressourcen abrufen), POST (Ressourcen hinzufügen), PUT / PATCH (Ressourcen ändern), DELETE (Ressourcen löschen). Da REST keine Vorgaben macht, welche Methode für welche Aktion verwendet wird, gibt es dafür keinen strikten Standard. Da Requests und Responses über HTTP/S übermittelt werden, gelten die HTTP Status Codes. Der Status Code ist eine Zusammenfassung der vom Server übermittelten Response (S.16). Der bekanntesten dabei ist wohl 404 – Page not found. Die Status Codes lassen sich in folgende Kategorien gruppieren (S.17):

- **1xx Information** die Response ist rein informativ.
- **2xx Erfolgreich** der Request konnte erfolgreich verarbeitet werden.
- **3xx Vershoben** die Ressource wurde verschoben.
- **4xx Client Fehler** der Request war fehlerhaft
- **5xx Server Fehler** schwerwiegender Fehler auf dem Server

Wie bereits oben erwähnt, zeichnet sich REST durch einheitliche Schnittstellen aus. Um die Schnittstellen zu beschreiben, werden APIs definiert. API steht für Application Programming Interface. Es kann als Vertrag zwischen Client und Server verstanden werden. Eine API-Spezifikation für ein REST-API beschreibt, unter welchem URL und in welcher Form Ressourcen abrufbar sind, sowie welche Aktionen zur Verfügung stehen.

5.1.2 Single-Page-Application

Single-Page-Applications (SPA) beschreiben eine neue Form der Webentwicklung. Bei traditionellen Webapplikationen wurden Webseiten serverseitig aufgebaut und dann an den Client übermittelt, der Browser hat die erhaltenen Seiten dann geladen. Sobald ein Benutzer eine Aktion durchführte, wurde wiederum ein Request an den Server geschickt, dieser schickte eine neue Seite zurück, welche vom Browser erneut geladen wurde. Dabei befand sich die meiste Logik auf der Server-Seite (Fink & Flatow, 2014, S. 8). Durch die

Weiterentwicklung von verschiedenen Web-Technologien wie HTML und JavaScript sowie dem aufkommenden Bedürfnis, Webapplikationen auf unterschiedlichen Endgeräten anzuzeigen, entstand die SPA. Im Gegensatz zur traditionellen Webapplikation besteht die SPA aus lediglich einem HTML File, welches der Entry-Point zur Webseite ist (S.3) und als Hülle für alle Seiten, die angezeigt werden können, dient. Die eigentlichen Aktionen des Users, wie etwa das Navigieren auf der Seite, findet in JavaScript auf der Client Seite statt. Das HTML File, sowie sämtliche benötigten JavaScript und CSS Files werden beim ersten Request des Users vom Server in den Client geladen (Initial Load). Das Laden von Daten von der Server-Seite erfolgt über die zuvor beschriebenen REST-APIs. In der SPA ist daher definiert, bei welcher Aktion des Benutzers, welches REST-API verwendet wird. Die Response wird dann dynamisch in die Seite geladen, ohne dass ein Page-Refresh stattfindet.

5.1.3 Übersicht

Die Abbildung 2 zeigt die Übersicht dieser Architektur, welche ein allgemeingültiges Designpattern ist, das von Programmiersprachen und Datenbanken unspezifisch angewandt werden kann. Für die Implementation von SPAs und REST-APIs stehen verschiedene Programmiersprachen und Frameworks zu Verfügung. Auf die Wahl dieser wird in den Kapiteln Backend Umsetzung und Frontend Umsetzung eingegangen.

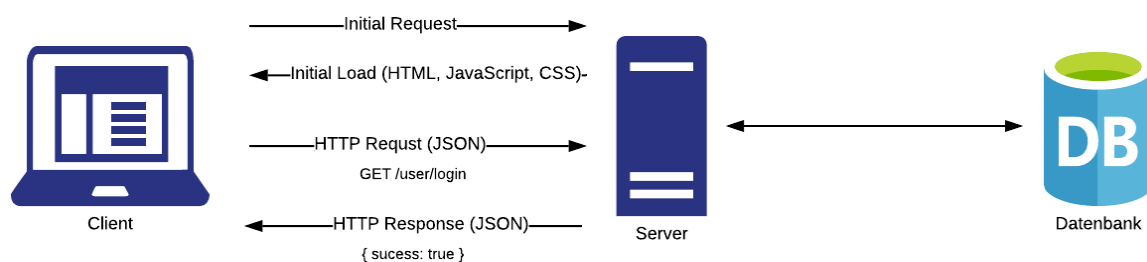


Abbildung 2: Applikationsarchitektur

5.2 Frontend Design

In der Folge wird die Benutzeroberfläche des Prototyps mittels Mockups beschrieben. Dabei handelt es sich um Entwürfe, welche vom fertigen Prototyp abweichen können. Sie wurden anhand der Anforderungen erstellt.

Es folgt die Beschreibung der einzelnen Seiten welche nach Login/Registrierung, Hauptseite und Admin-Bereich gegliedert wird.

5.2.1 Login/Registrierung

Der Einstieg in die Webseite erfolgt immer über die Login-Seite, hier kann sich der Benutzer mit seinen existierenden Login-Daten anmelden oder sich durch Klick auf «Registrieren» neu Login-Daten erstellen. Die beiden Seiten werden in der Abbildung 3 gezeigt.



The image shows two mockups for the login and registration process. On the left, the 'Tag Cloud' login form features two input fields for 'Email' and 'Passwort', a grey 'Einloggen' button, and a 'Registrieren' link. A blue arrow points from the 'Registrieren' link to the registration form on the right. The registration form, also titled 'Tag Cloud', includes input fields for 'Vorname', 'Nachname', 'E-Mail-Adresse', and 'Passwort', a grey 'Registrieren' button, and a 'zurück' link.

Abbildung 3: Mockup Login/Registrierung

Nach der Eingabe von gültigen Anmeldedaten wird der Benutzer auf die Hauptseite weitergeleitet.

5.2.2 Hauptseite

Die Hauptseite erfüllt sämtliche Anforderung betreffend des Hinzufügen von Tags, der Anzeige der Tag Cloud und deren Auswertungen. Nach dem Einloggen wird standardmässig die Tag Cloud des Benutzers mit den Tags des Typen «Arbeite an» angezeigt. Die Abbildung 4 zeigt den Entwurf der Hauptseite.

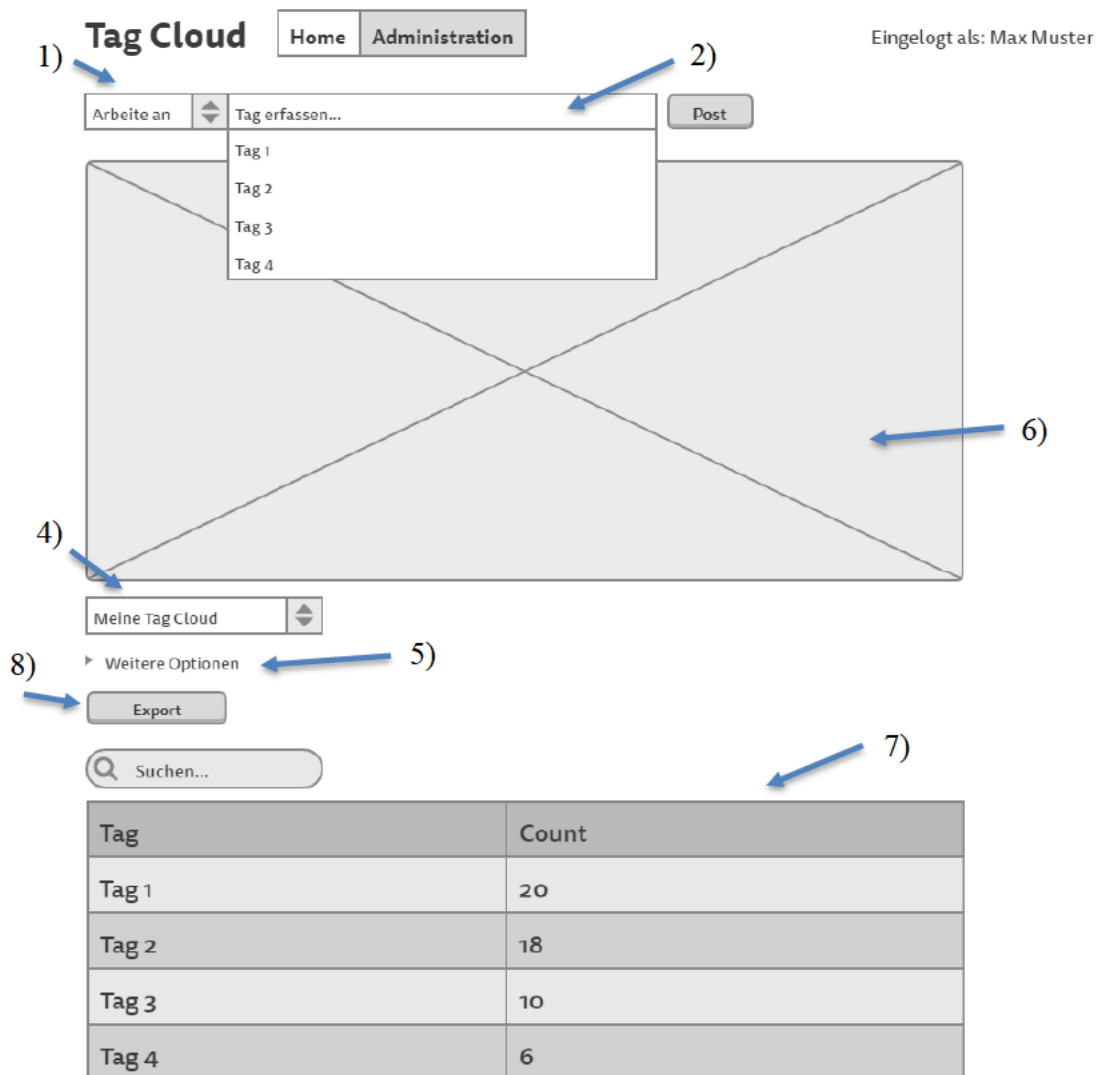


Abbildung 4: Mockup Hauptseite 1

Das **Dropdown 1)** verfügt über zwei Funktionen: Einerseits kann der Benutzer dadurch bestimmen, welchen Tag-Typen er hinzufügen möchte («Arbeite an» oder «Interessiert an»), andererseits werden bei einer Änderung auch jeweils nur die entsprechenden Tags in der Tag Cloud angezeigt. In der **Input-Box 2)** kann der User ein neues Tag erfassen, dabei werden ihm anhand seiner Eingaben bereits bestehende Tags aus dem Tag-Index angezeigt. Durch klicken kann er ein bestehendes Tag auswählen oder bestätigen, dass er ein neues Tag erstellen möchte. Mit Klick auf den Button «Post» wird das entsprechende Tag hinzugefügt. Im **Dropdown 4)** kann der Benutzer auswählen, ob er seine persönliche Tag Cloud («Meine Tag Cloud»), die Tag Cloud seiner oder einer beliebigen Gruppe, die Tag Cloud eines beliebigen Benutzers sowie diejenige sämtlicher Benutzer («Alle») an-

zeigen möchte. Mit Klick auf den **Link 5)** werden dem Benutzer weitere Optionen angezeigt, um seine Auswertung einzuschränken oder die Visualisierung der Tag Cloud **6)** anzupassen. Die **Tag Cloud 6)** zeigt eine Tag Cloud, anhand derjenigen Tags, die den Kriterien des Users entsprechen. Dieselben Tags werden in der **Tabelle 7)** angezeigt. Zur Unterstützung der gezielten Suche eines Tags, gibt es ein Suchfeld, die Eingaben werden auf die Tabelle angewandt. Mit dem **Export-Button 8)** kann der Benutzer Reports über die angezeigten Tags herunterladen. Abbildung 5 zeigt die zusätzlichen Einstellungen, die dem Benutzer bei Klick auf den Link 5) zur Verfügung stehen. Er hat die Möglichkeit, die Anzahl Tags in der Tag Cloud zu beschränken, den Zeitraum, aus dem Tags abgerufen werden sollen, festzulegen und die Einfärbung der Tags zu deaktivieren.

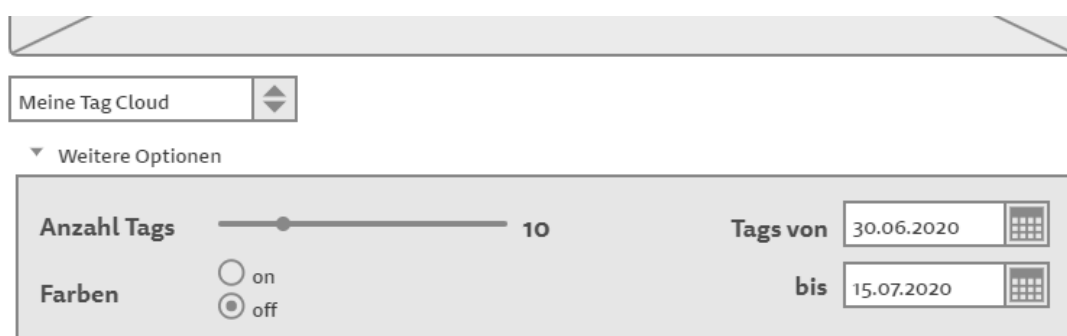


Abbildung 5: Mockup Hauptseite 2

Bei einem Klick auf ein Tag in der Tag Cloud **2)** oder in der Tabelle **7)** sollen die Detail-Informationen zum gewählten Tag angezeigt werden. Diese sind in Abbildung 6 ersichtlich. Im oberen Teil werden die allgemeinen Informationen zum Tag angezeigt. Im unteren Teil werden alle Benutzer, die das ausgewählte Tag bereits einmal gepostet haben, angezeigt. Dabei wird gruppiert nach Benutzern, die das Tag als «Interessiert an» oder «Arbeite an» gepostet haben.

BeispielTag																					
Erstellt von:	Max Muster																				
Erstellt am:	29.03.2020 15:30																				
Geändert von:	Max Muster																				
Geändert am:	29.03.2020 15:30																				
Total Posts:	15																				
Erstes mal gepostet am:	29.03.2020 15:30																				
Letztes mal gepostet am:	13.05.2020 15:30																				
<table border="1"> <thead> <tr> <th colspan="5">Interessierte Benutzer</th> </tr> <tr> <th>Vorname</th> <th>Nachname</th> <th>Anzahl Posts</th> <th>Letzter Post</th> <th>Erster Post</th> </tr> </thead> <tbody> <tr> <td>Max</td> <td>Muster</td> <td>5</td> <td>29.03.2020 15:30</td> <td>29.03.2020 15:30</td> </tr> <tr> <td>John</td> <td>Doe</td> <td>2</td> <td>13.05.2020 19:30</td> <td>13.05.2020 19:30</td> </tr> </tbody> </table>		Interessierte Benutzer					Vorname	Nachname	Anzahl Posts	Letzter Post	Erster Post	Max	Muster	5	29.03.2020 15:30	29.03.2020 15:30	John	Doe	2	13.05.2020 19:30	13.05.2020 19:30
Interessierte Benutzer																					
Vorname	Nachname	Anzahl Posts	Letzter Post	Erster Post																	
Max	Muster	5	29.03.2020 15:30	29.03.2020 15:30																	
John	Doe	2	13.05.2020 19:30	13.05.2020 19:30																	
<table border="1"> <thead> <tr> <th colspan="5">Arbeitende Benutzer</th> </tr> </thead> <tbody> </tbody> </table>		Arbeitende Benutzer																			
Arbeitende Benutzer																					
<input type="button" value="Schliessen"/>																					

Abbildung 6: Mockup Tag Details

5.2.3 Administration

Der Administrationsbereich ist nur für Benutzer mit der entsprechenden Administrator Rolle ersichtlich. Darin sind drei wesentliche Funktionen verfügbar: Der Index-Editor (vgl. Abbildung 7), die Benutzerverwaltung (vgl. Abbildung 8) und die Gruppenverwaltung (vgl. Abbildung 9). Alle drei Bereiche sollen über editierbare Tabellen verfügen, ausserdem soll in den Tabellen gesucht werden können.

Im Index Editor können existierende Tag gesucht, bearbeitet und gelöscht werden. Zudem lassen sich neue Tags zum Index hinzufügen. Bei der Änderung eines Tags muss beachtet werden, dass keine Duplikate zugelassen werden.



Abbildung 7: Mockup Index Editor

In der Benutzerverwaltung können Benutzer geändert werden. Dazu gehört, dass Benutzer einer Gruppe zugewiesen oder zum Administrator ernannt werden können.

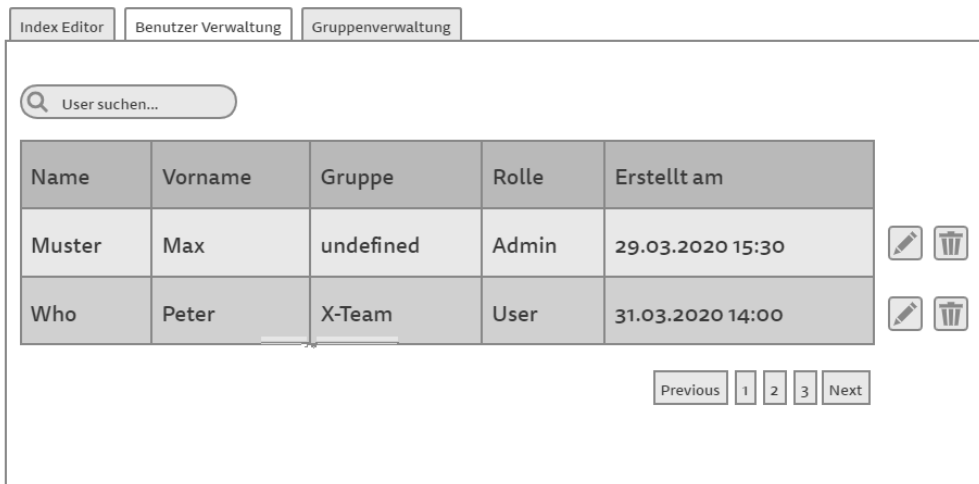


Abbildung 8: Mockup Benutzerverwaltung

Die Gruppenverwaltung ermöglicht es, Gruppen zu erstellen, zu ändern oder zu löschen. Bei der Löschung werden Benutzer der zu löschenden Gruppe der Default Gruppe zugewiesen. Auch hier sind keine Duplikate Gruppennamen zugelassen.



Abbildung 9: Mockup Gruppenverwaltung

5.3 Backend Design

Anhand der Anforderungen und dem Frontend Design, lassen sich die Anforderungen an das Backend ableiten. In einem ersten Schritt wird das Datenmodell für die Datenbank entworfen, darauf folgt die API-Spezifikation.

5.3.1 Datenmodell

Die Datenmodellierung ergab das in Abbildung 10 gezeigte ERD-Modell. Die einzelnen Entitäten werden in der Folge erläutert.

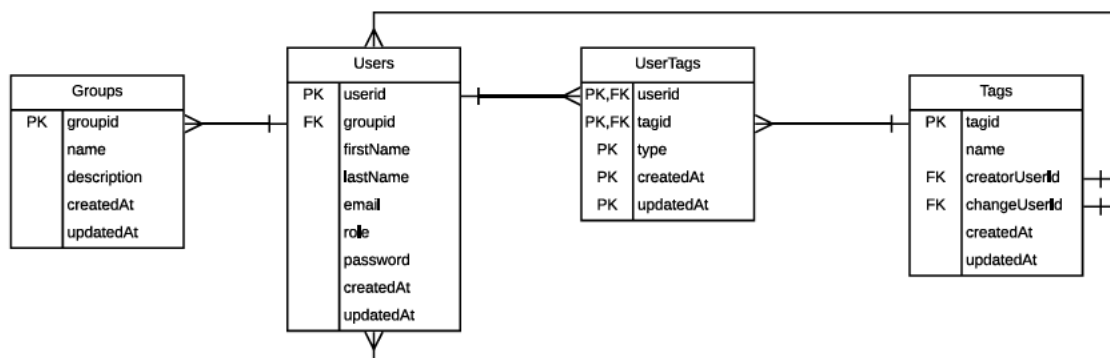


Abbildung 10: ERD-Modell

5.3.1.1 Tags

In der Entität Tag werden die eigentlichen Tags gespeichert, der Wert des Tags wird in der Spalte 'name' gespeichert. Aus Performance-Überlegungen wird 'name' hier nicht als Primärschlüssel verwendet, da dies negativen Einfluss auf Auswertungen über die Tabelle UserTags hätte. Jedoch wird serverseitig sichergestellt, dass keine Duplikate eingefügt werden. Diese Tabelle hat die Rolle des Tag-Index, darin sind sämtliche verfügbaren Tags enthalten.

5.3.1.2 Users

Jeder Eintrag in dieser Tabelle entspricht einem User. Für die Spalte 'role' sind die Werte «admin» für Administratoren und «user» für normale Benutzer vorgesehen.

5.3.1.3 Groups

In dieser Tabelle werden die Gruppen gespeichert. Jeder Benutzer in der Tabelle Users gehört zu einer Gruppe. Die Initial-Gruppe für neue Benutzer wird unter dem Primärschlüssel 1 angelegt.

5.3.1.4 UserTags

In dieser Tabelle werden die geposteten Tags gespeichert. Der kombinierte Primärschlüssel umfasst sämtliche Spalten, wodurch jeder eingefügte Datensatz einmalig wird. Wenn der Benutzer ein neues Tag postet, welches in der Tabelle Tags noch nicht existiert, wird dieses dort zuerst eingefügt. Danach wird der eigentliche Post in dieser Tabelle gespeichert. Anhand der Spalte 'type' wird zwischen Tags die als «Arbeite an» und «Interessiert an» erfasst wurden, unterschieden.

5.3.2 API-Design

Basierend auf dem Datenmodell und den Anforderungen, die sich aus dem Frontend Design ergeben, wurde das API spezifiziert. Dafür wurde eine OpenAPI Spezifikation im YAML-Format erstellt. Diese kann im Swagger Editor (Swagger, o.J.) in einem benutzerfreundlichen Format angezeigt werden, wie in Abbildung 11 gezeigt wird. Im Swagger Editor können zudem Beispiele für Requests und Responses angezeigt werden. Das

Schloss in Abbildung 11 zeigt an, dass man für die Nutzung dieses Endpoints authentifiziert sein muss, das zugrundeliegende Konzept wird in Kapitel 5.4.2.1 erläutert.

Tag Cloud API 1.0 OAS3

Servers

Authorize

users

- POST** `/users/signup` Register as a new user.
- PATCH** `/users/{userId}` Update an existing user.
- POST** `/users/login` Login with credentials to receive a token
- GET** `/users` Returns a list of users

group

- GET** `/groups` Get a list of all groups.
- POST** `/groups` Add a new group.
- PATCH** `/groups/{groupId}` Update an existing group.
- DELETE** `/groups/{groupId}` Delete a group.

tags

- POST** `/tags` Post a tag
- GET** `/tags` Get number of posts of tags matching the parameters
- GET** `/tags/{tagId}` Get the Details of a specific tag.

tags/index

- GET** `/tags/index` Get tags from index
- POST** `/tags/index` Add a Tag to the index.
- PUT** `/tags/index/{tagId}` Update an existing tag
- DELETE** `/tags/index/{tagId}` Delete Tag by Id.

Abbildung 11: OpenAPI Spezifikation

In der Tabelle 5 wird die Business-Logik der einzelnen Endpoints erläutert, dabei wird bereits auf das Autorisierungskonzept eingegangen. Für den Prototyp sind die Rollen ‘user’ und ‘admin’ vorgesehen, in der Tabelle ist ersichtlich, welchen Benutzern welche Endpoints zur Verfügung stehen.

Tabelle 5: Beschreibung Endpoints

Pfad	HTTP Methode	Rolle	Beschreibung
/users/signup	POST	alle	Ermöglicht es Benutzer sich für die Applikation zu registrieren. Dazu übermitteln sie ihren Namen, Vornamen, eine E-Mail-Adresse und ihr gewähltes Passwort, dieses wird serverseitig verschlüsselt, bevor es in der Datenbank gespeichert wird.
/users/login	POST	alle	Erlaubt es Benutzern, sich anzumelden (vgl. Kapitel 5.4.2.1).
/users/{id}	PATCH	alle	Der Benutzer mit der übermittelten ID wird geändert. Admin-Benutzer können sämtliche Benutzer ändern, Standardbenutzer nur ihren eigenen Benutzer.
/users	GET	alle	Gibt eine Liste mit allen Benutzern zurück. Benutzer mit der Rolle ‘admin’, erhalten zudem die Rolle der Benutzer.
/groups	GET	alle	Gibt eine Liste mit allen Gruppen zurück. Jede Gruppe zeigt die darin enthaltenen User.
/groups	POST	admin	Ermöglicht das Erstellen einer neuen Gruppe.
/groups/{id}	PATCH	admin	Die Gruppe mit der übermittelten ID wird geändert.

/groups/{id}	DELETE	admin	Die Gruppe mit der übermittelten ID wird geändert. Die Benutzer der gelöschten Gruppe werden der Default-Gruppe zugewiesen.
/tags	POST	alle	Ein Tag wird gepostet, sollte es noch nicht existieren, wird es dem Tag-Index hinzugefügt.
/tags	GET	alle	Gibt eine Liste aller Tags und deren Anzahl Posts zurück, die den übermittelten Parameter (z.B. Typ, Zeit von-bis und/oder Gruppe) entsprechen.
/tags/{id}	GET	alle	Die Detail-Informationen zu einem Tag werden zurückgegeben.
/tags/index	GET	alle	Gibt alle verfügbaren Tags aus dem Index zurück.
/tag/index	POST	admin	Ein Tag lediglich dem Tag-Index hinzufügen, jedoch nicht posten.
/tags/index/{id}	PUT	admin	Ein Tag im Tag-Index ändern.
/tags/index/{id}	DELETE	admin	Ein Tag aus dem Index entfernen, dabei werden alle zugehörigen Posts ebenfalls gelöscht.

5.4 Backend Umsetzung

Anhand der Informationen aus dem vorherigen Kapitel kann das Backend umgesetzt werden. Das Backend setzt sich aus der Datenbank für die Datenspeicherung und des REST-APIs zusammen. Der Autor dieser Arbeit hat sich für eine Umsetzung mit Node.js und PostgreSQL entschieden. Node.js wurde 2009 entwickelt und baut auf der Google V8 Engine auf, die es ermöglicht, JavaScript serverseitig auszuführen (Doglio, 2015, S. 47). PostgreSQL (o.J.), kurz Postgres, ist ein objekt-relationales open source Datenbanksystem. Postgres vereint dabei Funktionalitäten von klassischen relationalen Datenbanken (z.B. MySQL Database) mit objektorientierten NoSQL Ansätzen. Diese Wahl wird in der

Folge kurz begründet, ist aber auch wesentlich von den Präferenzen der umsetzenden Person abhängig.

Bereits in Kapitel 5.1 wurde der Wandel von herkömmlichen Webapplikation hin zu JavaScript basierten SPA erwähnt. Ähnlich zeigt sich die Situation serverseitig, wo die Verwendung von JavaScript mittels Node.js immer verbreiteter wird. Viele der Gründe dafür sind nicht primär Node.js geschuldet, sondern auf JavaScript und dessen Beliebtheit zurück zu führen. JavaScript ist eine verhältnismässig einfache Programmiersprache, die den Basisprinzipien von Script-Sprachen folgt. Dadurch können Programmierer in kürzester Zeit die Konzepte verstehen und mit der Umsetzung beginnen (Doglio, 2015, S. 56). Ein weiterer wesentlicher Vorteil ist, dass sowohl auf der Client- als auch auf der Serverseite dieselbe Programmiersprache verwendet werden kann. Zudem überzeugt Node.js mit starker Performance, die es hauptsächlich dem Single-Thread Ansatz verdankt, und verfügt mit npm (Node Package Manager) eine Community mit einer grossen Zahl an Erweiterungen. Für REST-APIs eignet sich Node.js besonders gut, da es hervorragend mit JSON-basierten Services interagiert (S.56). Für den Autor dieser Arbeit war hauptsächlich die Idee einer einzigen Programmiersprache (JavaScript) für die Umsetzung von Client und Server ausschlaggebend.

Die Many-to-Many Beziehung zwischen den Entitäten Users und Tags spricht gegen die Verwendung einer NoSQL Datenbank, weshalb nach einer geeigneten relationalen Datenbank gesucht wurde. Die Wahl fiel auf Postgres, es wurden für die Umsetzung der Datenbank jedoch keine Postgres spezifischen Funktionen verwendet, wodurch Postgres durch eine beliebige relationale Datenbank (z.B. MySQL) ersetzt werden könnte.

Folgende Softwareversionen wurden für das Backend verwendet: Für Node.js kamen diverse Erweiterungen zum Einsatz (in Tabelle 6 zusammengefasst dargestellt), solche mit hoher Relevanz für diese Arbeit werden in den nächsten Kapiteln genauer erläutert.

- PostGresSQL Version 12.2
- Node.js Version 12.16.1

Tabelle 6: Node.Js Erweiterungen

Name	Version	Beschreibung
Bcryptjs	2.4.3	Für die Passwortverschlüsselung (vgl. Kapitel 5.4.2.1)
Body-parser	1.19.0	Ermöglicht das Parsen von Request-Bodies
Concurrently	5.2.0	Zur Ausführung von mehreren Commands im Startskript
Cors	2.8.5	CORS Middleware
Express	4.17.1	Webframework für Node.js
Helmet	3.21.3	Für HTTP/S Headers in Express Applikationen
Jsonwebtoken	8.5.1	JWT Verwaltung (vgl. Kapitel 5.4.2.1)
Pg	7.18.2	Diverse Postgres Erweiterungen und Adapter
Pg-hstore	2.3.3	
Psql	0.0.1	
Sequelize	5.21.5	Node.js ORM (vgl. Kapitel 5.4.1)

5.4.1 Datenbank / ORM Sequelize

Zur Erstellung des Datenbankschemas wurde die Erweiterung Sequelize verwendet. Sequelize ist ein ORM für Node.js. Es unterstützt dabei verschiedene relationale Datenbanksysteme wie MySQL, Microsoft SQL Server oder Postgres (Sequelize, o.J.). ORM steht für Object-Relational Mapper und bieten den Vorteil, dass keine SQL-Scripts oder Abfragen geschrieben werden müssen. Die Objekte werden in einer beliebigen Programmiersprache definiert, und ORM übersetzt die Objekte und die zugehörigen Methoden (z.B. lesen, schreiben, ändern) in die Datenbanksprache. Dies hat den Vorteil, dass viele Methoden bereits durch das ORM zur Verfügung gestellt werden und nicht wiederholt programmiert werden müssen. Ausserdem kann sich der Entwickler so primär auf die

Objekte fokussieren. Sequelize bietet mit sogenannten «Migrations» zusätzlich die Möglichkeiten, SQL-Skripts für die Erstellung von Tabellen und Beziehungen anhand von Definitionen in JavaScript zu generieren. Dazu müssen zuerst die Objekte mit den zugehörigen Beziehungen modelliert werden. Die Beziehungen werden mit ORM typischen Bezeichnungen («hasOne», «hasMany», «belongsTo» usw.) modelliert. Dadurch kann Sequelize Beziehungen bei Datenabfragen selbständig auflösen, ohne dass SQL-Abfragen mit Join-Statements entwickelt werden müssen. Der nachfolgende Codeausschnitt zeigt die zuvor genannte Modellierung am Beispiel des User Objekts:

```
User.associate = function(models) {
  // each user belongs to a group
  User.belongsTo(models.Group, {
    foreignKey: 'groupid',
    onDelete: 'CASCADE',
  });
  // each user has many tags posted
  User.hasMany(models.UserTag, {
    foreignKey: 'userid',
    onDelete: 'CASCADE',
  });
  // each user has many tags created
  User.hasMany(models.Tag, {
    as: 'creatorUser',
    foreignKey: 'creatorUserId',
    onDelete: 'CASCADE',
  });
  // each user has many tags edited
  User.hasMany(models.Tag, {
    as: 'changerUser',
    foreignKey: 'changerUserId',
    onDelete: 'CASCADE',
  });
};
```

Danach können Migration-Files generiert und vorbereitet werden. Im Gegensatz zu den Objekten, die lediglich die Attribute und Beziehungen kennen, werden hierfür auch Datenbank typische Attribute wie Primär- und Fremdschlüssel, Not Null Constraints, Datentypen und erweiterte Optionen wie «On Delete Cascade» definiert. «On Delete Cascade» führt dazu, dass beim Löschen eines Primärschlüssel, die darauf referenzierenden Datensätze mitgelöscht werden. Die Migration-Files können über das Commandline

Interface (sequelize-cli) auf der Datenbank durchgeführt werde, die hierzu benötigten SQL-Befehle generiert Sequelize zur Laufzeit. Der dafür verwendete Datenbankdialekt ist konfigurierbar, wodurch die Abhängigkeit zu dem Datenbank-Provider weiter reduziert wird.

Die Verwendung von Sequelize hatte in dieser Arbeit die Folge, dass keine SQL-Skripts zur Erstellung des Datenbankschemas und bis auf wenige Ausnahmen, keine Datenbankabfragen mittels SQL implementiert wurden.

5.4.2 REST-API

Das API wurde vorab spezifiziert (vgl. Kapitel 5.3.2) und dann mittels Node.js und Express.js implementiert. Express.js ist ein leichtgewichtiges Webframework für Node.js, das Tools für HTTP/S Server zur Verfügung stellt. Es eignet sich damit für viele Arten von Webapplikationen so auch für REST-APIs (Doglio, 2015, S. 85).

Bereits in Kapitel 5.3.2 wurde angedeutet, dass gewisse Endpoints eine Authentifizierung benötigen. Da es sich beim IT-Artefakt dieser Arbeit um einen Prototyp handelt, wurde eine Basic Authentifizierung umgesetzt, die nun erläutert wird.

5.4.2.1 Authentifizierung und Autorisierung

Die Authentifizierung beantwortet die Frage, wer einen Endpoint aufruft. Die Autorisierung hingegen beantwortet, welche Funktionalitäten dem eingeloggen Benutzer zur Verfügung stehen (Doglio, 2015, S. 37). In Kapitel 5.1.1 wurde REST als zustandslos definiert, das Speichern von Benutzer-Sessions auf dem Server widerspricht diesem Paradigma. Jedoch gibt es bis heute keine absolut zustandslose Authentifizierungsmethode für REST-APIs (Doglio, 2015, S. 37).

Bei der Registrierung von Benutzern (Endpoint /users/signup) wird ein Passwort im HTTP/S Request übermittelt, dieses wird serverseitig mittels der Erweiterung bcryptjs unter Verwendung eines geheimen Wertes (Salt) verschlüsselt und in der Datenbank gespeichert. Ab diesem Zeitpunkt kann sich der Benutzer mittels Endpoint /users/login Authentifizieren. Dafür übermittelt der Benutzer sein Passwort und seine E-Mail-Adresse, serverseitig wird geprüft, ob die E-Mail-Adresse in der Datenbank verfügbar und das Passwort mit dem verschlüsselten Passwort in der Datenbank übereinstimmt. Wenn dies

der Fall ist, wird dem Benutzer ein Token ausgestellt. Dieses Token ist für eine konfigurierbare Zeit gültig und wird auf dem Server gespeichert. Die Erweiterung `jsonwebtoken` übernimmt dabei das Prüfen, Verwalten und Ausstellen von Token. Dadurch, dass dem Token Objekte zugewiesen werden können, lässt sich auch die Autorisierung realisieren. Der folgende Codeausschnitt zeigt die Verwendung von `jsonwebtoken`, das dabei erstellte Token ist für einen Tag gültig und wird um das Objekt `userDetails` erweitert, die Rolle ('user' oder 'admin') findet sich im Attribut `is_auth`.

```
const jwt = require('jsonwebtoken');
...
var userDetails = {
    userid: user[0].dataValues.userid,
    firstName: user[0].dataValues.firstName,
    lastName: user[0].dataValues.lastName,
    email: user[0].dataValues.email,
    groupid: user[0].dataValues.groupid,
    is_auth: user[0].dataValues.role
}

var token = jwt.sign({
    user: userDetails
}, secret, {
    expiresIn: '1d'
});
...
```

Der Client erhält in der Response das Token zurück und muss dieses für die Verwendung der geschützten Endpoints (vgl. Kapitel 5.3.2) im Request Header-Feld `Authorization` übermitteln. Das Token wird dann validiert und die zugehörigen Benutzerdaten werden wieder extrahiert. Somit kann serverseitig für jeden Request geprüft werden, ob 1) das Token valid ist (Authentifizierung) und sofern diese Prüfung positiv ausfällt, 2) welche Rolle der Benutzer innehat. Somit lassen sich Endpoints den Anforderungen entsprechend schützen. Der nachfolgende Beispielcode zeigt, dass der Endpoint um neue Gruppen zu erstellen, nur für User mit der Rolle 'admin' zur Verfügung steht.

```

router.post('/groups', checkJWT, (req, res, next) => {
  // get role provided in token
  const role = req.decoded.user.is_auth;
  // check if role is sufficient
  if (role !== 'admin'){
    return res.sendStatus(403);
  }
  next();
}, GroupController.createGroup);

```

Falls ein Benutzer ohne 'admin' Rolle diesen Endpoint verwendet, wird eine Status Code 403 (Forbidden) zurückgegeben.

5.4.2.2 Aufbau Backend

Der Code des Backend ist wie folgt gegliedert:

app.js	Beinhaltet die Serverdefinition für Node.js / Express.js und ist der Einstiegspunkt für sämtliche HTTP/S Requests.
package.json	Projekt Definition, Build- und Abhängigkeitsmanagement.
/config/database.js	In diesem File wird die Verbindung zur Datenbank aufgebaut und exportiert.
/routes/api.js	Konfiguration der Endpoints und der Weiterleitung zu den entsprechenden Funktionen in den Controller
/middlewares	Hier sind Middleware Funktionen abgelegt, wie etwa die Funktion zur Prüfung der Token (checkj-jwt.js).
/middlewares/config	Sequelize Konfigurationsdatei (config.js), in welcher der Datenbankdialekt (Postgres) konfiguriert wird.
/middlewares /models	Hier befinden sich die ORM Objekt Definitionen (vgl. Kapitel 5.4.1).
/middlewares/migrations	Migrationsdateien von Sequelize zur Erstellung der Tabellen in der Datenbank.

/middlewares/seeders	Sequelize Dateien um Daten in die Datenbank zu laden, verwendet um die Tabellen mit initialen Daten zu befüllen.
/controllers	Pro Ressource (vgl. Kapitel 5.1.1) ein Controller File, mit sämtlichen JavaScript Funktionen, die im API spezifiziert sind.

5.4.3 Testing

Um das Backend zu testen, wurde das Tool Postman (o.J.) in der Version 7.24.0 verwendet. Es übernimmt dabei die Rolle des Clients und ermöglicht, Requests über eine intuitive Benutzeroberfläche an den Server zu übermitteln und die Response anzuzeigen. Es erlaubt zudem Testfälle vorzubereiten und auszuführen. Eine weitere nützliche Funktion ist, dass OpenAPI Spezifikationen importiert werden können und Postman für jeden vorhandenen Endpoint einen Testfall mit den benötigten Parametern generiert. Dadurch lassen sich Tests in kürzester Zeit vorbereiten, und die Arbeit, die in das Erstellen der API-Spezifikation investiert wurde, effizient nutzen. Durch das Testen der Endpoints wird sichergestellt, dass das im nächsten Schritt umzusetzende Frontend Zugriff auf die Daten in der Datenbank hat, sowie dass die Authentifizierung und Autorisierung wie spezifiziert funktioniert.

5.5 Frontend Umsetzung

In Kapitel 5.1.2 wurde das Konzept der Single-Page-Application vorgestellt. Für die Umsetzung der SPA hat sich der Autor für React.js entschieden. React.js ist eine Open Source JavaScript Library die 2013 von Facebook veröffentlicht wurde und der Erstellung von User Interfaces dient (Thakkar, 2020, S. 41).

5.5.1 React.js Konzepte

Für eine bessere Lesbarkeit, wird React.js in der Folge mit React abgekürzt. Wie bereits zuvor erwähnt, handelt es sich bei React um eine Library, im Gegensatz zu einem Framework, ist React aber keine vollständige Lösung und muss häufig um weitere Libraries

ergänzt werden. Dies hat den Vorteil, dass React eine kleine Library ist, die beliebig ergänzt werden kann. React ist Komponenten basiert. Komponenten erhalten Input-Parameter, liefern als Output ein UI Element und sind wiederverwendbar (Thakkar, 2020, S. 45). Es können Komponenten selber programmiert sowie von Dritten zur Verfügung gestellte Komponenten oder Libraries verwendet werden, dies funktioniert weitgehend identisch wie die Verwendung von Erweiterungen in Node.js. Komponenten werden dynamisch interpretiert, React erkennt dabei Inputänderungen und passt die Ausgabe des Frontend Elements automatisch an. In dieser Arbeit wurden ausschliesslich Funktionale-Komponenten verwendet, die als JavaScript Funktionen aufgebaut sind, welche als Output eine React-Komponente erstellen. Während früher gewisse Funktionen nur für Klassen-Komponenten zur Verfügung standen, können seit der React Version 16.8 über APIs sämtliche Funktionen auch in Funktionalen-Komponenten verwendet werden. Diese haben den Vorteil, dass sie einfacher zu interpretieren sind, da sie in der Umsetzung wesentlich weniger Code benötigen. Es wird zwischen Komponenten mit und ohne Zustand unterschieden. Während erstere vollständig über die Input-Parameter („Props“) gesteuert werden, haben zweitere Kenntnisse über den Zustand in der Applikation. Der Zustand (engl. State) ist nebst den Komponenten ein weiteres wichtiges Element in React Frontend Applikationen. Redux (o.J) ist die Erweiterung, die ein einfaches Zustands-Management in React Applikationen ermöglicht. Zustände sind etwa Server Responses, zwischengespeicherte Daten oder clientseitige Daten, die serverseitig noch nicht persistiert wurden (Redux, o.J). Durch die Verwendung von Redux werden Zustände genau und verlässlich definiert und zentral verwaltet. Das letzte Konzept hängt mit der Charakteristik von SPA zusammen, React Router ermöglicht es clientseitig zwischen den verschiedenen Seiten zu navigieren, ohne die Page neu zu laden. Dafür wird auf der Einstiegsseite der SPA ein Router konfiguriert, indem Seiten den verschiedenen URLs zugewiesen werden. Wenn der Benutzer durch die Seite navigiert, weiss React welche Seite bzw. welche Komponente angezeigt werden muss (Maki, 2017).

Während Redux bei den meisten React Applikationen eingesetzt wird, wurden noch weitere Erweiterungen verwendet, die die Entwicklung des Prototyps prägten. Diese werden im nächsten Kapitel vorgestellt.

5.5.2 Erweiterungen

5.5.2.1 React-bootstrap

React-bootstrap (o.J) ist die Bootstrap Implementierung für React. Bootstrap ist ein Open Source Frontend-Framework, das Web-Entwickler bei der Erstellung von Websites unterstützt und den Entwicklungsprozess beschleunigen soll. Es eignet sich damit optimal für das Erstellen von Prototypen. Die Library stellt HTML und CSS basierte Design Templates der gängigen Elemente wie Formulare, Buttons, Navigationsleisten zur Verfügung. Während Bootstrap ursprünglich aus CSS, HTML und optionalen JavaScript Plugins bestand, werden in der react-bootstrap Library dieselben Elemente in der Form von React Komponenten zur Verfügung gestellt. Durch die Verwendung von vordefinierten Komponenten, die auf Grund des Bootstrap CSS bereits ansprechend aussehen, lassen sich in kurzer Zeit die wesentlichen Elemente einer Webseite aufbauen, ohne das viel Zeit in die Verbesserung der Optik investiert wird. In dieser Arbeit wurden über neun verschiedene Komponenten der react-bootstrap Library verwendet, primär für die zuvor genannten Grundelemente einer Webseite.

Verwendete Version: 1.0.0-beta.17 (Bootstrap 4)

5.5.2.2 React-wordcloud

Die Erweiterung react-wordcloud (o.J) bietet eine React Komponente zur Erstellung von Tag Clouds. Diese Komponente ist wohl die wichtigste Erweiterung für das in dieser Arbeit erstellte Frontend. Die Library verwendet intern die d3-cloud, eine JavaScript Library zur Erstellung von Tag Clouds. React-wordcloud erlaubt es über diverse Parameter, das Erscheinungsbild anzupassen und lässt sich mit eigenen Funktionen erweitern. Dies ermöglichte es, eine selbst geschriebene Funktion zu verwenden, um die Text Farbe des jeweiligen Tags zu bestimmen. Die Erweiterung ist somit in der Lage, sämtliche in Kapitel 4.3 formulierten Anforderungen an das Erscheinungsbild der Tag Cloud zu erfüllen.

Verwendete Version: 1.2.2

5.5.2.3 Diverse

Zusätzlich wurden die in der Tabelle 7 zusammengefassten Erweiterungen verwendet:

Tabelle 7: Diverse React Erweiterungen

Name	Version	Beschreibung
moment.js	2.18.1	Für das Umrechnen von Zeitstempel und Formatieren von Datum-Strings.
react-bootstrap-icons	1.0.1-alpha3	Bootstrap Icons für React wurden im Administrationsbereich verwendet für die Buttons in den Tabellen.
react-bootstrap-range-slider	0.3.1	Slider Eingabefeld für die Auswahl der Anzahl anzuzeigenden Tags auf der Hauptseite.
react-bootstrap-table-next react-bootstrap-table2-editor react-bootstrap-table2-filter react-bootstrap-table2-paginator react-bootstrap-table2-toolkit	4.0.0 1.4.0 1.3.1 2.1.2 2.1.2	Tabelle im Bootstrap Style mit zusätzlichen Funktionen wie Tabelle nach CSV exportieren, suchen, sortieren, bearbeiten und blättern. Umfasst wesentlich mehr Funktionen als die Tabelle in der react-bootstrap Erweiterung.
react-bootstrap-typeahead	5.0.0-alpha.1	Inputfeld mit automatischem Vorschlag anhand bestehender Daten. Wurde auf der Hauptseite für die Erfassung der Tags verwendet.
react-modal-promise	0.5.2	Benötigt um Asynchrone Modals zu verwenden, damit der Benutzer seine Änderung bestätigen oder abbrechen kann.
react-dates	21.8.0	Von Airbnb erstellte Datumsauswahl-Komponente, wird verwendet um den Zeitraum für die anzuzeigenden Tags einzugrenzen
save-svg-as-png	1.4.17	JavaScript Library die ein SVG Element in PNG konvertiert. Wird verwendet, um

		die von react-wordcloud erstellte Tag Cloud im SVG Format nach PNG zu konvertierten und herunterladen, sowie für das Hinzufügen der Tag Cloud zum PDF Report.
@react-pdf/renderer	2.0.0-beta.5	Ermöglicht die clientseitige Erstellung von PDF-Dateien.
@david.kucsai/react-pdf-table	0.3.0	Erweiterung für react-pdf, ermöglicht das dynamische Erstellen von Tabellen auf dem PDF-Report.

5.5.3 Aufbau Frontend

Der Code des Frontends ist wie folgt gegliedert:

- `/_actions` Actions sind JavaScript Funktionen, die den Zustand in Redux verändern können. Eine Action kann einen API-Aufruf beinhalten oder auch nur einen Redux-Zustand verändern.
- `/_components` Unterstützende Komponenten, die sicherstellen, dass Benutzer nur auf Routes gelangen, für die sie auch berechtigt sind.
- `/_constants` In Redux verwendete Konstanten für die verschiedenen Zustände.
- `/_helpers` Verschiedene unterstützende Funktionen etwa zur Einbindung und Konfiguration von Redux oder aufbereiten des Authorisation-Headers für API-Calls.
- `/_reducers` Reducer sind JavaScript Funktionen, die den letzten Zustand und eine Aktion als Input erhalten, um den nächsten Zustand festzulegen und abzubilden.
- `/_services` JavaScript Funktionen für die API-Aufrufe.
- `/App` Ist der Einstiegspunkt der SPA, hier ist der React-Router definiert, der basierend auf der angeforderten URL eine Seite anzeigt.

/NamePage Komponenten deren Namen mit „Page“ enden, sind Seiten, die im Router einer URL zugewiesen sind.

/Name} Komponenten die auf den Seiten verwendet werden, z.B. für spezielle Inputfelder.

5.5.4 Komponenten

In diesem Kapitel werden die wichtigsten erstellten Komponenten und ergänzende Informationen dazu erläutert. Abbildung 12 zeigt eine Übersicht der Komponenten der Hauptseite (TagCloudPage) und des Administrationsbereichs (AdminPage). Dabei wurden unwesentliche Komponenten wie Buttons oder einfache Formulare nicht abgebildet.

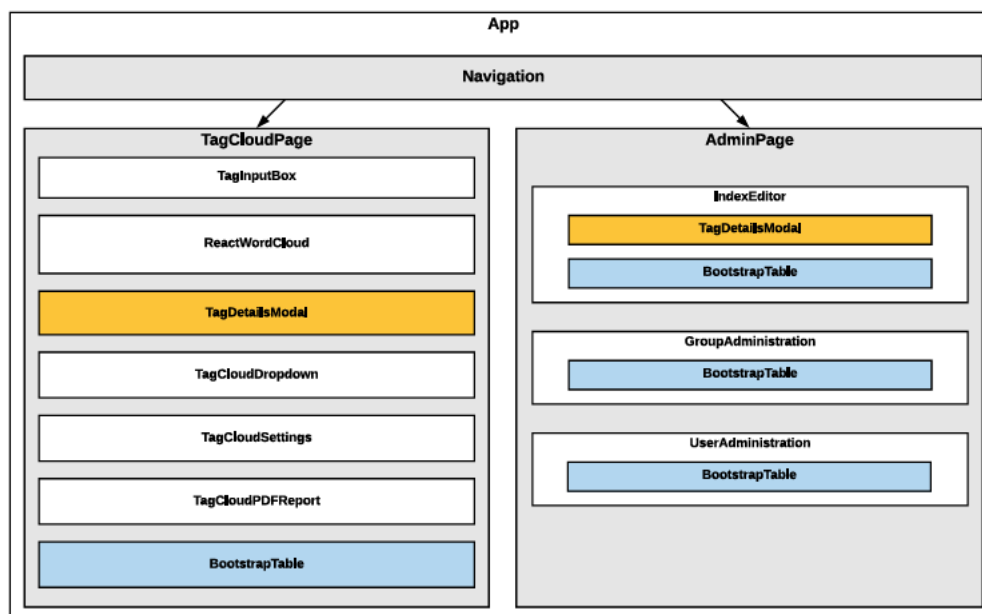


Abbildung 12: Frontend Komponenten Übersicht

In der Abbildung sind Komponenten, die an mehreren Orten eingesetzt werden, farblich markiert. Die Pfeile zwischen Navigation und den beiden Seiten, visualisieren die Router Funktion, dabei wird jeweils die Seite angezeigt, auf die der Benutzer navigiert. Dabei sei erwähnt, dass die AdminPage nur für Benutzer mit der Rolle ‘admin‘ ersichtlich ist.

Die Funktionalität der Komponenten wurde in Kapitel 5.2 bereits detailliert erläutert, weshalb hier auf eine Wiederholung verzichtet wird. Jedoch werden nun einige erwähnenswerte Informationen zu dessen Umsetzung erläutert.

5.5.4.1 Tag Cloud

Die Tag Cloud wird mittels der Erweiterung React-Wordcloud generiert. Sie erhält als Input ein Array mit den Tags und der jeweiligen Anzahl Post pro Tag. Die Tags werden mittels REST-API (Endpoint GET /tags) vom Backend abgerufen, dabei werden für die Selektion Parameter verwendet, die der Benutzer wählt. Die verschiedenen Parameter werden in den Komponenten TagInputBox, TagCloudDropdown, TagCloudSettings bestimmt. Die Komponente TagInputBox verwaltet den Tag-Typen («Arbeite an» oder «Interessiert an»). In der Komponente TagCloudDropdown wird bestimmt, ob die Tags des eingeloggten Benutzers («Persönliche Tag Cloud»), seiner Gruppe («Meine Gruppe»), einer sonstigen Gruppe oder aller Benutzer abgerufen werden sollen. Die Komponente TagCloudSettings verwaltet zwar mehrere Benutzereinstellungen, jedoch ist nur die Wahl des Zeitfensters für den API-Aufruf relevant, die übrigen Einstellungen (Farben und Anzahl Tags) werden nur clientseitig verwendet.

Für die Bestimmung der Farbe des jeweiligen Tags werden die Quantile der Anzahl Posts bestimmt. Die Anzahl Posts eines Tags wird dann mit den Quantilen verglichen und einer definierten Farbe zugeordnet, hierfür wurden JavaScript Funktionen erstellt, die über das Callback-Interface der React-Wordcloud aufgerufen werden:

Anzahl Posts \leq 25%-Quantil	Schwarz
Anzahl Posts \leq 50%-Quantil	Grün
Anzahl Posts \leq 75%-Quantil	Blau
Anzahl Posts $>$ 75%-Quantil	Rot

Die Schriftfarben lassen sich über ein Schalter-Inputfeld in der Komponente TagCloudSettings deaktivieren, um sämtliche Tags schwarz darzustellen.

Die Schriftgrösse wird von React-Wordcloud proportional zu den Anzahl Posts berechnet, die Konfiguration erlaubt jedoch die Mindest- und Maximalgrösse festzulegen. Ebenfalls unterstützt React-Wordcloud die Anzahl anzuzeigender Tags zu limitieren, das Attribute wird über den Slider in der TagCloudSettings Komponente gesteuert.

5.5.4.2 Tag Details

Die Komponente TagDetailsModal wird bei einem Klick auf ein Tag in der Tag Cloud oder der Tabelle angezeigt. Dabei werden die Meta-Informationen über die Erstellung oder Änderung des Tags angezeigt, sowie einige statistische Daten über die Verwendung

des Tags. Weiter werden sämtliche Benutzer gezeigt, die dieses Tag bereits einmal hinzugefügt haben. Somit kann anhand der Tag Details herausgefunden werden, wer an einem Thema arbeitet oder daran interessiert ist. Um die zusätzlichen Informationen abzurufen, wird ein weiterer REST-API Aufruf ausgeführt (Endpoint GET /tags/{Id}).

5.5.4.3 Administration

Der Administrationsbereich wird von der Komponente AdminPage zusammengefasst, darin sind die vier Hauptkomponenten IndexEditor, GroupAdministration und UserAdministration über drei Tabs auswählbar. Die drei Komponenten sind weitgehend gleich aufgebaut und verwenden als Kernkomponente die React-Bootstrap-Table2 Erweiterung mit diversen Plugins. Die Tabelle unterstützt viele Funktionen und ist bedürfnisentsprechend konfigurierbar. Durch ihr vielfältiges Interface lässt sie sich mit den eigenen Actions erweitern, womit bei Bearbeitungsvorgängen API-Aufrufe ausgeführt werden.

5.5.4.4 Reports

Da die Tag Cloud in React mittels der Erweiterung react-wordcloud erstellt wird, werden auch die Reports clientseitig erstellt. Dabei können Benutzer Reports im CSV- und PDF-Format erstellen, zudem haben sie die Möglichkeit, die Tag Cloud als PNG-Bilddatei herunterzuladen. Für den CSV-Report wurde das CSV-Export Plugin der Erweiterung react-bootstrap-table2 verwendet. Für das Erstellen der PDF-Reports, wird die Erweiterung react-pdf und react-pdf-table verwendet. In der Komponente TagCloudPDFReport wird der Inhalt und das Layout des PDF definiert. Da react-pdf SVG nicht unterstützt, wird die Tag Cloud von SVG zu PNG konvertiert und dann dem PDF hinzugefügt.

Aufgrund der dabei genutzten asynchronen Funktionen musste die Erstellung der PDF-Reports zweiphasig umgesetzt werden. Durch klicken auf PDF im Download-Dropdown, wird die PDF-Datei von react-pdf erstellt (erste Phase), sobald dieser Vorgang abgeschlossen ist, kann der Benutzer die Datei über einen Link herunterladen (zweite Phase). Ein Beispiel eines erstellten PDF-Reports befindet sich im Anhang H.

5.5.5 Testing

Für das Frontend wurden die Testfälle ausschliesslich manuell durchgeführt. Die Tests mussten somit bei Code Anpassungen erneut manuell durchgeführt werden, dies ist auch

einer der wesentlichen Nachteile von manuellen gegenüber automatisierten Tests. Aufgrund von fehlender Erfahrung in der Frontend Entwicklung und des knappen Zeitplans, hat sich der Autor jedoch bewusst für ein Verzicht auf ein weiteres JavaScript Framework und das automatisierte Testing entschieden. Diese Entscheidung schien im Verhältnis zu der Grösse der Applikation, der Anzahl zu testenden Anwendungsfälle und der Tatsache, dass es sich um einen Prototyp handelt, vertretbar – der Autor möchte jedoch anmerken, dass dieses Vorgehen bei einem produktiven System nicht empfehlenswert ist. Die Resultate des Testings werden im Kapitel 6.1 besprochen.

5.6 Entwicklungs- und Testumgebung

Für die Umsetzung wurde der Sourcecode Editor Visual Studio Code (VSC) verwendet. VSC ist eine Open Source Editor, der für sämtlichen Betriebssysteme (Windows, Linux, macOS) verfügbar ist. Del Sole (S. 3) nennt als Vorteile von VSC, dass diverse Programmiersprachen unterstützt werden, bereits ein Node.js Debugger integriert und das Code-Versionierungstool Git vollständig eingebunden ist. Zusätzlich sind viele Erweiterungen verfügbar, die den Entwicklungsprozess in verschiedenen Programmiersprachen und Frameworks unterstützen. Für diese Arbeit wurde insbesondere die Docker-Erweiterung verwendet. Docker (o.J) wurde für die lokale Entwicklung und um die Applikation zu testen verwendet.

Der Einsatz von Docker in der vorliegenden Arbeit ist als nicht zwingend zu betrachten, da sämtliche verwendeten Technologien wie Node.js, npm, React.js und Postgres auch auf dem Windows System, auf dem diese Arbeit umgesetzt wurde, installiert und verwendet werden können. Jedoch bietet Docker entscheidende Vorteile, die nun erläutert werden.

Der Hintergrund von Software-Containers bildet die Problematik, dass die lokale Entwicklungsumgebung der Softwareentwickler sich von der Produktionsumgebung, auf der die Software letztendlich verwendet wird, unterscheidet (Jangla, 2018, S. 1). Dadurch können lokal andere Versionen oder gar andere Betriebssysteme zum Einsatz kommen, wodurch Test Resultate in der lokalen und der produktiven Umgebung unterschiedlich ausfallen können. Um dieses Problem zu lösen, wurde das Konzept der Container, das bereits den Warenversand durch genormte Container revolutionierte, auf Software angewandt. Dabei besteht ein Container aus dem Sourcecode und denn Abhängigkeiten, die

der Sourcecode benötigt, um zu funktionieren. Der Container ist dadurch für sich isoliert funktionsfähig, unabhängig von dem System (z.B. Betriebssystem), das den Container ausführt. Dadurch werden die Systemanforderungen, die eine Applikation bzw. der Sourcecode hat, weg abstrahiert (Jangla, 2018, S. 2). Ein weiterer Vorteil ist, dass Containers in der Regel kleiner sind, als die vollständige Installation der benötigten Abhängigkeiten, im Falle dieser Arbeit etwa die Installation von Postgres unter Windows.

Jangla (2018, S. 11) beschreibt Docker als ein Tool, das es einfach macht, Applikationen in Container zu installieren und auszuführen. Durch die Abstrahierung von Systemabhängigkeiten, können sich Entwickler auf die Anforderungen der Applikation selbst konzentrieren und erhalten so einen besseren Überblick. Zudem nennt er den Vorteil, dass Docker sehr geringe Systemanforderung hat und das Ausführen von Applikationen nur wenig Arbeitsspeicher benötigt. Zudem lassen sich Container innerhalb von kürzester Zeit starten und wieder beenden, wodurch Systemanforderungen oder Konfigurationen in kurzer Zeit geändert und angewandt werden können.

Ein Konzept von Container ist, dass jeder Service seinen eigenen Container hat. Da moderne Applikationen, wie auch der Prototyp in dieser Arbeit, häufig aus mehreren Services bestehen, müssen die verschiedenen Container aufeinander abgestimmt werden (Jangla, 2018, S. 2). Docker bietet diese Funktion mit dem Tool Docker-Compose. Es ermöglicht es, mehrere Container mittels einem Command auszuführen, dazu wird ein YAML-File erstellt, das die benötigten Container miteinander verknüpft. Der Prototyp dieser Arbeit besteht aus den drei Services REST-API, Postgres Datenbank und React Frontend. Durch die Definition im docker-compose.yml File können alle drei Services bzw. die zugehörigen Container mit einem Befehl gestartet und beendet werden. Während für die Erstellung der Container für die Services REST-API und React Frontend jeweils ein individuell angepasstes Dockerfile erstellt werden musste, konnte für die Datenbank das offizielle postgres Image von Docker-Hub (Library mit veröffentlichten Docker Images) verwendet werden. Das Postgres Image kann über Environment Variablen im YAML-File direkt konfiguriert werden, so dass beim Start automatisch eine Datenbank mit den verwendeten Angaben erstellt wird. Dadurch ist für die Entwicklung weder eine lokale Installation von Postgres notwendig, noch muss die Datenbank und ein zugehöriger Benutzer manuell erstellt werden. Um Daten auch nach dem Beenden von Container

nicht zu verlieren, wurden Docker Volumes verwendet. Sie ermöglichen es, das Filesystem des Containers mit dem Host-System (im der vorliegenden Arbeit Windows) zu verlinken.

Da Sequelize verwendet wurde (vgl. Kapitel 5.4.1), konnte der Startbefehl des REST-APIs so gestaltet werden, dass wenn die Tabellen in der Datenbank noch nicht angelegt sind, die ausstehenden Migrations (vgl. Kapitel 5.4.1) ausgeführt und die initialen Datensätze (Administrator Benutzer und Initiale Gruppe) automatisch eingefügt werden. Die Applikation lässt sich dadurch mit einem einzigen Befehl installieren und starten (vorausgesetzt Docker ist installiert), ohne dass zusätzliche Software installiert oder manuelle Schritte ausgeführt werden.

Um die Entwicklung zu vereinfachen, wurden Erweiterungen der jeweiligen Frameworks und Konfigurationen verwendet, die Code-Änderungen auf dem Host-System direkt in den gestarteten Container übernehmen. Dazu wurde unter anderem das Verzeichnis des Source-Codes direkt mit dem Dateisystem des Containers verknüpft. Da dies zur Folge hat, dass unter Umständen lokale (möglicherweise systemabhängige) Erweiterungen in den Container gelangen, wurde für das Testen ein zweites YAML-File erstellt, dass diese Abhängigkeit auflöst und somit vollständig eigenständig ist. Damit ist sichergestellt, dass der Sourcecode vollständig Host-System unabhängig funktionsfähig ist.

6 Validierung

Hevner et al. (2004) nennen die Evaluation von IT-Artefakten als einen entscheidenden Aspekt der IS Forschung. Dabei sollen die erstellten Artefakte auf Nützlichkeit, Qualität und Effizienz geprüft werden. Zudem können qualitative Aspekte wie die Funktionalität, Vollständigkeit, Konsistenz, Genauigkeit, Zuverlässigkeit und wie gut das Artefakt in eine Organisation passt, untersucht werden (Hevner et al., 2004). In dieser Arbeit wurde nebst einer technischen Validierung mittels Black-Box-Testing ein User-Acceptance-Testing (UAT) mit elf Probanden durchgeführt. Die beiden Validierungsprozesse und die zugehörigen Resultate werden in den nachfolgenden Kapiteln erläutert.

6.1 Technische Validierung

Bei der technischen Validierung wurde geprüft, ob der Prototyp die in Kapitel 4.3 definierten Anforderungen erfüllt. Dazu wurde die Black-Box-Testmethode, auch behavioral oder functional Testing genannt, verwendet. Dabei werden Testfälle basierend auf den Anforderungen definiert (Laski und Stanley, 2009, S. 82). Laski und Stanley (2009, S. 82) nennen Intuition, Erfahrung, Problem- und Lösungsverständnis als entscheidende Faktoren beim Black-Box-Testing. Die Vorteile sind, dass nebst der Korrektheit des Systems auch die Anforderung geprüft wird, denn die Testfälle und die zu erwartenden Ergebnisse werden aufgrund der Anforderung definiert, Unstimmigkeiten bei der Ausführung können dann sowohl der Implementation des Systems oder auch der unklaren Formulierung der Anforderung geschuldet sein. Beim Black-Box-Testing bestehen Testfälle typischerweise aus dem zu testenden Szenario, bestehend aus der Beschreibung der Funktionalität und des einleitenden Systemzustands sowie dem zu erwartenden Ergebnis. Der wohl schwerwiegendste Nachteil ist, dass sich Black-Box-Tests nur schwer automatisieren lassen. Der für das in dieser Arbeit durchgeführte Black-Box-Testing verwendete Testplan mit den Testfällen befindet sich im Anhang A. Durch die Tatsache, dass der Programmierer und Tester in dieser Arbeit dieselbe Person waren, konnten bei durchgeführten Systemanpassungen, jeweils die Testfälle durchgeführt werden, die für die Anpassung relevant waren, dabei wurde iterativ vorgegangen, wobei auf Anpassungen jeweils unmittelbar Tests durchgeführt wurden.

Das Black-Box-Testing wurde auf der in Kapitel 5.6 beschriebenen lokalen Testumgebung durchgeführt. Im nächsten Abschnitt werden die Resultate beschrieben.

6.1.1 Resultate

Während der Entwicklung wurden zahlreiche kleinere Fehler im System gefunden und behoben, auf die hier nicht weiter eingegangen wird. Bei der abschliessenden Validierung vor dem User-Acceptance-Testing wurden nochmals sämtliche Testfälle ausgeführt, die dabei gefundenen Probleme und Lösungsansätze werden nun erläutert.

1. Unsichtbare Tags

Testfall: 4a-d, 6a-d

Beschreibung: Bei der Verwendung der Einstellungen, hinzufügen von neuen Tags oder Wechseln zwischen den verschiedenen Tag Clouds, werden gewisse Tags unsichtbar dargestellt. Dies führt zu einer wesentlichen Verschlechterung des Nutzungserlebnisses und verwirrt bei Auswertungen.

Lösung: Die Fehleranalyse zeigte, dass die Tags durch React-Wordcloud zwar zur Tag Cloud hinzugefügt, jedoch unsichtbar (`fill-opacity=0`) dargestellt werden. Als schnelle Lösung wurde die Funktion `displayHiddenTags()` in der React Komponente `TagCloudPage` implementiert, um die unsichtbaren Tags nachträglich manuell anzuzeigen. Die Lösung war zwar eine Verbesserung jedoch nicht zufriedenstellend, da sie nicht in jeder Situation zur Lösung des Problems führte. Zu einem späteren Zeitpunkt konnte die Implementation der ersten Lösung verbessert werden und die Tag Cloud wurde in der Folge zuverlässig und stabil angezeigt.

2. Abbruch von Operationen im Administrationsbereich

Testfall: 9a, 9b, 10b

Beschreibung: Wenn Operationen im Index-Editor und der Gruppenverwaltung im Popup abgelehnt werden, wird der Wert in der Tabelle nicht auf den Ursprungswert zurückgesetzt.

Lösung: Bei der Fehleranalyse wurde festgestellt, dass dieses Verhalten nicht konsistent ist. Während das Abbrechen in der Gruppenverwaltung wie erwartet funktioniert (selbe Implementation), tritt das Problem im Index-Editor und der Gruppenverwaltung bei der ersten Operation auf. Bis zur Abgabe dieser Arbeit, konnte für dieses Verhalten keine Lösung gefunden werden. Der Autor geht davon aus, dass es sich um einen Fehler in der Erweiterung react-bootstrap-table handelt, wenn eine Asynchrone Funktion für die Speicherung der Änderung genutzt wird. Die Operation wird bei Abbruch datenbankseitig nicht ausgeführt, clientseitig wird jedoch nicht der ursprüngliche Wert angezeigt.

3. Sortierung nach Anzahl Posts

Testfall: 4a-d, 6a-d

Beschreibung: Die Sortierung der Tabelle ist für Tags mit mehr als 10 Posts falsch.

Lösung: Die Erweiterung React-Bootstrap-Table wandelt numerische Strings nicht automatisch um, weshalb die Sortierung für numerische Werte dann nicht korrekt funktioniert. Anzahl Posts wurde mit der Funktion `parseInt()` umgewandelt um das Problem zu lösen.

4. Auswahl von Datumsbereich in der Zukunft

Testfall: 6c

Beschreibung: Im Fenster für die Auswahl des Datumsbereich in den Tag Cloud Einstellungen können zukünftige Daten ausgewählt werden.

Lösung: Implementation wurde so angepasst, dass zukünftige Daten nicht mehr ausgewählt werden können.

Durch die vollständige Durchführung aller Testfälle sowie den getroffenen Massnahmen, konnte die Applikation für das User-Acceptance-Testing freigegeben werden. Die gemäss

den Anforderungen geforderten Datenerhebungen und Auswertungen sind im erstellten Prototyp möglich. Bis auf das inkonsistente Verhalten im Administrationsbereich konnten sämtliche gefundenen Probleme behoben werden.

6.2 Validierung mittels User Acceptance Testing

Beim UAT untersuchen spätere Benutzer, ob das System wie erwartet funktioniert und für die spätere produktive Verwendung taugt. In dieser Arbeit wurden den Probanden dafür verschiedene Aufgaben gestellt und ausgewertet. Leung und Wong (1997) beschreiben dies als verhaltensbasierten bzw. szenariobasierten UAT Ansatz, bei dem das System aus der Benutzerperspektive getestet wird. Da bei der Auswertung des UAT nebst der Funktionalität (sind die Aufgaben überhaupt lösbar?) auch Usability Aspekte wie Nützlichkeit oder Brauchbarkeit erhoben wurden, entsteht ein enger Zusammenhang mit Usability Testing. Beim Usability Testing werden Benutzern möglichst realistische Aufgaben gestellt und es wird beobachtet, wie gut sie diese lösen können (Moser, 2012, S. 220). Während funktionale Tests die Frage beantworten, ob ein System wie erwartet funktioniert (UAT), beantworten Usability Tests die Frage, wie gut ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen (Moser, 2012, S. 220). Im Kontext der von Hevner et al. (2004) vorgeschlagenen Richtlinien der IS Forschung handelt es sich dabei um eine experimentelle Evaluation, die beantworten soll, ob mit dem entstandenen IT-Artefakt das definierte Problem gelöst werden kann.

In den folgenden Abschnitten wird auf die Rahmenbedingungen und Resultate des User-Acceptance-Testings eingegangen.

6.2.1 Rahmenbedingungen

Das UAT wurde aufgrund der Umweltsituation und den geltenden Richtlinien des Bundes (Eidgenössischer Bundesrat, 2020) während der Dauer dieser Arbeit ohne physischen Kontakt zu den Probanden durchgeführt. Tullis et al. (2002) untersuchten, in wie fern sich die Ergebnisse von Usability Tests, die in einer Laborumgebung stattfanden, von solchen ohne physische Anwesenheit unterscheiden und konnten keine erheblichen Nachteile feststellen. Durch die fehlende Möglichkeit, die Benutzer während der Durchführung zu

beobachtet, können jedoch gewisse Usability Probleme (z.B. exzessives Scrollen auf der Webseite oder Probleme Elemente direkt zu finden) nicht unmittelbar erfasst werden (Tullis et al., 2002). Diese Limitation gilt auch für das UAT dieser Arbeit.

6.2.1.1 Teilnehmer

Für das UAT wurden elf Probanden aus dem Umfeld des Autors rekrutiert. Von den elf Teilnehmer gab lediglich eine Person an, bereits vor der vorliegenden Arbeit einmal in Kontakt mit Tag Clouds gekommen zu sein. Zudem mussten die Kandidaten ihre Kenntnisse betreffend IT-Applikationen auf einer fünf Punkte Skala einschätzen (5 – sehr gute Kenntnisse), wobei die Antworten von eins bis fünf reichten (Arithmetisches Mittel 2.55).

6.2.1.2 Testumgebung

Um den Prototyp den Probanden zur Verfügung zu stellen, wurde dieser auf der Cloud-Plattform Heroku (o.J.) installiert. Die Applikation war daher im World-Wide-Web über eine URL verfügbar und die Probanden konnten die ihnen gestellten Aufgaben an einem Endgerät, Standort und Zeitpunkt ihrer Wahl durchführen. Der Fragebogen wurde mit dem von der Zürcher Hochschule zur Verfügung gestellten Online-Umfrage-Tool LimeSurvey realisiert.

Um den Probanden eine möglichst realitätsnahe Testumgebung zur Verfügung zu stellen, wurden der Prototyp mit Testdaten initialisiert. Dazu wurden 80 Tags, dreizehn Benutzer (ein Administrator Benutzer, dessen Anmeldedaten den Probanden bekannt gegeben wurden) und drei Gruppen hinzugefügt. Abbildung 13 zeigt die Tag Cloud, die im Prototyp in der Testumgebung angezeigt wurde. Sämtliche Probanden benutzten dieselbe Applikation und Datenbank, die Daten wurden einmalig aufbereitet und nicht für jeden Probanden neu initialisiert. Der Autor dieser Arbeit hat die Datengrundlage während des Testzeitraums regelmässig überprüft, um sicherzustellen, dass sich die Ausgangslage zwischen den Probanden nicht zu stark verändert.

4	Tags finden	Tags in der persönlichen und der Tag Cloud der zugehörigen Gruppe finden. Ein bestimmtes Tag in der Tag Cloud finden, mit deaktivierter Tag Einfärbung.	A5, B1
5	Auswerten	Verschiedene Auswertungen vornehmen mit Hilfe der Tag Cloud. Die Probanden können die Einstellungen verwenden, die sie bevorzugen. Die Aufgabe umfasst sowohl spezifische Tag- und Informationssuche als auch das Erkennen von Trends (populäre Tags).	A6, A7, A8,
6	Administration	Den Administrationsbereich verwenden um Tags/Gruppen hinzuzufügen, zu ändern und zu löschen. Einen Benutzer einer Gruppe zuweisen.	A3, A9, B1, B2, B3, B4

6.2.1.4 Fragebogen

Der Fragebogen wurde in vier Teile gegliedert, wobei im ersten Teil Fragen zum Benutzer erhoben wurden, dabei wurde erhoben, welche Probanden schon einmal eine Tag Cloud verwendet haben und wie die Kenntnisse betreffend IT-Applikationen eingeschätzt werden. Im zweiten Teil wurde befragt, welche Aufgaben-Kategorien gelöst werden konnten und wie anspruchsvoll das Lösen empfunden wurde. Sofern eine Aufgabe nicht gelöst werden konnte, wurden die aufgetretenen Probleme erhoben. Abschliessend wurde befragt, was bei der Bearbeitung der Aufgaben verhältnismässig viel Zeit in Anspruch genommen hat. In diesem Teil wurden bis auf die letzte Frage primär die funktionalen Aspekte befragt, der dritte Teil fokussierte sich auf die Usability der Tag Cloud als Visualisierungstool. Dabei mussten die Probanden ihr Eindruck betreffend Nützlichkeit und Eignung der Tag Cloud und des Prototyps als Ganzes bewerten. Zudem wurde befragt, wie viele Tags in einer Tag Cloud angezeigt werden sollen, damit diese die volle Aussagekraft entfaltet. Im letzten Teil wurde erhoben, ob die Probanden sich vorstellen könnten, den Prototyp oder eine ähnliche Applikation in einem Team einzusetzen und ob diese hilfreich sein könnte, um bessere Kenntnisse über die Tätigkeiten und Interessen der Teamkollegen zu erlangen. Der vollständige Fragebogen befindet sich im Anhang C.

6.2.1.5 Ablauf

Den Probanden wurde die Aufgabenstellung, die URL zum Online-Prototyp sowie zur Online-Umfrage via E-Mail zugestellt. Zudem wurden einige wesentlichen Informationen zum Kontext dieser Arbeit übermittelt, jedoch ohne Hilfestellungen für die Verwendung des Prototyps oder zur Interpretation der Tag Cloud zu liefern.

6.2.2 Resultate

In der Folge werden die Resultate des UAT ausgewertet, dabei wird zuerst auf die Funktionalen und Usability Aspekte des Prototyps eingegangen, darauf folgt die Auswertung der Antworten betreffend Usability des Konzept Tag Cloud und deren Verwendung in Teams zur Unterstützung von TMS.

6.2.2.1 Funktionale Aspekte

Die Auswertung zeigt, dass 9 der 11 Probanden sämtliche Aufgaben lösen konnten. Wobei die Aufgaben 3 und 4 jeweils von einer Person und die Aufgabe 6 von zwei Personen nicht gelöst werden konnten. Eine Person konnte die Aufgaben 3, 4 und 6 nicht lösen, eine weitere Person konnte die Aufgabe 6 nicht lösen. Die beiden Personen schätzten ihre Kenntnisse betreffend IT-Applikationen als gering und sehr gering ein. Sie nannten die in der Tabelle 9 gelisteten Problem bei den jeweiligen Aufgaben:

Tabelle 9: Aufgetretene Probleme pro Aufgabe und Proband

Aufgabe	Proband (ID)	Genannte Probleme
3	4	Schwierigkeiten beim Erfassen eines Tags, das Tag wurde folglich in der Tag Cloud nicht angezeigt.
4	4	In der persönlichen Tag Clouds wurden keine Tags angezeigt, in der Tag Cloud der Gruppe wurden jedoch Tags angezeigt (selbe Person wie oben).

6	4	Konnte den Benutzer keiner anderen Gruppe zuweisen, hat in dem Bereich «Gruppenadministration» nach der Funktion gesucht.
	11	Konnte den Benutzer keiner anderen Gruppe zuweisen. Konnte das Tag im Index-Editor nicht ändern.

Durchschnittlich wurden die Aufgaben 1 und 2 als am einfachsten und die Aufgabe 5 am anspruchsvollsten wahrgenommen. Die Aufgaben 3 und 6 zeigte die grösste Streuung der Antworten betreffend wahrgenommene Komplexität auf der fünf Punkte Skala (1 sehr einfach - 5 sehr anspruchsvoll/komplex), das Erfassen der Tags sowie die Aufgaben im Administrationsbereich wurde von den Personen von sehr einfach bis sehr anspruchsvoll wahrgenommen. Das Auswerten von Tags im Prototyp wurde von einer Person als sehr einfach, von sechs Personen als ziemlich einfach und von jeweils zwei Personen als weder einfach noch anspruchsvoll sowie relativ anspruchsvoll wahrgenommen. Das in Abbildung 14 gezeigte Balkendiagramm fasst die durchschnittlich wahrgenommene Komplexität der Aufgaben auf der fünf Punkte Skala zusammen.

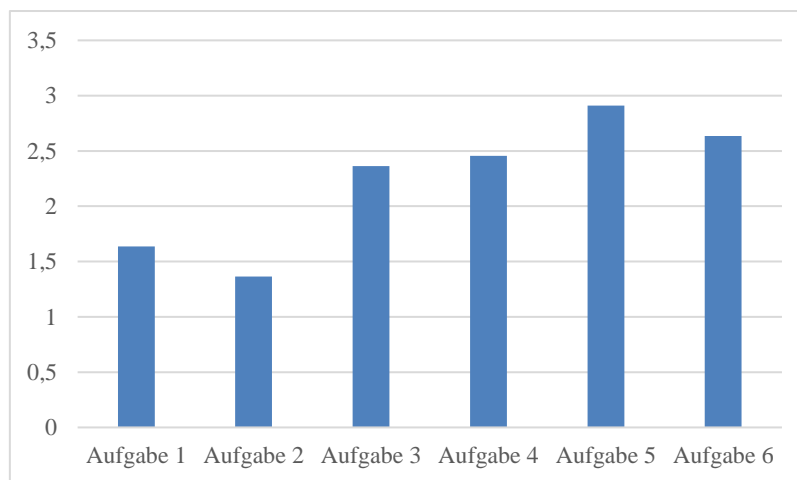


Abbildung 14: Durchschnittlich wahrgenommene Komplexität pro Aufgabe

Aufgrund dessen gaben auch gleich mehrere Probanden an, dass sie für die Auswertungen in Aufgabe 5 verhältnismässig viel Zeit benötigten. Eine Person merkte an, dass ohne die Suchfunktion (CTRL+F) für das Finden von wenig populären Tags viel Zeit benötigt wird. Mehrere Probanden fügten an, dass sie etwas Zeit benötigten, bis sie das Bedienungskonzept der Hauptseite mit den zwei Dropdowns verstanden hatten. Eine Person,

die die Aufgaben auf einem iPad mittels Safari-Browser bearbeitete, hatte zudem angemerkt, dass die Applikation nicht immer korrekt reagiert habe. Ein ähnliches Verhalten hatte eine weitere Person festgestellt, bei der die angezeigte Anzahl Tags beim Slider nicht der effektiven Anzahl Tags in der Tag Cloud entsprach.

6.2.2.2 Usability Aspekte

Die Tag Cloud wurde von den Probanden durchaus für interpretier- und auswertbar empfunden. Auf einer Skala von 1 (umständlich / nicht hilfreich) bis 5 (hilfreich / nützlich) vergaben drei Personen die Wertung 3, sechs Personen die Wertung 4 und zwei Personen die maximale Wertung 5. Die Einfärbung der Tags wurde von vier Probanden als hilfreich und von den restlichen sieben Probanden als sehr hilfreich empfunden. Auf die Frage wie viele Tags in der Tag Cloud angezeigt werden sollen, damit diese die volle Aussagekraft entfaltet, wählten die meisten Probanden den Bereich von 21 bis 30 Tags. Keine Person würde mehr als 40 Tags anzeigen. In der Abbildung 15 wird die Verteilung der Antworten auf die zuvor genannte Frage gezeigt.

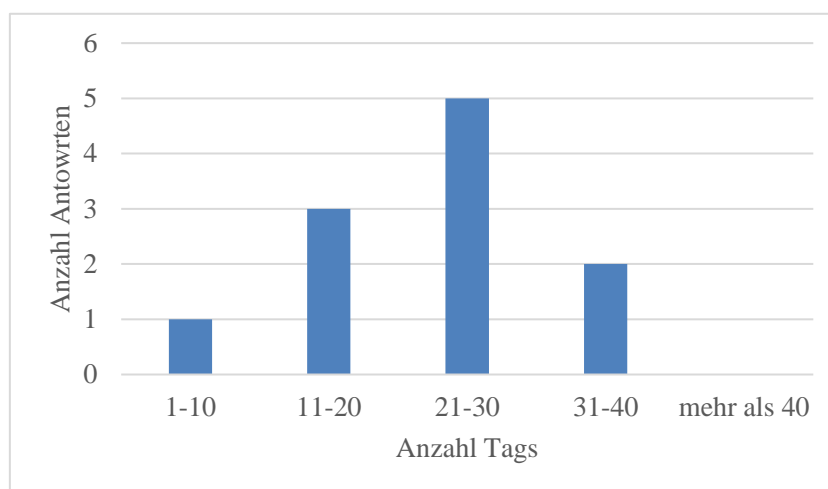


Abbildung 15: Wie viele Tags sollen angezeigt werden?

Während eine Person befand, dass der Prototyp nicht hilfreich ist oder sein könnte, waren die restlichen zehn Personen gegenteiliger Meinung. Die Person begründete die Antwort damit, dass sie sich kein Einsatzgebiet einer solchen Applikation vorstellen könne und sieht entsprechend auch keinen Nutzen. Die übrigen zehn begründeten ihre Antwort unter anderem wie folgt:

- «Ich denke, dass es gerade für ein grösseres Team nützlich sein kann, so schnell auf einen gemeinsamen Nenner zu kommen, besonders dann, wenn es zu viel Zeit kosten würde dies mit allen persönlich zu machen.»
- «Man könnte das Tool Departement übergreifend oder auch komplett firmenintern verwenden für gewisse Auswertungen ob nach Interesse, Aufgabenbereiche, Mitarbeiterzufriedenheit etc.»
- «Auf kreative Weise sehen, wer an was arbeitet oder Interesse hat.»
- «Für das zusammentragen und das bessere Verständnis von Personen Gruppen»
- «Hilft, um zu priorisieren, strukturieren. Übersichtlichkeit bei einer Vielzahl von Anwendern.»
- «Innerhalb grosser Teams können Personen ermittelt werden, die sich mit dem im Tag beschriebenen Thema auskennen.»
- «Übersichtlich, auf den ersten Blick erkennbar welches Tag im Vordergrund steht.»

Ebenfalls zehn Probanden könnten sich vorstellen, diese oder eine ähnliche Applikation, die Tag Clouds für die Datenpräsentation verwenden, in einem Team einzusetzen. Sie argumentierten dies damit, dass die Tag Cloud eine schnelle Übersicht bietet und Tags einfach und klar darstellt. Eine Person könnte sich vorstellen, Tag Clouds für Präsentationen über Interessen und Aufgaben zu verwenden. Eine andere Person findet jedes Tool, welches dabei hilft, Leute zu verbinden und dazu noch kreativ ist, hilfreich. Zweifach wurde genannt, dass eine solche Applikation dabei helfen könnte, Gemeinsamkeiten und Überschneidungen zu erkennen, dies könnte in Teams genutzt werden. Die eine Person, die bereits den Prototyp für nicht hilfreich befand, kann sich folglich auch keinen Einsatz in einem Team vorstellen, sie sehe darin keinen Nutzen.

Von den zehn Probanden, die eine solche Applikation als hilfreich befanden, denken sechs, dass sie dadurch definitiv bessere Kenntnisse darüber erlangen, an was ihre Kollegen arbeiten bzw. wofür sie sich interessieren. Die übrigen vier sind sich ziemlich sicher, dass dem so ist.

Zudem haben vier Probanden zusätzliche Anmerkungen und Verbesserungsvorschläge gemacht, dabei hat eine Person, das Erfassen von Tags als für zu umständlich befunden, da bei neuen Tags zuerst ein Klick auf das Dropdown notwendig ist, sie schlägt vor, dass neue Tags auch ohne zusätzliche Eingaben hinzugefügt werden können. Weiter schlägt sie vor, dass wenn ein Tag ausgewählt wurde, dies visuell ersichtlich gemacht wird im

Stil von GitHub-Tags. Weiter wurde angemerkt, dass die Bearbeitung von Tags im Tag-Index zu wenig intuitiv sei. Diese Person schlägt vor, dass mit einem zusätzlichen Bearbeitungs-Button der Bearbeitungs-Modus aktiviert werden soll. Eine Person schlägt vor, dass Tags nach Kontext gruppiert werden sollen. Zudem fügt eine Person an, dass die Auswahlelemente auf der Hauptseite besser beschrieben oder erklärt sein könnten.

7 Diskussion

In diesem Kapitel werden die Arbeitsergebnisse der vorliegenden Arbeit diskutiert. Dazu werden zuerst die Resultate der Validierung interpretiert, danach werden die erlangten Erkenntnisse genutzt, um die erarbeitete Lösung kritisch zu hinterfragen.

7.1 Beurteilung der Resultate

Mit dem entwickelten Prototyp konnten neun Probanden sämtliche Aufgaben lösen. Aufgrund dieser Tatsache sowie den Resultaten aus der technischen Validierung, wurde gezeigt, dass der Prototyp die formulierten Anforderungen erfüllt und die Erfassung, Darstellung und Auswertung der Meta-Informationen «who does what» und «who is interested in what» in Teams grundsätzlich möglich sind. Dass eine Person jedoch gleich mehrere Aufgaben nicht lösen konnte, weist darauf hin, dass Usability Probleme bestehen, das heisst, dass die Bedienung des Prototyps gerade für Benutzer mit wenig IT-Kenntnissen zu wenig selbsterklärend ist. Mehrere Probanden berichteten zudem von teils instabilem Verhalten der Tag Cloud, dies konnte bereits beim technischen Validieren festgestellt und erst nach der Durchführung des UAT behoben werden (vgl. Kapitel 6.1.1). Durch die Probanden wurden weitere Usability Probleme im Prototyp erkannt und teils auch durch Verbesserungsvorschläge ergänzt, welche sich wie folgt zusammenfassen lassen:

- **Tag erfassen:** Das Erfassen soll ohne zusätzliche Benutzereingaben möglich sein. Ein visuelles Feedback, dass ein Tag ausgewählt wurde, könnte weiter unterstützen (nachträglich implementiert).
- **Administration:** Das Bearbeiten in den Tabellen im Administrationsbereich ist zu wenig intuitiv, der Bearbeitungsmodus sollte durch einen Bearbeitungs-Button aktiviert werden. Das Zuweisen von Benutzern zu Gruppen ist zu wenig selbsterklärend.
- **Tags gruppieren:** Um das Suchen von Tags zu vereinfachen, sollen die Tags nach Kontext gruppiert werden.
- **Tags benennen:** Tags sind von der Schreibweise abhängig, durch unterschiedliche Schreibweise könnten Duplikate entstehen, die das Erfassen und Suchen von Tags erschweren.

- **Dropdowns auf Hauptseite:** Die beiden Dropdowns auf der Hauptseite sind ohne Erläuterung zu wenig selbsterklärend, um das Navigations-Konzept zu verstehen.
- **Login / Registrierung:** Die Seiten für Login und Registrierung sollten mit Titel versehen werden, um Verwechslungen vorzubeugen (nachträglich implementiert).

Die Vorschläge fließen in das nachfolgende Kapitel ein.

Die Einfärbung der Tags wurde als nützlich empfunden. Dieses Ergebnis deckt sich mit den Erkenntnissen von Bateman et al. (2008), die in ihrer Studie die Tag Farbe als ein Merkmal identifizierten, dass von Betrachtern einfach erkannt wird, sie konnten jedoch nicht feststellen, in wie fern und welche Farben die Aufmerksamkeit besonders auf sich ziehen. Die meisten Probanden bevorzugten eine Tag Cloud mit 21 bis 30 Tags, in welchem Bereich die optimale Anzahl Tags liegt, scheint jedoch noch ungeklärt. Es ist anzunehmen, dass dies stark von der Aufgabe, z.B. suchen oder browsen (vgl. Kapitel 3.4) sowie den persönlichen Präferenzen des Benutzers abhängt. Khusro et al. (2018) empfehlen zwar die Tag Clouds mehrschichtig zu gestalten, und pro Schicht nur eine limitierte Anzahl von Tags anzuzeigen, machen jedoch keine Empfehlung wie viele Tags pro Schicht angezeigt werden sollen. Der Vorschlag eines Probanden, Tags nach Kontext zu gruppieren, steht im Einklang mit der Empfehlung von Deutsch et al. (2011).

Bis auf eine Person waren sich alle Probanden einig, dass der Prototyp hilfreich ist oder sein kann. Sie können sich in der Folge auch vorstellen, diese oder eine ähnliche Applikation in einem Team einzusetzen. Sie heben dabei die Stärke der Tag Cloud hervor, Daten übersichtlich und zusammenfassend darzustellen. Zudem schätzten einige Probanden die kreativen Eigenschaften der Tag Cloud. Die Tag Cloud ist für die Probanden ein geeignetes und akzeptiertes Visualisierungsmittel um die Meta-Informationen «who does what» und «who is interested in what» in Teams darzustellen. Dass damit Auswertungen als anspruchsvoll wahrgenommen wurden, stimmt mit den Resultaten von Sinclair und Cardew-Hall (2008) überein, sie nennen das Browsen oder die unspezifische Informationssuche als Stärke und erkennen Einschränkungen bei der spezifischen Informationssuche mittels der Tag Cloud. Auswertungen mittels Tag Cloud scheinen grundsätzlich möglich, jedoch wie bereits von Cardew-Hall (2008) empfohlen, sollen auch weitere Möglichkeiten (z.B. Begriffssuche) bestehen, um die spezifische Informationssuche zu unterstützen und den Schwächen der Tag Cloud entgegenzuwirken.

Das Erlangen eines besseren Verständnisses über Personengruppen sowie das rasche erkennen von Trends (populäre Tätigkeiten und Interessen) stellt für die Probanden den Hauptnutzen dar. Zwei Personen sind der Ansicht, dass der Prototyp besonders dann nützlich ist, wenn der persönliche Austausch in grossen Personengruppen schwierig oder ineffizient wäre. Damit scheint auch die Daseinsberechtigung von Informationssystemen zur Unterstützung von TMS, die auch Jackson (2011) durch das Wachstum und die Globalisierung von Organisationen sieht, gegeben. Ebenfalls zwei Probanden nannten, dass der Prototyp unterstützen könnte, um Personen zu finden, die sich mit einem Thema auskennen oder Personen mit denselben Interessen zu verbinden. Alle Probanden, die sich vorstellen könne, eine solche Applikation in einem Team einzusetzen, teilen die Ansicht, dass dies dabei helfen kann, bessere Erkenntnisse darüber zu erlangen, an was ihre Teamkollegen arbeiten und wofür sie sich interessieren. Es ist davon auszugehen, dass der entstandene Prototyp und die damit verbundenen Konzepte, TMS in Teams unterstützen könnte. Der Funktionsumfang entspricht weitgehend den Forderungen von Yuan et al. (2007) und Wang et al. (2018).

Um abschliessend zu beantworten, wie ein Informationssystem zur Unterstützung von TMS in Teams die Akzeptanz der Benutzer erlangt, müssten weiter Untersuchungen gemacht werden. Die vorliegende Arbeit macht mit dem Hilfsmittel Tag Cloud zwar einen kreativen Vorschlag zur Gestaltung eines solchen Systems, ob und wie ein solches in der Praxis durch echte Teams akzeptiert wird und ob es effektiv einen positiven Einfluss auf das TMS hat, kann mit der in dieser Arbeit durchgeführten Validierung nicht beantwortet werden.

Dass nicht sämtliche Personen einen Nutzen in einer solchen Applikation sehen, deutet darauf hin, dass der Kontext, z.B. die Art von Team, Tätigkeit, und Zusammenarbeit, einen wesentlichen Einfluss darauf hat, ob einen Nutzen entsteht oder nicht. Da TMS laut Lewis (2003) besonders in Teams, die wissensintensiven Arbeiten nachgehen, wirkungsvoll ist, lässt vermuten, dass dies entsprechend auch für Informationssysteme zur Unterstützung von TMS gilt.

7.2 Weiterentwicklung und alternative Lösungsmöglichkeiten

Der Prototyp erfüllt einige Charaktereigenschaften des von Jackson und Klobas (2008) vorgeschlagenen TMS Informationssystem. Es besteht ein Verzeichnis mit Labels in der

Form von Tags, diese weisen jedoch keine Verknüpfungen zu weiteren Wissensquellen auf, auch Teile der zweiten Basis-Komponente, die die Möglichkeit der Verwaltung des Verzeichnisses vorsieht, sind durch den Administrationsbereich abgedeckt. Die dritte Komponente, das Abrufen und Suchen von Informationen im Verzeichnis, wurde mittels der Hauptseite und insbesondere der Tag Cloud umgesetzt. Die vierte und letzte Komponente, die Zuweisung von Verantwortlichkeit zu Wissensbereichen, fand im Prototyp jedoch keine Anwendung. Der Prototyp steht damit wohl den von Lehner (2019, S. 300) definierten Groupware Systemen näher als einem TMS Informationssystem nach Jackson und Klobas (2008), auch wenn gewisse Überschneidungen zu erkennen sind. Es wäre denkbar, dass aufgrund des in dieser Arbeit erstellten Prototyps und den zugehörigen Konzepten weitere Komponenten des von Jackson und Klobas vorgeschlagenen TMS Informationssystem implementiert werden, so können etwa externe Systeme angebunden werden. Die Resultate dieser Arbeit zeigen, dass die Tag Cloud im Kontext von TMS in Teams akzeptiert wird, woraus spekuliert werden kann, dass sie sich auch für die Darstellung der Labels in einem umfänglichen TMS Informationssystem eignet.

Um die Usability für Benutzer mit wenig IT-Kenntnissen zu verbessern, wäre es denkbar, dass neue Benutzer bei der ersten Anmeldung durch ein interaktives Tutorial durch die Applikation geführt werden, dabei werden die verschiedenen Funktionen und Elemente der Applikation demonstriert. Dadurch könnte sichergestellt werden, dass die Benutzer das Nutzungskonzept verstehen, was in einem engen Zusammenhang mit dem empfundenen Mehrwert steht.

Tags mit derselben Bedeutung aber unterschiedlicher Schreibweise haben einen negativen Einfluss auf die Resultate der Auswertungen mittels der Tag Cloud. Es wäre denkbar, dass im Index-Editor Duplikate, also Tags die einen grossen Übereinstimmungsgrad aufweisen, angezeigt und den Administratoren ermöglicht wird, solche Tags zu vereinen. Durch das Gruppieren von Tags, die im selben Kontext stehen, wäre auch die von Khusro et al. (2018) empfohlene, mehrschichtige Umsetzung, der Tag Cloud möglich. Dazu müssten Clustering Algorithmen aus dem Bereich des Machine Learnings verwendet werden. Für eine solche fortgeschrittene Tag Cloud, müsste die Generierung der Tag Cloud, im Gegensatz zum umgesetzten Prototyp in dieser Arbeit, serverseitig geschehen.

Eine weitere Möglichkeit um die Interaktivität und den Austausch der Benutzer zu fördern, wäre es, die Tag Clouds von Benutzern auch direkt in der Tag Details Ansicht an-

zuzeigen, so könnten sich Benutzer direkt einen Eindruck über andere Benutzer verschaffen, welche an einem gewissen Tag arbeiten oder sich dafür interessieren. Zudem wurde die Benutzer- und Gruppenadministration als für zu wenig intuitiv befunden, eine Möglichkeit wäre, die beiden Bereiche zu vereinen und die Benutzer gruppiert nach ihrer zugehörigen Gruppe anzuzeigen.

Die clientseitige Generierung der Tag Cloud und PDF Reports sieht der Autor dieser Arbeit rückwirkend als eine fragliche Lösungsvariante an, da clientseitig unterschiedliche Endgeräte und Browser eingesetzt werden, die Resultate somit von Client zu Client unterschiedlich ausfallen können. Eine serverseitige Umsetzung wäre durchaus denkbar und sollte bei der Entwicklung eines ähnlichen Systems in Betracht gezogen werden.

Auch soll an dieser Stelle nochmals genannt werden, dass das Authentifizierungs- und Autorisierungskonzept für einen Prototyp konzipiert wurde und so auf keinen Fall für ein produktives System genutzt werden sollte. Unter dem Stichwort OAuth2 finden sich in der Forschung sicherere Konzepte als die in der vorliegenden Arbeit verwendeten.

8 Schlussfolgerung

Die in dieser Forschungsarbeit erlangten Erkenntnisse betreffend Nützlichkeit und Akzeptanz von Tag Clouds in Applikationen zur Unterstützung von TMS in Teams dienen als Anhaltspunkte für die Entwicklung von solchen Systemen sowie der Verwendung von Tag Clouds zur Darstellung und Auswertung von Meta-Wissen. Die Ergebnisse zeigen, dass die Tag Cloud für die Erfassung und Darstellung der Meta-Informationen «who does what» und «who is interested in what» genutzt werden kann. Durch die einfache und zugleich kreative Darstellung von Informationen wird sie von Benutzern akzeptiert und im Kontext vom TMS als nützlich wahrgenommen. Aufgrund der Eigenschaften der Tag Cloud eignet sie sich jedoch nicht für sämtliche Arten von Auswertungen. Während sie Benutzern dabei hilft, sich einen Überblick über die vorhandenen Meta-Informationen zu verschaffen sowie Trends zu erkennen, offenbart sie bei der Suche und Auswertung von spezifischen Informationen Schwächen. Wie bereits Cardew-Hall (2008) kommt auch diese Arbeit zum Schluss, dass die Tag Cloud komplementär verwendet werden soll und Benutzern, zusätzlich zur Tag Cloud, alternative Möglichkeiten zur spezifischen Informationssuche zur Verfügung gestellt werden müssen. Zudem gibt es diverse Ansätze, um den Nutzen und die Usability der Tag Cloud zu erhöhen. Im Falle des in dieser Arbeit vorgestellten Informationssystems, sollte weiter untersucht werden, wie sich die Informationen gruppieren und die Datengrundlage sauber halten lässt, um eine längerfristige Auswertbarkeit mittels der Tag Cloud sicherzustellen.

Während in dieser Arbeit mit dem Konzept Tag Cloud ein kreativer Vorschlag eines Informationssystems zur Unterstützung von TMS in Teams gemacht wurden, konnte die Frage, wie ein solches System die Akzeptanz der Benutzer erlangt, nur teilweise beantwortet werden. So wurde erkannt, dass die Usability einen wesentlichen Einfluss darauf hat, ob ein System mittels Tag Cloud akzeptiert wird. Wie ein solches System in der Praxis in echten Teams akzeptiert und wahrgenommen wird und welche Faktoren dabei von Bedeutung sind, müsste durch weitere Versuche evaluiert werden.

Literatur- und Quellenverzeichnis

- Alavi, M. & Tiwana, A. (2002). Knowledge integration in virtual teams: The potential role of KMS. *Journal of the American Society for Information Science and Technology*, 53(12), 1029–1037. <https://doi.org/10.1002/asi.10107>
- Bateman, S., Gutwin, C. & Nacenta, M. (2008). Seeing Things in the Clouds: The Effect of Visual Features on Tag Cloud Selections. In P. Brusilovsky (ed.), *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia* (S. 193). New York, NY: ACM.
- Boland, R. J., Tenkasi, R. V. & Te'eni, D. (1994). Designing Information Technology to Support Distributed Cognition. *Organization Science*, 5(3), 456–475. <https://doi.org/10.1287/orsc.5.3.456>
- Dayan, M. & Basarir, A. (2009). Antecedents and consequences of team reflexivity in new product development projects. *Journal of Business & Industrial Marketing*, 25(1), 18–29. <https://doi.org/10.1108/08858621011009128>
- Dayan, M. & Di Benedetto, C. A. (2009). Antecedents and consequences of teamwork quality in new product development projects. *European Journal of Innovation Management*, 12(1), 129–155. <https://doi.org/10.1108/14601060910928201>
- Del Sole, A. (2019). *Visual Studio Code Distilled. Evolved Code Editing for Windows, macOS, and Linux*. Berkeley, CA: Apress. <https://doi.org/10.1007/978-1-4842-4224-7>
- Deutsch, S., Schrammel, J. & Tscheligi, M. (2011). Comparing Different Layouts of Tag Clouds: Findings on Visual Perception. In A. Ebert, A. Dix, N. D. Gershon & M. Pohl (Eds.), *Human aspects of visualization. Second IFIP WG 13.7 Workshop on Human-Computer Interaction and Visualization, HCIV (INTERACT) 2009, Uppsala, Sweden, August 24, 2009 ; revised selected papers* (Lecture Notes in Computer Science, vol. 6431, pp. 23–37). Berlin: Springer. https://doi.org/10.1007/978-3-642-19641-6_3
- Docker [Computer Software]. Abgerufen von <https://www.docker.com/>

- Doglio, F. (2018). *REST API Development with Node.js. Manage and Understand the Full Capabilities of Successful REST Development* (2nd ed.). Berkeley, CA: Apress. <https://doi.org/10.1007/978-1-4842-3715-1>
- Eidgenössischer Bundesrat. (2020). Verordnung 2 über Massnahmen zur Bekämpfung des Coronavirus (COVID-19). Abgerufen von <https://www.admin.ch/opc/de/classified-compilation/20200744/index.html>
- Fink, G. & Flatow, I. (Eds.). (2014). *Pro single page application development. Using Backbone.js and ASP.NET* (Books for professionals by professionals). New York, NY: Apress/Springer.
- Gottron, T. (2009). Document Word Clouds: Visualising Web Documents as Tag Clouds to Aid Users in Relevance Decisions. In M. Agosti, J. Borbinha, S. Kapidakis, C. Papatheodorou & G. Tsakonias (Eds.), *Research and advanced technology for digital libraries. 13th European conference, ECDL 2009, Corfu, Greece, September 27 - October 2, 2009 ; proceedings* (Lecture Notes in Computer Science, vol. 5714, pp. 94–105). Berlin: Springer. https://doi.org/10.1007/978-3-642-04346-8_11
- Heroku [Computer Software]. Abgerufen von <https://www.heroku.com/>
- Hevner, March, Park & Ram. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75. <https://doi.org/10.2307/25148625>
- Hsu, J. S.-C., Shih, S.-P., Chiang, J. C. & Liu, J. Y.-C. (2012). The impact of transactive memory systems on IS development teams' coordination, communication, and performance. *International Journal of Project Management*, 30(3), 329–340. <https://doi.org/10.1016/j.ijproman.2011.08.003>
- Jackson, P. (2011). Requirements for an information system to support transactive memory. *VINE*, 41(4), 401–427. <https://doi.org/10.1108/03055721111188511>
- Jackson, P. & Klobas, J. (2008). Transactive memory systems in organizations: Implications for knowledge directories. *Decision Support Systems*, 44(2), 409–424. <https://doi.org/10.1016/j.dss.2007.05.001>

- Jangla, K. (Ed.). (2018). *Accelerating Development Velocity Using Docker. Docker Across Microservices*. Berkeley, CA: Apress. <https://doi.org/10.1007/978-1-4842-3936-0>
- Khusro, S., Jabeen, F. & Khan, A. (2018). Tag Clouds: Past, Present and Future. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*. <https://doi.org/10.1007/s40010-018-0571-x>
- Laski, J. & Stanley, W. (2009). Prepare Test for Any Implementation: Black-Box Testing. In W. Stanley & J. Laski (Eds.), *Software Verification and Analysis. An Integrated, Hands-On Approach* (pp. 81–99). London: Springer-Verlag London. https://doi.org/10.1007/978-1-84882-240-5_4
- Lehner, F. (2019). *Wissensmanagement. Grundlagen, Methoden und technische Unterstützung* (6. Auflage). Abgerufen von <https://doi.org/10.3139/9783446461000>
- Leung, H. K.N. & Wong, P. W.L. (1997). A study of user acceptance tests. *Software Quality Control*, 6(2), 137–149. <https://doi.org/10.1023/A:1018503800709>
- Lewis, K. (2003). Measuring transactive memory systems in the field: scale development and validation. *The Journal of Applied Psychology*, 88(4), 587–604. <https://doi.org/10.1037/0021-9010.88.4.587>
- Maki, M. (2017, 3. Dezember). A Brief Overview of React Router and Client-Side Routing [Blog-Beitrag]. Abgerufen von <https://medium.com/@marcellamaki/a-brief-overview-of-react-router-and-client-side-routing-70eb420e8cde>
- Moreland, R. L. & Myaskovsky, L. (2000). Exploring the Performance Benefits of Group Training: Transactive Memory or Improved Communication? *Organizational Behavior and Human Decision Processes*, 82(1), 117–133. <https://doi.org/10.1006/obhd.2000.2891>
- Nikodemus, P. (2017). *Lernprozessorientiertes Wissensmanagement und kooperatives Lernen*. Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-17681-5>
- PostgreSQL [Computer Software]. Abgerufen von <https://www.postgresql.org/>

- Postman [Computer Software]. Abgerufen von <https://www.postman.com/>
- Probst, G. J. B., Raub, S. P. & Romhardt, K. (2012). *Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen* (7. Aufl.). Wiesbaden: Springer Gabler. <https://doi.org/10.1007/978-3-8349-4563-1>
- React-bootstrap [Computer Software]. Abgerufen von <https://react-bootstrap.github.io/>
- React-wordcloud [Computer Software]. Abgerufen von <https://react-wordcloud.netlify.app/>
- Redux [Computer Software]. Abgerufen von <https://redux.js.org/>
- Rivadeneira, A. W., Gruen, D. M., Muller, M. J. & Millen, D. R. (2007). Getting our head in the clouds. In M. B. Rosson (ed.), *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (S. 995–998). New York, NY: ACM.
- Schüppel, J. (1997). *Wissensmanagement. Organisatorisches Lernen im Spannungsfeld von Wissens- und Lernbarrieren* (Gabler-Edition Wissenschaft, 1. Aufl., Nachdr). Zugl.: St. Gallen, Univ., Diss., 1996. Wiesbaden: Dt. Univ.-Verl.
- Sequelize [Computer Software]. Abgerufen von <https://sequelize.org/>
- Sinclair, J. & Cardew-Hall, M. (2008). The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34(1), 15–29. <https://doi.org/10.1177/0165551506078083>
- Spindler, A. de, Leone, S., Nebeling, M., Geel, M. & Norrie, M. C. (2011). Using Synchronised Tag Clouds for Browsing Data Collections. In H. Mouratidis & C. Rolland (Eds.), *Advanced information systems engineering. 23rd international conference, CAiSE 2011, London, UK, June 20 - 24, 2011 ; proceedings* (Lecture Notes in Computer Science, vol. 6741, vol. 6741, pp. 214–228). Berlin: Springer. https://doi.org/10.1007/978-3-642-21640-4_17
- Swagger [Computer Software]. Abgerufen von <https://swagger.io/>

- Thakkar, M. (2020). *Building React Apps with Server-Side Rendering. Use React, Redux, and Next to Build Full Server-Side Rendering Applications* (1st ed. 2020). <https://doi.org/10.1007/978-1-4842-5869-9>
- Trattner, C., Helic, D. & Strohmaier, M. (2014). Tag Clouds. In R. Alhajj & J. Rokne (Eds.), *Encyclopedia of social network analysis and mining* (pp. 2103–2107). New York, NY: Springer. https://doi.org/10.1007/978-1-4614-6170-8_126
- Tullis T., Fleischman S., McNulty M., Cianchette C. & Bergel M. (2002). *An empirical comparison of lab and remote usability testing of Web sites*. Abgerufen von https://www.researchgate.net/publication/228540469_An_empirical_comparison_of_lab_and_remote_usability_testing_of_Web_sites
- Wang, Y., Huang, Q., Davison, R. M. & Yang, F. (2018). Effect of transactive memory systems on team performance mediated by knowledge transfer. *International Journal of Information Management*, 41, 65–79. <https://doi.org/10.1016/j.ijinfomgt.2018.04.001>
- Wegner, D. M. (1987). Transactive Memory: A Contemporary Analysis of the Group Mind. In B. Mullen & G. R. Goethals (Eds.), *Theories of Group Behavior* (Springer Series in Social Psychology, vol. 46, pp. 185–208). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4612-4634-3_9
- Yuan, Y. C., Fulk, J. & Monge, P. R. (2007). Access to Information in Connective and Communal Transactive Memory Systems. *Communication Research*, 34(2), 131–155. <https://doi.org/10.1177/0093650206298067>
- Zhong, X., Huang, Q., Davison, R. M., Yang, X. & Chen, H. (2012). Empowering teams through social network ties. *International Journal of Information Management*, 32(3), 209–220. <https://doi.org/10.1016/j.ijinfomgt.2011.11.001>

Anhang

A Testplan

Testfall #	Anforderung	Beschreibung	Erwartetes Ergebnis
1 Registration Login/Logout			
1a	A1	Als neuen Benutzer registrieren	Registration erfolgreich wird angezeigt, Benutzer wird auf Login-Seite weitergeleitet.
1b	A1	Mit bereits registrierter E-Mail-Adresse registrieren	Registration fehlgeschlagen. Fehlermeldung wird angezeigt.
1c	A1	Registration mit unvollständiger Eingabe	Felder werden validiert, Request wird nicht übermittelt.
2 Login/Logout			
2a	A2	Mit gültigen Anmeldedaten einloggen	Login erfolgreich, die Hauptseite wird angezeigt.
2b	A2	Mit ungültigen Anmeldedaten einloggen	Login fehlgeschlagen. Fehlermeldung wird angezeigt.
2c	A3	Benutzer abmelden	Benutzer wird ausgeloggt, Login Seite wird angezeigt.
3 Tags erfassen			
3a	A4	Neues Tag erfassen (Interessiert an)	Neues Tag erfasst und wird in Tag Cloud angezeigt.
3b	A4	Neues Tag erfassen (Arbeiten an)	Neues Tag erfasst und wird in Tag Cloud angezeigt.
3c	A4	Bestehendes Tag aus Dropdown auswählen und hinzufügen	Post wird erfasst, Anzahl Posts erhöht sich um 1.

3d	A4	Tag posten ohne Eingabe oder nur Leerzeichen	Fehlermeldung wird angezeigt.
4 Tag Cloud anzeigen			
4a	A5, A7	Persönliche Tag Cloud anzeigen (Interessiert an oder Arbeite an)	Tag Cloud und Tabelle wird angezeigt, nur Tags von eingeloggtem Benutzer ersichtlich.
4b	A6, A7	Tag Cloud aller Benutzer anzeigen («Alle», Interessiert an oder Arbeite an)	Tag Cloud und Tabelle mit sämtlichen verfügbaren Posts des entsprechenden Typs werden angezeigt.
4c	B2, A7	Tag Cloud «Meine Gruppe» anzeigen	Tag Cloud und Tabelle mit sämtlichen verfügbaren Posts des entsprechenden Typs aller Benutzer der Gruppe des eingeloggten Benutzers werden angezeigt.
4d	B2, A7	Tag Cloud einer anderen Gruppe anzeigen	Tag Cloud und Tabelle mit sämtlichen verfügbaren Posts des entsprechenden Typs aller Benutzer der gewählten Gruppe werden angezeigt.
5 Tag Details anzeigen			
5a	A7, A8	Tag in angezeigter Tag Cloud anzeigen	Tag Details werden angezeigt, pro Tag ist ein Ranking nach Benutzern pro Typ ersichtlich.
6 Filter verwenden			
6a	A5, A6	Anzahl Tags limitieren durch Verwenden des Sliders	Anzahl Tags in der Tag Cloud und Tabelle reduziert sich auf den ausgewählten Wert.
6b	A5, A6	Datumsbereich einschränken	Es werden nur Tags angezeigt, die im gewählten Datumsbereich gepostet wurden.

6c	A5, A6	Buttons für Datumsbereich verwenden: <ul style="list-style-type: none"> - Heute - - 7 Tage - - 30 Tage 	Datum von-bis wird entsprechend der Auswahl angepasst. Es werden nur Tags angezeigt, die im gewählten Datumsbereich gepostet wurden.
6d	A5, A6	Farben deaktivieren/aktivieren mittels Schalter	Tags werden nur schwarz bzw. mit Farben dargestellt.
7 Reports			
7a	B5	Angezeigte Tags als CSV exportieren	Die angezeigten Tags und deren Anzahl Posts werden in ein CSV exportiert (entspricht der angezeigten Tabelle).
7b	B5	Angezeigte Tag Cloud als PNG exportieren	PNG entspricht der angezeigten Tag Cloud.
7c	B5	PDF Report erstellen	PDF enthält Tag Cloud wie abgebildet auf der Seite und eine Tabelle mit den angezeigten Tags (entspricht der angezeigten Tabelle).
8 Autorisierungskonzept			
8a	B1	Eingeloggt als Benutzer mit Rolle 'user'	Administrationsbereich wird nicht angezeigt.
8b	B1	Eingeloggt als Benutzer mit Rolle 'admin'	Administrationsbereich wird angezeigt.
9 Index Editor			
9a	A9	Tag zum Index hinzufügen	Neues Tag wird im Index-Editor angezeigt, wenn Details angezeigt werden, erscheint die Anzahl Posts = 0. Bestätigung wird angezeigt.

9b	A9	Tag im Index bearbeiten	Wert des Tags ändert sich in der Tabelle, Änderungsbestätigung wird angezeigt. Geändert es Tag wird in der Tag Cloud mit neuem Wert angezeigt. Die Änderung muss im Popup bestätigt werden.
9c	A9	Tag aus Index entfernen	Tag wird aus Tabelle entfernt, Lösungsbestätigung wird angezeigt. Sämtliche Posts zu diesem Tag werden gelöscht, Tag erscheint in keiner Tag Cloud. Das Löschen muss im Popup bestätigt werden.
9d	A9	Suchen, Blättern, Sortieren und Anzahl Zeilen pro Tabelle funktioniert	Die Navigation in der Tabelle funktioniert wie erwartet.
10 Benutzerverwaltung			
10a	B4	Benutzer anzeigen	Alle Benutzer werden angezeigt bis auf den eingeloggtten Benutzer.
10b	B4	Benutzer bearbeiten	Felder Name, Vorname und E-Mail, Rolle und Gruppe können bearbeitet werden. Änderung der Rolle muss in Popup bestätigt werden.
11 Gruppenverwaltung			
11a	B3	Gruppen anzeigen	Alle Gruppen werden angezeigt.
11b	B3	Gruppe hinzufügen	Neue Gruppe erscheint in Tabelle. Bestätigung wird angezeigt.
11c	B3	Gruppe bearbeiten	Der Name und die Beschreibung lassen sich bearbeiten. Bestätigung wird angezeigt.

11d	B3	Gruppe entfernen	Gruppe lässt sich entfernen, Benutzer der zu entfernenden Gruppe werden der Gruppe mit ID 1 (Initiale Gruppe) zugewiesen, Änderung ist in Benutzerverwaltung ersichtlich. Das Löschen muss im Popup bestätigt werden.
-----	----	------------------	---

B Aufgabenstellung User-Acceptance-Testing

1. Registrierung

- a. Öffnen Sie die Website <https://tag-cloud-app.herokuapp.com/>
- b. Registrieren Sie sich (die E-Mail-Adresse muss nicht zwingend existieren, jedoch ein gültiges Format aufweisen und wird nur fürs Login benötigt)

2. Anmelden

- a. Melden Sie sich mit ihren zuvor gewählten Angaben an.

3. Tag erfassen

- a. Erfassen Sie ein beliebiges Tag an dem Sie gerade arbeiten (z.B. «User Acceptance Testing»)
- b. Erfassen Sie Ihr Hobby als Interesse

4. Tags finden

- a. Suchen Sie die von Ihnen erfassten Tag in ihrer persönlichen Tag Cloud.
- b. Finden Sie dieselben Tags in der Tag Cloud ihrer Gruppe
- c. Deaktivieren Sie die Farben, finden Sie nun das Tag «Verhaltenskodex»?

5. Auswerten (verwenden Sie die Einstellungen nach Ihren persönlichen Präferenzen)

- a. Wie viele Tags befinden sich in der Tag Cloud Ihrer Gruppe?
- b. Welches ist das meist gepostete Tag in deiner Gruppe?
- c. Reduzieren Sie die Anzahl angezeigter Tags soweit, dass sämtliche Tags die gleiche Farbe haben.
- d. In ihrer Gruppe
 - i. Wer hat das Tag «UXDesign» erstellt?
 - ii. Wer interessiert sich wohl am meisten für «UXDesign»?
 - iii. Wer arbeitete am meisten an «UrbanGardening»?
 - iv. An was war im Zeitraum von 05.05.2020 bis 06.05.2020 das Interesse am grössten?

6. Administration

- a. Wechsel zum Administrations Bereich
 - i. Loggen Sie sich aus

- ii. Loggen Sie sich als Administrator User ein:
 - 1. E-Mail: admin@mail.ch
 - 2. Password: admin
- iii. Gehen Sie zum Administrations Bereich
- b. Ändern Sie die Gruppe des Users «Heinz Günter» zu einer beliebigen Gruppe.
- c. Fügen Sie ein beliebiges Tag zum Index hinzu
- d. Ändern Sie das zuvor hinzugefügte Tag zu einem neuen Wert
- e. Löschen Sie das Tag mit Ihrem Hobby.
- f. Exportieren Sie die Tags aus dem Tag Index als CSV

C Fragebogen User-Acceptance-Testing

1. Fragen zum Benutzer

- a. Hatten Sie bereits vor Ihrer Teilnahme an dieser Arbeit Kontakt mit Tag Clouds?
[Ja/Nein]
- b. Wie schätzen Sie Ihre Kenntnisse betreffend IT-Applikationen ein?
[5 Punkte Skala]

2. Auswertung der Aufgaben

- a. Welche der gestellten Aufgaben konnten Sie erledigen?
Aufgabe 1-6 [Ja/Nein]
Wie anspruchsvoll haben Sie das Lösen der Aufgaben empfunden?
Aufgabe 1-6 [5 Punkte Skala]
- b. Sie konnten die Aufgabe x nicht lösen. Bitte erläutern Sie kurz, welche Probleme Sie hatten.
[freier Text – pro Aufgabe die nicht gelöst werden konnte]
- c. Was hat aus Ihrer Sicht verhältnismässig viel Zeit für die Abarbeitung der Aufgaben beansprucht?

3. Die Tag Cloud als Visualisierungs-Instrument

- a. Wie empfanden Sie die Tag Cloud in Bezug auf Interpretier- und Auswertbarkeit?
[5 Punkte Skala]
- b. Wie hilfreich empfinden Sie die Einfärbung der Tags? (Hinweis: In Aufgabe 4 mussten Sie ein Tag ohne Einfärbung suchen)
[5 Punkte Skala]
- c. Wie viele Tags sollten Ihrer Meinung nach maximal angezeigt werden, damit die Tag Cloud die volle Aussagekraft entfaltet?
[Auswahl: 1-10, 11-20, 21-30, 31-40, mehr als 40]
- d. Denken Sie, dass die Applikation hilfreich ist / sein kann?
[Ja/Nein]
- e. Sie haben die letzte Frage mit 'Ja' beantwortet, bitte begründen Sie kurz. Welchen Nutzen schafft Ihrer Meinung nach diese Applikation?
[Wenn d. Ja; freier Text]

Sie haben die letzte Frage mit 'Nein' beantwortet, bitte begründen Sie kurz.

[Wenn d. Nein; freier Text]

4. Blick in die Zukunft

- a. Können Sie sich vorstellen diese oder eine ähnliche Applikation, die Tag Clouds für die Datenpräsentation verwendet, in einem Team einzusetzen?

[Ja/Nein]

- b. Warum?

[Wenn a. Ja; freier Text]

- c. Denken Sie, dass Sie dadurch bessere Kenntnisse darüber erlangen, an was ihre Kollegen arbeiten bzw. wofür sie sich interessieren?

[Wenn a. Ja; freier Text]

- d. Warum nicht? Was müsste sich ändern, dass Sie die vorherige Frage mit 'Ja' beantworten würden?

[Wenn a. Nein; freier Text]

- e. Haben Sie weiter Anmerkungen oder Verbesserungsvorschläge?

D Resultate User-Acceptance Testing

Die Resultate des Fragebogens, welcher Teil der Evaluation ist, wurden zusammen mit der Bachelor-Thesis in Form einer ZIP-Datei eingereicht.

- Statistik_UAT_Fragebogen.xls

E Installations- und Betriebsanleitung

1. Installation

Voraussetzungen

Docker

Für die Installation bzw. das Starten ist eine Docker Desktop Installation notwendig. In dieser Arbeit wurde die zu diesem Zeitpunkt aktuellste Version 2.3.0.2 verwendet, wichtig ist, dass die Docker-Engine (mit Docker Desktop v 2.3.0.2 ist die Docker-Engine v19.03.8 installiert) docker-compose Version 3.0 unterstützt, dies ist ab Docker-Engine v.1.13.0 der Fall.

Internetverbindung

Da während der Installation die benötigten npm (Node package manager) Packages und die benötigten Container-Images (Postgres:latest + Node:12-alpine) heruntergeladen werden, ist eine Internetverbindung notwendig. Nach der ersten Installation, sofern die Docker-Images nicht gelöscht wurden, ist der Betrieb auch ohne Internetverbindung möglich.

Applikation Kompilieren und Starten

Im Root-Verzeichnis des Sourcecodes ist in der Kommandozeile (unter Windows: Start → Ausführen → CMD) der folgende Befehl auszuführen:

```
> docker-compose -f docker-compose-standalone.yml up
```

Da bei der Erstinstallation die Container noch nicht verfügbar sind, dauert der Vorgang einige Minuten. Dieser Vorgang beinhaltet folgendes:

- Herunterladen der benötigten Images
- Installation mittels Node.js von Client (React Frontend) und Server (Node.js Backend)
- Starten der Container
- Initialisierung der Applikation inklusive Erstellung des Admin Benutzers und der Initial-Gruppe (die Einträge für die Initialen DB Inserts mittels Sequelize seeds befinden sich in der Datei: /server/middlewares/Database-init.js)

Das folgende Log (zusammengefasst) ist während des ersten Starts ersichtlich sein:

```
Pulling db-standalone (postgres:)...
Status: Downloaded newer image for postgres:latest
Building backend-standalone
Successfully built 962f3bb0b414
Building client-standalone
Successfully built fc926578ef7e
Creating tagcloud-app_db-standalone_1 ... done
Creating tagcloud-app_backend-standalone_1 ... done
Creating tagcloud-app_client-standalone_1 ... done
db-standalone_1      | 2020-05-23 17:34:02.549 UTC [1] LOG:  database
system is ready to accept connections
backend-standalone_1 | == Database-init: migrating =====
backend-standalone_1 | Initial Group does not exist - gets inserted
backend-standalone_1 | Admin does not exist - gets inserted
backend-standalone_1 | Server started on port 5200...
backend-standalone_1 | Running Mode: development
backend-standalone_1 | Connection String DB:
postgres://dbuser:mypassword@db-standalone:5432/standalonedb
client-standalone_1  | API Url: http://localhost:3000/api
client-standalone_1  | i [wdm]: Compiled with warnings.
```

Nach erfolgreichem Startup ist die Applikation wie folgt verfügbar:

Frontend: <http://localhost:8080>

Backend: <http://localhost:3000/api>

Anmeldeinformationen Administrator:

E-Mail-Adresse: admin@admin.ch

Passwort: admin1234

Applikation Stoppen

Die Applikation lässt sich wie folgt stoppen, in der Kommandozeile:

1. Unter Windows: CTRL+ C drücken oder alternativ:
2. In der Kommandozeile: `> docker-compose -f docker-compose-standalone.yml down` ausführen

Beim Stoppen werden die Daten in der Datenbank auf dem Host-System mittels Docker Volumes persistiert und stehen beim nächsten Start wieder zur Verfügung. Sollen die Daten beim Stoppen gelöscht werden, kann folgender Befehl ausgeführt werden:

```
> docker-compose -f docker-compose-standalone.yml down -v
```

Applikation von System entfernen

Um sämtliche von Docker angelegten Daten zu entfernen, kann folgender Befehl verwendet werden (Achtung! Es werden sämtliche von Docker angelegte Daten gelöscht, auch solche die nicht von dieser Applikation stammen.):

```
> docker system prune -a
```

Daten in Datenbank laden

Als Teil des Quellcodes dieser Arbeit wurde ein Backup der Datenbank mit der Datengrundlage, die für das User-Acceptance-Testing verwendet wurde abgeben.

Mit den folgenden zwei Befehlen, lassen sich die Daten in DB laden:

```
> docker cp .\dev-tools\testdata.sql tagcloud-app_db-standalone_1:/var/lib/postgresql/data
```

```
> docker exec tagcloud-app_db-standalone_1 pg_restore --no-owner --no-privileges --no-tablespaces --clean --no-comments --verbose -h localhost -U dbuser -d standalonedb /var/lib/postgresql/data/testdata.sql
```

2. Betrieb

Die folgenden Abschnitte beinhalten die Betriebsanleitung des Prototyps.

Registrieren und Anmelden

Beim Aufruf des Prototyps erscheint zuerst immer die Anmelde-Seite. Durch Eingabe von bestehenden Anmeldedaten kann man sich anmelden, sollten noch keine Anmeldedaten vorhanden sein, kann über den Link «Registrieren» die Registrations-Seite aufgerufen werden. Die Registrierung ist erfolgreich, wenn die Bestätigungsmeldung «Registration erfolgreich, Sie können sich nun einloggen.» erscheint, mit den erstellten Anmeldedaten können Benutzer sich ab dann anmelden. In der folgenden Abbildung sind die beiden Seiten und Navigationselemente gezeigt.

Login

E-Mail-Adresse

Passwort

[Registrieren](#)

Registrierung

Vorname

Nachname

E-Mail Adresse

Passwort

[zurück](#)

Hauptseite

Wenn valide Anmeldedaten eingegeben wurden, wird der Benutzer auf die Hauptseite weitergeleitet. Für Benutzer ohne Administrator-Rolle sind sämtliche Funktionen des Prototyps auf der Hauptseite, der in der nachfolgenden Abbildung gezeigt wird. Die Beschrifteten Elemente und deren funktionsweise wird in der Folge näher beschrieben.

The screenshot shows the main page interface with the following elements and callouts:

- 1) Login status: "Eingeloggt als: Albert Stein" with a dropdown arrow.
- 2) Post form: "Arbeite an" dropdown, "Tag hinzufügen..." text input, and "Post" button.
- 3) Tag Cloud: A collection of tags including "WissenschaftsartikelEinstein", "LohnentwicklungImGesundheitswesen", "TransaktivesGedächtnis HomeOfficeGuide", and "Wirtschaft4.0".
- 4) "Meine Tag Cloud" dropdown.
- 5) "Tag Cloud Einstellungen" link.
- 6) "Suchen..." search input field.
- 7) "↓" dropdown arrow.
- 8) A table showing tag statistics.

Tag	Posts
Wirtschaft4.0	3
LohnentwicklungImGesundheitswesen	2
WissenschaftsartikelEinstein	2

Die Tags werden in Form der Tag Cloud 5) und der Tabelle in Bereich 8) angezeigt. Die Tag Cloud und die Tabelle beinhaltet immer dieselben Tags. Über das Feld «Suchen...» werden nur die Einträge der Tabelle eingeschränkt.

Abmelden

Durch Auswählen von «Logout» im Dropdown 1) kann sich der Benutzer ausloggen.

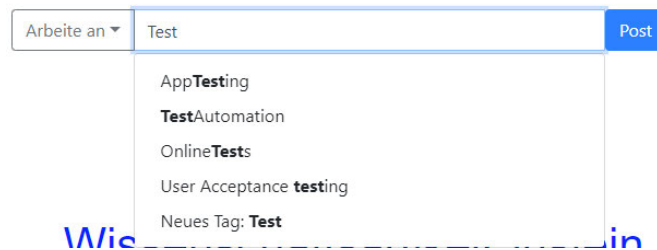
Gruppe anzeigen

Durch Auswählen des Dropdown 1) wird die Gruppe, der der eingeloggte Benutzer angehört angezeigt.

Tag erfassen

Der umrandete Bereich zeigt das Eingabe Feld, das genutzt wird, um neue Tags zu erfassen. Im Dropdown auf der linken Seite kann der Benutzer auswählen, ob er ein Tag als Typ «Interessiert an» oder «Arbeite an» erfassen will. Bei der Eingabe erhält der Benutzer im Dropdown Vorschläge, von bestehenden Tags die seiner Eingabe entsprechen. Er kann mittels Mauszeiger oder Pfeiltasten ein bestehendes Tag auswählen und durch Klicken auf den Button «Post» erfassen. Alternativ kann die Eingabe auch ohne Mauseinsatz erfolgen durch Drücken der Tab- gefolgt von der Enter-Taste.

Hinweis: Auch wenn kein vorgeschlagenes Tag ausgewählt wird, die Eingabe aber einem bestehenden Tag entspricht, wird kein neues Tag zum Index hinzugefügt, sondern lediglich ein neuer Post erfasst. Das Erfassen von Duplikaten ist nicht möglich.



Tag Clouds anzeigen

Durch die beiden Dropdowns 3) und 4) kann der Benutzer auswählen, welche Tags in der Tag Cloud angezeigt werden. Im Dropdown 3) wählt er ob er Tags die als Typ «Arbeite an» oder «Interessiert an» anzeigen will. Im Dropdown 4) wählt er von welchen Benutzern er Tags anzeigen will:



Im Dropdown 5) kann der Benutzer Auswählen, dass:

- Nur von ihm erfasste Tags → «Meine Tag Cloud»
- Nur Tags von Benutzer seiner Gruppe → «Meine Gruppe»

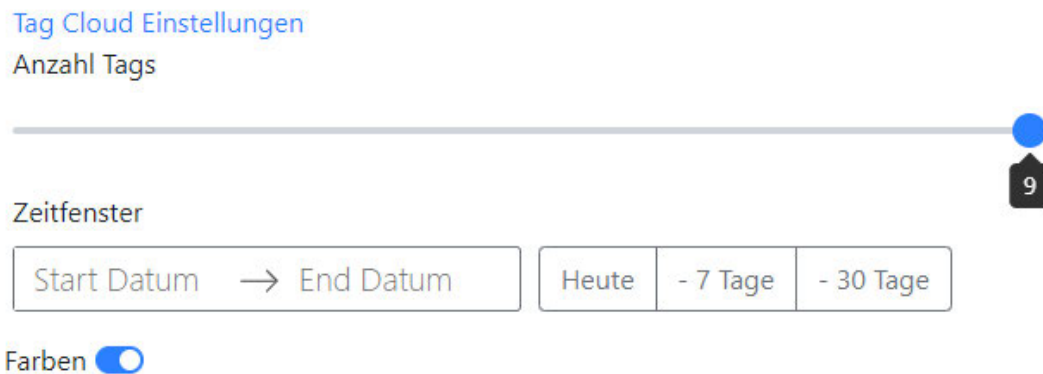
- Tags aller Benutzer → «Alle»
- Tags einer bestimmten anderen Gruppe → Werte unter ersten Trennlinie
- Tags eines bestimmten Benutzers → Werte unter der zweiten Trennlinie

angezeigt werden.

Durch jede Auswahl wird die Tag Cloud entsprechend der gewählten Parameter neu geladen.

Tag Cloud Einstellungen verwenden

Durch Anwählen des Links «Tag Cloud Einstellungen» werden weitere Input-Felder angezeigt, mit denen die Tag Cloud bzw. die anzuzeigenden Tags weiter eingeschränkt werden können. Die folgende Abbildung zeigt die verfügbaren Einstellungen.



Durch den Slider kann die Anzahl Tags in der Tag Cloud reduziert werden, es wird jeweils das am wenigsten populäre (kleinste Anzahl Posts) ausgeblendet.

Über die Eingabe eines Zeitfensters werden nur die Tags die in diesem Zeitraum hinzugefügt wurden angezeigt. Über die drei Buttons können die vordefinierten Zeiträume direkt angewählt werden.

Über den Schalter «Farben» lässt sich die Einfärbung der Tags in der Tag Cloud deaktivieren bzw. aktivieren.

Reports erstellen

Über das Dropdown 7) lassen die die angezeigten Daten in Form von CSV, PNG und PDF exportieren:

- | | |
|-----|---|
| CSV | Die angezeigten Tags werden in der Form der Tabelle 8) als CSV heruntergeladen. |
| PNG | Die angezeigte Tag Cloud wird als PNG Bilddatei heruntergeladen. |
| PDF | Es wird ein PDF-Dokument erstellt, dass die angezeigte Tag Cloud als Bild, die verwendeten Parameter und die Tabelle 8) enthält. Durch die Auswahl von PDF im Dropdown 7) wird das Dokument erstellt, in einem zweiten Schritt kann es vom Benutzer über den Link «PDF herunterladen» heruntergeladen werden. |

Tag Details anzeigen

Durch Anklicken eines Tags in der Tag Cloud 5) oder der Tabelle im Bereich 8) werden die Details des Tags angezeigt. Die angezeigten Daten werden anhand der folgenden Abbildung erläutert:

The screenshot shows a window titled 'UrbanGardening' with a close button (x) in the top right corner. The window is divided into two main sections, labeled 1) and 2).

Section 1) displays meta-information in a table:

Erstellt von:	Manuel Votapek
Erstellt am:	06.05.2020 16:43:40
Geändert von:	Manuel Votapek
Geändert am:	06.05.2020 16:43:40
Total Posts:	15
Erstes mal gepostet am:	06.05.2020 16:46:32
Letztes mal gepostet am:	23.05.2020 20:53:08

Section 2) displays user information in two parts:

Interessierte Benutzer

Arbeitende Benutzer

Vorname	Nachname	Anzahl Posts	Letzter Post	Erster Post
Alena	Barth	5	06.05.2020 17:19:41	06.05.2020 17:19:14
Admin	Admin	1	06.05.2020 20:14:29	06.05.2020 20:14:29

Im oberen Bereich 1) werden Meta-Informationen wie der Ersteller und Bearbeiter des Tags und zugehörige Zeitstempel oder der totalen Anzahl Posts zu diesem Tag angezeigt.

Im unteren Bereich 2) können die Anzahl Posts pro Benutzer angezeigt werden, jeweils zur entsprechenden Kategorie («Interessierte Benutzer» haben das Tag als Typ «Interessiert an» und «Arbeitende Benutzer» als «Arbeite an» hinzugefügt).

Administrationsbereich

Wenn der angemeldete Benutzer über die Administrator-Rolle verfügt, kann er zusätzlich zur Hauptseite zum Administrationsbereich wechseln. Dieser besteht aus den drei Teilen «Tag Index Editor», «Benutzerverwaltung» und «Gruppenverwaltung». Die verschiedenen Funktionen werden nun beschrieben.

Für sämtliche Funktionen gilt, dass bei erfolgreichem Abschluss einer Operation (Änderung in der Datenbank) jeweils eine Bestätigung und bei Fehlern eine Fehlermeldung angezeigt wird. Die folgende Abbildung zeigt je ein Beispiel:

Tag hinzugefügt



Hinzufügen fehlgeschlagen: Tag existiert bereits



Index Editor

Die folgende Abbildung zeigt den Index Editor.

The screenshot shows the 'Tag Index Editor' interface. At the top, there are navigation links for 'Benutzerverwaltung' and 'Gruppenverwaltung'. A search bar labeled 'Suchen...' is present. Below it is a table with columns: 'Tag Name', 'Erstellt von', 'Erstellt am', 'Geändert von', 'Geändert am', and 'Aktion'. The 'Aktion' column contains trash and information icons. Numbered callouts 1) through 6) point to the 'Tag' input field, the 'CSV Exportieren' button, the search bar, the 'Aktion' column header, the trash icon, the information icon, and the 'AppTesting' row respectively.

Tag Name	Erstellt von	Erstellt am	Geändert von	Geändert am	Aktion
AdminArbeit	Administrator	06.05.2020 20:13:43	Administrator	06.05.2020 20:13:43	
AgileMethoden	Peter Neu	06.05.2020 17:45:52	Peter Neu	06.05.2020 17:45:52	
Altersvorsorge	Beat Ernst	06.05.2020 17:17:07	Beat Ernst	06.05.2020 17:17:07	
AppTesting	Max Tester	06.05.2020 16:52:05	Max Tester	06.05.2020 16:52:05	

Verfügbare Funktionen:

Tag hinzufügen

Über das Eingabefeld 1) können neue Tags zum Index hinzugefügt werden. Entspricht die Eingabe einem bestehenden Tag wird eine Fehlermeldung angezeigt.

Tags exportieren

Über den Button 2) können sämtliche Tags aus dem Index als CSV exportiert werden, die Daten entsprechen der gezeigten Tabelle.

Tags suchen

Über das Suchfeld 3) können Tags gesucht werden.

Tag löschen

Durch klicken auf den Abfalleimer 4) können Tags aus dem Index gelöscht werden, dabei werden sämtliche zugehörigen Posts von Benutzern ebenfalls gelöscht.

Tag Details anzeigen

Durch klicken auf das Informationssymbol 5) werden die Tag Details analog zu den Tag Details auf der Hauptseite angezeigt.

Tags bearbeiten

Durch klicken auf ein Feld in der Spalte «Tag Name» wird der Bearbeitungsmodus aktiviert und der Tag Name kann bearbeitet werden.

Benutzerverwaltung

Die folgende Abbildung zeigt die Benutzerverwaltung. Die Felder Vorname, Nachname, E-Mail, Rolle und Gruppe können bearbeitet werden. Das Bearbeiten der Felder Vorname, Nachname, E-Mail funktioniert gleich wie im Index Editor. Für Rolle und Gruppe werden Dropdowns mit den verfügbaren Werten angezeigt, wie die folgende Abbildung zeigt:

Tag Index Editor Benutzerverwaltung Gruppenverwaltung

Suchen...

Vorname ↑↓	Nachname ↑↓	E-Mail ↑↓	Rolle ↑↓	Gruppe ↑↓	Erstellt am ↑↓	Geändert am ↑↓
Albert	Stein	albert@stein.ch	admin	Resear Nicht zugewiesen	06.05.2020 17:48:23	23.05.2020 21:58:28
Alena	Barth	alena@b.ch	user	Test-Team	05.2020 18:34	06.05.2020 19:45:32
Alexander	Muster	a.muster@email.ch	user	Dev-Team Researcher Researcher Nicht zugewiesen	05.2020 16:15:35	06.05.2020 19:43:15

Gruppenverwaltung

In der Gruppenverwaltung lassen sich Gruppen erstellen und entfernen.

Tag Index Editor Benutzerverwaltung Gruppenverwaltung

Gruppenname Beschreibung Gruppe Hinzufügen

Suchen...

Gruppenname ↑↓	Beschreibung ↑↓	Erstellt am ↑↓	Geändert am ↑↓	Aktion
Dev-Team	Dieser Gruppe werden Entwickler zugewiesen.	06.05.2020 17:10:52	06.05.2020 17:10:52	
Nicht zugewiesen	Dieser Gruppe werden neue Benutzer zugewiesen.	16.03.2020 17:01:10	03.05.2020 16:08:59	
Researcher	Alle Researcher gehören in diese Gruppe.	06.05.2020 17:12:38	06.05.2020 17:12:38	
Test-Team	Dieser Gruppe werden Tester zugewiesen.	06.05.2020 17:10:32	06.05.2020 17:10:32	

10 ▾

Die Bearbeitung von Gruppenname und Beschreibung, sowie das Löschen von Gruppen funktioniert gleich wie im Index-Editor und der Benutzerverwaltung.

Über die beiden Eingabefelder im oberen Bereich können neue Gruppen erstellt werden, sie sind in der Benutzerverwaltung danach verfügbar, um Benutzer zuzuweisen.

Wird eine Gruppe gelöscht, werden die Benutzer der Initial Gruppe «Nicht zugewiesen» hinzugefügt, diese Gruppe lässt sich nicht löschen.

F Quellcode

Zusammen mit der Bachelorarbeit wurde der komplette Quellcode in Form einer ZIP-Datei eingereicht.

- TagCloudApp.zip

G Demo Video

Zusammen mit der vorliegenden Arbeit wurde ein Video eingereicht, welches die Funktionen des Prototyps demonstriert.

- Demo_Video_Tag_Cloud_App.mp4

H PDF-Report

Zusammen mit der vorliegenden Arbeit wurde ein Beispiel eines mit dem Prototyp erstellten PDF-Reports abgegeben:

- tag_cloud_report.pdf