



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

Matthias Siano

Integration sozialer Medien in den Newsworkflow  
einer Rundfunkanstalt

**Matthias Siano**

Integration sozialer Medien in den Newsworkflow  
einer Rundfunkanstalt

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Wirtschaftsinformatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

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

Abgegeben am 15.06.2016

**Matthias Siano**

**Thema der Bachelorarbeit**

Integration sozialer Medien in den Newsworkflow einer Rundfunkanstalt

**Stichworte**

Soziale Medien, Nachrichten, Rundfunkanstalt, Nachrichtensuche

**Kurzzusammenfassung**

Diese Arbeit beschäftigt sich mit der Integration von sozialen Medien in den Newsworkflow einer Rundfunkanstalt. Zunächst wird der Einsatz von Nachrichten innerhalb einer Rundfunkanstalt analysiert, sowie die mögliche Anbindungen von Nachrichten aus den sozialen Medien. Daraus resultierend ergibt sich, dass die Nachrichten aus den sozialen Medien in bereits bestehende Suchvorgänge integriert werden sollen. Anschließend wird ein Entwurf konzipiert und bewertet, welches diese Integration von Nachrichten aus sozialen Medien ermöglichen soll.

**Matthias Siano**

**Title of the paper**

Integration of Social Media into the Newsworkflow of a Broadcaster.

**Keywords**

Social Media, News, Broadcaster, News Search

**Abstract**

This bachelor thesis is about the integration of social media into the newsworkflow of a broadcaster. At first the thesis analysis the usage of news within the broadcaster and the possibilities for the integration of news from social media. As a result of the analysis, the news from social media will be integrated into already existing search operations. As a result of the analysis a framework will be designed and evaluated, which should enable the integration of the news from social media.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
1.1	Ziele	7
1.2	Abgrenzungen	8
1.3	Struktur	8
<b>2</b>	<b>Grundlagen</b>	<b>9</b>
2.1	Soziale Medien	9
2.1.1	Netzwerke	11
2.1.2	Multimedia	13
2.1.3	Blogs	13
2.2	Rundfunk	15
2.3	Nachricht	15
2.4	Dokumentenorientierte Datenbank	16
<b>3</b>	<b>Analyse</b>	<b>17</b>
3.1	Szenario	17
3.1.1	Ad-Hoc-Suche	18
3.1.2	Keyword Tracking	19
3.2	Anforderungen	19
3.2.1	Funktionale Anforderungen	19
3.2.2	Nichtfunktionale Anforderungen	21
<b>4</b>	<b>Konzeption</b>	<b>24</b>
4.1	Systemarchitektur	24
4.1.1	Visualisierung	26
4.1.2	Datenspeicherung	28
4.1.3	Hintergrundprozesse	30
4.2	Fachliche Architektur	31
4.3	Außenansicht der Komponenten	32
4.3.1	SocialNews-Komponente	32

4.3.2	Facebook-Komponente .....	37
4.3.3	Twitter-Komponente .....	40
4.4	Innenansicht der Komponenten .....	44
4.4.1	SocialNews-Komponente.....	44
4.4.2	Such-Komponente .....	45
4.4.3	Facebook-Komponente .....	46
4.4.4	Twitter-Komponente .....	47
4.5	Fremdsoftware .....	48
4.6	Testkonzept .....	49
4.7	Nachrichten aus sozialen Medien .....	50
4.7.1	Nachrichtenbeschaffung .....	50
4.7.2	Nachrichtenaktualisierung.....	52
4.7.3	Limitierung der Datenanfragen .....	53
4.7.4	Suche der Nachrichten .....	55
4.7.5	Nachrichtenbereinigung.....	56
4.7.6	Zusammenfassung der Nachrichten .....	57
<b>5</b>	<b>Realisierung.....</b>	<b>58</b>
5.1	Realisierungsumfang .....	58
5.1.1	Prototyp des Systems .....	59
5.1.2	Abweichungen im Prototyp .....	60
5.2	Test.....	61
5.3	Bewertung .....	62
5.3.1	Erfüllung der Anforderungen.....	62
5.3.2	Bewertung des Prototyps .....	65
5.3.3	Zukünftige Perspektive .....	67
5.3.4	Methodische Abstraktion .....	68
<b>6</b>	<b>Fazit .....</b>	<b>69</b>
6.1	Zusammenfassung.....	69
6.2	Ausblick .....	70
	<b>Anhang A Inhalt der CD-ROM .....</b>	<b>72</b>
	<b>Glossar.....</b>	<b>73</b>

Abkürzungsverzeichnis .....	74
Abbildungsverzeichnis .....	75
Tabellenverzeichnis .....	76
Literaturverzeichnis .....	77

# 1 Einleitung

Soziale Medien sind fester Bestandteil des alltäglichen Lebens geworden. Ob im privaten oder beruflichen Umfeld, der Einsatz von sozialen Medien nimmt stetig zu.

Die sozialen Medien ermöglichen einen direkten Austausch der Nutzer miteinander. Dazu zählen auch Diskussionen zu aktuellen Geschehnissen in der Welt. Die zunehmende Benutzung der sozialen Medien führt zu einer immer größer werdenden Anzahl an Nachrichten und einer hohen Datenmenge innerhalb der sozialen Medien.

Durch diesen regen Austausch in den sozialen Medien wächst auch die Bedeutung der sozialen Medien für die Rundfunkanstalten im Bereich der Nachrichten.

Da Rundfunkanstalten bereits Anbindungen zu klassischen Nachrichtenquellen haben, ergeben sich aus diesen bereits Anwendungsszenarien für die sozialen Medien.

Es soll nach einer Möglichkeit gesucht werden, soziale Medien in den Newsworkflow einer Rundfunkanstalt zu integrieren.

## 1.1 Ziele

Das Ziel der Arbeit ist es, soziale Medien in den Newsworkflow einer Rundfunkanstalt zu integrieren. Zunächst sollen die bisherigen Anwendungsszenarien von Nachrichten aus den klassischen Medien beleuchtet werden. Die in den Anwendungsszenarien erläuterten Suchen sollen auf den Nachrichten der sozialen Medien angewendet werden können. Anhand der Anwendungsszenarien sollen Anforderungen an die Integration der sozialen Medien gestellt werden. Die Erfüllung der Anforderungen soll durch die Erstellung eines Entwurfs sichergestellt werden.

Es soll ein Entwurf konzipiert werden der Nachrichten aus unterschiedlichen sozialen Medien als Nachrichtenquelle verwenden kann.

Weiterhin sollen diese Nachrichten, durch die Verwendung von Suchkriterien, durchsucht werden können.

Anschließend soll der vorgestellte Entwurf durch eine prototypische Entwicklung realisiert werden.

## 1.2 Abgrenzungen

Durch die Fokussierung auf die Anwendungsfälle der Nachrichten und der damit verbundenen Ausrichtung auf den Nachrichtenerhalt und die anschließende Nachrichtensuche, wurden nichtfunktionale Anforderungen wie z.B. Sicherheit und Datenschutz nur teilweise berücksichtigt. Das System dient nicht zum Verwalten von Profilen auf sozialen Medien.

## 1.3 Struktur

Im **2. Kapitel** werden die grundlegenden Begriffe der Bachelorarbeit erläutert, um ein besseres Verständnis für die Arbeit zu vermitteln.

Im **3. Kapitel** wird eine Analyse der gegenwärtigen Situation durchgeführt. Zunächst wird erläutert, in welchem Szenario die Arbeit stattfindet und welche Anwendungsfälle das zu entwickelnde System zu erfüllen hat. Im Anschluss werden die Anforderungen an das System aufgestellt.

Aufbauend auf den aufgeführten Punkten aus dem vorherigen Kapitel, wird im **Kapitel 4** ein Entwurf zur Realisierung vorgeschlagen. Es wird ein Überblick über ein mögliches System und die benötigten Gegenstände zur Realisierung gegeben. Außerdem wird der Umgang mit den Nachrichten aus den sozialen Medien behandelt.

Im **5. Kapitel** wird betrachtet, in wie fern die in **Kapitel 4** behandelten Entwürfe, die in **Kapitel 3** geforderten Anforderungen erfüllen. Weiterhin wird Auskunft über die prototypische Entwicklung des Systems gegeben und in wie weit der Entwurf aus **Kapitel 4** realisiert werden konnte.

Im **6. Kapitel** wird die Arbeit zusammengefasst. Das Kapitel gibt weiterhin einen Ausblick auf weitere Möglichkeiten der Fortsetzung des in der Arbeit behandelten Themas.

## 2 Grundlagen

In diesem Kapitel soll ein grundlegendes Verständnis von sozialen Medien und Rundfunk geschaffen werden.

Beginnen wird dieses Kapitel mit dem **Abschnitt 2.1** Soziale Medien. In diesem Abschnitt wird zunächst der Begriff soziale Medien definiert und anschließend Arten von sozialen Medien vorgestellt.

Im darauffolgenden **Abschnitt 2.2** Rundfunk wird der Rundfunk definiert. Außerdem wird im **Abschnitt 2.3** der Begriff der Nachrichten erläutert und aus welchen Quellen sie kommen.

Der anschließende **Abschnitt 2.4** erklärt den Begriff der Dokumentenorientierte Datenbanken.

### 2.1 Soziale Medien

Soziale Medien oder auch Social Media ist ein Begriff der heutzutage inflationär benutzt wird und dadurch an Wirkung und Klarheit verliert. Um Klarheit über den Begriff Soziale Medien zu schaffen, wähle ich im Rahmen dieser Bachelorthesis die Definition des 25. Plenums vom 18. Februar 2013 in Frankfurt am Main:

„Social Media ist eine Vielfalt digitaler Medien und Technologien, die es Nutzern ermöglichen, sich auszutauschen und mediale Inhalte einzeln oder in Gemeinschaft zu gestalten. Die Interaktion umfasst den gegenseitigen Austausch von Informationen, Meinungen, Eindrücken und Erfahrungen sowie das Mitwirken an der Erstellung von Inhalten. Die Nutzer nehmen durch Kommentare, Bewertungen und Empfehlungen aktiv auf die Inhalte Bezug und bauen auf diese Weise eine soziale Beziehung untereinander auf. Die Grenze zwischen Produzent und Konsument verschwimmt. Diese Faktoren unterscheiden Social Media von den klassischen Medien. Als Kommunikationsmittel setzt Social Media einzeln oder in Kombination auf Text, Bild, Audio oder Video und kann plattformunabhängig stattfinden.“

(Quelle: [NEON 2013])

Wie dieser Definition zu entnehmen ist, besteht die Interaktion in soziale Medien vor allem aus den folgenden genannten Punkten:

- Austausch von Informationen, Meinungen, Eindrücken und Erfahrungen
- Erstellung von medialen Inhalten (einzeln oder in Gemeinschaft)
- Aufbau sozialer Beziehungen

Diese Interaktionen sind stark in den sozialen Media ausgeprägt. Anders sieht das bei klassischen Medien aus. Dort beschränkt sich das Maß an Interaktionen mit dem Konsumenten.

In den klassischen Medien sind Produzenten und Konsumenten stark getrennt, was die Gestaltung der Inhalte und den Austausch zwischen ihnen schwieriger macht.

Anders sieht das bei den sozialen Medien aus, deren Schwerpunkt meist auf Austausch basiert und somit eine Gestaltung von Inhalten durch andere ermöglicht.

Soziale Medien werden teils von Millionen Menschen genutzt. In den folgenden Abbildungen werden verschiedene soziale Plattformen gegenübergestellt. Dabei werden ihre Marktanteile nach Seitenabrufen betrachtet.

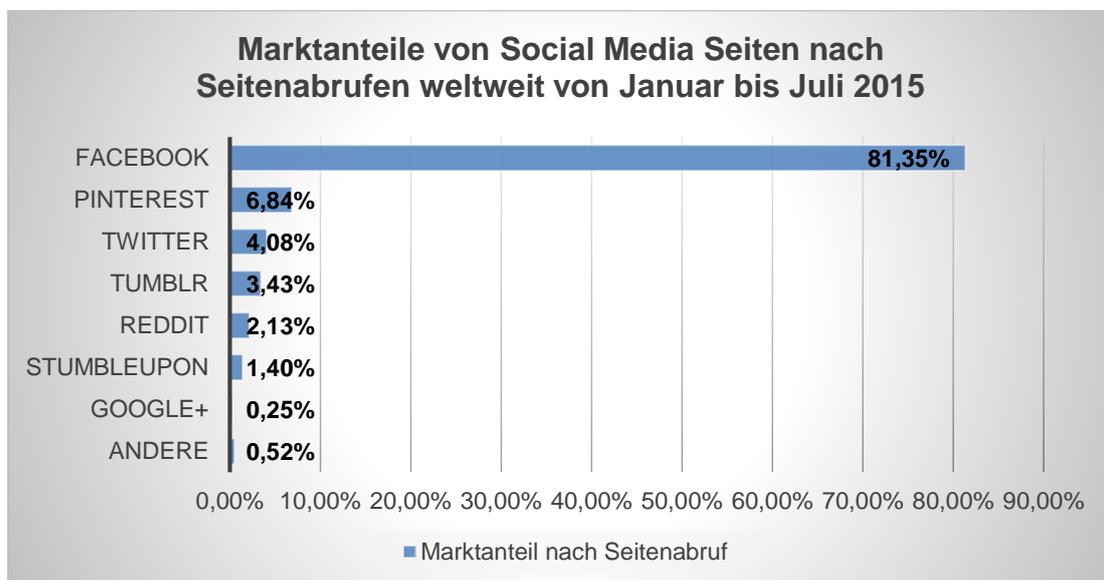


Abbildung 1 Marktanteile (weltweit) von Social Media Seiten (in Anlehnung an: [StatCounter 2016a])

In Abbildung 1 ist der Marktanteil von Social Media Seiten, anhand ihrer weltweiten Seitenabrufe von Januar bis Juli 2015 zu sehen. Die X-Achse beschreibt dabei die Marktanteile in Prozent und die Y-Achse die jeweiligen Plattformen.

Wie wir sehen können, hat Facebook den größten Marktanteil mit 81,35%. Die restlichen Seiten sind dicht beieinander.

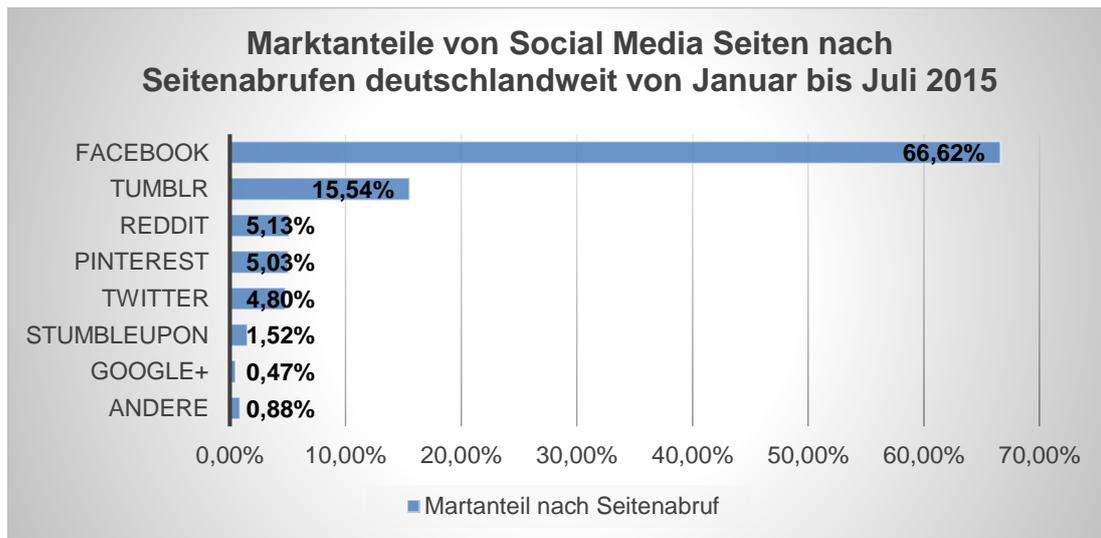


Abbildung 2 Marktanteile (deutschlandweit) von Social Media Seiten (in Anlehnung an: [StatCounter 2016b])

In Abbildung 2 Marktanteile anhand der Seitenabrufe deutschlandweit gegenübergestellt. Auf der X-Achse finden wir die Marktanteile der jeweiligen Plattformen und auf der Y-Achse die Plattformen. Zu erkennen ist, dass sowohl weltweit, als auch deutschlandweit, Facebook den größten Marktanteil hat. Tumblr besitzt neben Facebook als einzige Plattform einen zweistelligen Marktanteil.

### 2.1.1 Netzwerke

Die Kategorie der Netzwerke findet man auch häufig unter den Namen „soziale Netzwerke“ oder „Online-Communities“. In diesen Netzwerken werden, wie der Name es schon vermuten lässt, Personen bzw. Dinge miteinander vernetzt. Ein Nutzer der eine Plattform dieser Kategorie benutzt und sich dort registriert, wird häufig ein individuelles Profil zur eigenen Person erstellen.

Dieses Profil wird dann mit personenbezogenen Information versehen. Dies können Interessen, Vorlieben, Kontaktinformationen oder aber auch Bilder o.ä. sein, die im Zusammenhang mit der Person und damit dem Profil stehen. Auf diesen Plattformen gibt es ein „Beziehungssystem“ mit dem ein Nutzer die Möglichkeit hat, sich mit anderen Nutzern in Verbindung zu setzen, sofern die Verbindung vom andern bestätigt wird. Oft gebrauchte Bezeichnungen für solche Verbindungen und somit Beziehungen, sind dabei „Freunde“ oder „Kontakte“.

Die Plattformen bieten außerdem die Möglichkeit an, sich mit seinen Kontakten oder aber auch mit fremden Nutzern, durch direkte Nachrichten, themenspezifische Gruppen o.ä., auszutauschen (vgl. [Schmidt 2013], S.11).

Plattformen die zur Kategorie der Netzwerke gehören sind u.a. Facebook, Google+, Myspace und XING.

Im Folgenden wird das soziale Netzwerk Facebook näher betrachtet:

### **Nutzersicht:**

Am 4. Februar 2004 ging das Netzwerk von Mark Zuckerberg, Dustin Moskovitz, Chris Hughes und Eduardo Saverin zum ersten Mal online. (vgl. [Facebook 2016a])

Facebook ist ein Netzwerk auf dem jede private oder öffentliche Person ein Profil anlegen kann.

Auch Unternehmen, Marken und Organisationen können sich auf Facebook ein Profil anlegen. In Facebook werden Profile für Unternehmen, Marken und Organisationen „Page“ bzw. Facebook-Seiten genannt.

Die Profile auf Facebook können durch benutzerbezogene Informationen erweitert werden. Sozialer Austausch entsteht zwischen diesen Profilen durch Beiträge. Beiträge können von jedem Profil aus erstellt werden. Beiträge auf Facebook werden „Post“ genannt.

Ein Post kann aus Links, Bildern, Videos, Umfragen oder Texten bestehen.

Auf diese Posts können andere Nutzer kommentieren, den „Gefällt Mir“-Button drücken und diesen teilen.

Durch das Kommentieren, „Gefällt Mir“-Drücken und das Teilen von Beiträgen können neue Inhalte entstehen. So können Diskussionen zu einem Thema im Kommentarbereich zu neuen Diskussionen führen, die wiederum andere Themen behandeln können.

Wird der „Gefällt Mir“-Button gedrückt, wird positives Feedback gegeben und eine Verbindung zum Gegenstand erstellt (vgl. [Facebook 2016d]). Diese neu entstandene Verbindung wird öffentlich kommuniziert und kann von jeder bereits vorhandenen Verbindung gesehen werden. Solche Gegenstände können u.a. Pages oder Posts sein.

Wird der Teilen-Button gedrückt so kann ein Nutzer den Beitrag weiterleiten und somit seine Reichweite erhöhen. (vgl. [Facebook 2016c])

Hinzukommt dass jeder Nutzer eine zentrale Sammelstelle von Beiträgen besitzt, den „News Feed“. Auf diesem „News Feed“ erhält der Nutzer eine Übersicht aktueller Beiträge aller Profile, mit denen er in Beziehung steht. (vgl. [Facebook 2016b])

### **Entwicklersicht:**

Mittels der von Facebook angebotenen Graph API lassen sich verschiedenste Nutzerdaten von Facebook abrufen. Um die Graph API zu verwenden ist ein Entwicklerkonto auf Facebook notwendig. Die aktuelle Version ist v2.5 (Stand 07.01.2016). (vgl. [Facebook 2016g])

Alle öffentlichen Post von Facebook-Seiten sind über die API abrufbar, dies gilt für die von der Page selbst erstellten, als auch für die von Nutzern erstellten Posts und den dazugehörigen Kommentaren und „Gefällt-Mir“-Angaben.

Eine Anwendung kann die Timeline (Sammlung von persönlichen Daten) eines Nutzers nur mit dessen Einwilligung auslesen.

Die API ermöglicht u.a. die Suche nach Nutzern, Pages und Gruppen. Seit der Version v2.0 ist das Suchen nach Post nicht mehr möglich.

Eine Limitierung der Anzahl der API Aufrufe wird von Facebook in seltenen Fällen erteilt und umfasst 200 API Aufrufe pro User alle 60 Minuten. (vgl. [Facebook 2016i])

### 2.1.2 Multimedia

Auf den Multimediaplattformen geht es, anders als bei den Netzwerken, nicht um die einzelnen Personen, sondern eher um die Inhalte, die veröffentlicht werden. Diese Inhalte können Videos, Bilder oder Musik sein. Ein Austausch entsteht, wenn nach dem Hochladen der Inhalte andere Nutzer diese kommentieren, bewerten oder aber auch miteinander kommunizieren (vgl. [Schmidt 2013], S.12). Durch diesen Austausch kann ein gemeinschaftliches gestalten von weiteren Inhalten entstehen, wenn zum Beispiel Themenvorschläge für den nächsten Inhalt gesammelt werden.

Soziale Beziehungen sind in Multimediaplattformen weniger direkt als in Netzwerken. In den Multimediaplattformen können einzelne Nutzer andere Nutzer abonnieren. Je nach Plattform und Einstellung des Abonnierten, ist ein Bestätigen dieses Abonnement nicht erforderlich. Durch die Funktion des Abonnements wird der Nutzer benachrichtigt, falls ein abonniertes Nutzer einen neuen Inhalt hochlädt.

Plattformen die zur Kategorie der Multimediaplattformen gehören, sind für Videos z.B. Youtube, bei Bildern z.B. Flickr oder Musik z.B. Soundcloud.

### 2.1.3 Blogs

Eine weitere Kategorie der sozialen Medien sind ist der Blog. Blogs bestehen aus einzelnen Beiträgen die chronologisch rückwärts geordnet sind, d.h. der neuste Beitrag steht oben. Jeder Beitrag kann kommentiert und verlinkt werden, wodurch ein Austausch entsteht, wie in der Definition von sozialen Medien vorgegeben. In den einzelnen Beiträgen entstehen dadurch weitere Diskussionen, die wiederum das Verlinken andere Blogbeiträgen mit sich ziehen können. Dies hat zur Folge, dass Konsumenten den Blog indirekt mitgestalten können.

Blogs und ihre Beiträge sind inhaltlich unabhängig, nur das Thema eines Blogs kann den Inhalt der Beiträge beschränken. Dies geht jedoch vom Nutzer bzw. Urheber des Blogs selber aus (vgl. [Schmidt 2013], S.12).

Eine besondere Art der Blogs sind Microblogs, zu der u.a. die Plattform Twitter gehört. In Microblogs ist es dem Nutzer nur durch kurze Textbeiträge möglich, Inhalte zu schaffen und somit im Austausch mit anderen zu stehen. Oft werden solche Texte auch Updates bzw. Status-Updates genannt (vgl. [Schmidt 2013], S.13).

Im Folgenden wird die Microblogging Plattform Twitter näher betrachtet:

**Nutzersicht:**

Twitter ist eine Microblogging Service mit ca. 320 Millionen aktiven Nutzern pro Monat. Seit der Gründung im März 2006 und dem ersten Tweet am 21. März vom Gründer Jack Dorsey (vgl. [Twitter 2016a] u. [Twitter 2016b]) hat sich Twitter zu einen der größten Websites weltweit entwickelt.

Beiträge auf Twitter werden Tweets genannt und können jedes Thema behandeln. Ein Tweet ist eine Kurznachricht und wesentlicher Bestandteil von Twitter, dessen Länge auf maximal 140 Zeichen beschränkt ist. Twitter bietet außerdem die Möglichkeit Fotos, Videos oder Umfragen an einem Tweet anzuhängen. Dies hat eine geringe Anzahl an Textzeichen im Beitrag zur Folge.

Der Austausch auf Twitter erfolgt durch das Retweeten, Liken und Antworten auf Tweets anderer Nutzer.

Die Beiträge anderer Nutzern kann man abonnieren, auf Twitter wird diese Funktion „follow“ genannt. Durch das Abonnieren eines anderen Nutzers wird man zu dessen Follower. Diese Beziehung erfordert nicht die Bestätigung des „gefollowten“.

Das Hashtag (Symbol: #) ist auf Twitter ein beliebtes Mittel um einem Tweet einem Thema einzuordnen oder Schlagwörter zu markieren. Aufgrund dieser Eigenschaft sind Hashtags oft in den Trends von Twitter vertreten (vgl. [Twitter 2016d]).

Die Trends in Twitter werden durch einen Algorithmus von Twitter bestimmt. Dieser Algorithmus identifiziert die Themen, die aktuell beliebt sind (vgl. [Twitter 2016g]).

**Entwicklersicht:**

Um Zugriff auf die Daten von Twitter zu bekommen, ist ein Entwicklerkonto auf Twitter notwendig. Die folgenden APIs (Version 1.1) bietet Twitter an:

**Rest API:**

Mit der Rest API ist es möglich, auf die Daten von Twitter lesend und schreibend zuzugreifen. Mithilfe der Rest API ist es u.a. möglich Tweets zu schreiben, aber auch Profildaten verschiedener Nutzer auszulesen (vgl. [Twitter 2016h]).

Ein weiterer Bestandteil der REST API ist die Search API.

**Search API:**

Die Search API ist ein Teil der Rest API. Durch die Search API kann mittels Queries auf Twitter nach Daten gesucht werden. Twitter selber merkt dabei an, dass die Search API sich mehr auf Relevanz, als auf Vollständigkeit der Tweets fokussiert ist. Die Search API ermöglicht es nach bereits vergangenen Tweets, die gewöhnlich nicht älter als eine Woche sind, zu suchen (vgl. [Twitter 2016i]).

**Streaming API:**

Die Streaming API gibt dem Entwickler Zugang zu Twitters globalen Stream von Tweet Daten. Damit ist es möglich, Tweets in Echtzeit zu beobachten und zu verarbeiten (vgl. [Twitter 2016j]).

## 2.2 Rundfunk

Der Rundfunk ist ein linearer Informations- und Kommunikationsdienst und ist für die Allgemeinheit vorgesehen. Der Rundfunk bietet der Allgemeinheit ein Angebot von Inhalten als Bewegbild oder Ton entlang eines Sendepfades an. Der Allgemeinheit werden diese Inhalte zum zeitgleichen Empfangen unter Benutzung von elektromagnetischen Schwingungen bereitgestellt (vgl. [Rund 2016] §2, Abs. 1).

Der Rundfunk ist eine Untergruppe der klassischen Medien, die sich in Print- und elektronische Medien unterteilen (vgl. [Klein 2012], S.18).

In Deutschland gehören Hörfunk und Fernsehen zu den am häufigsten und am längsten genutzten Medien (vgl. [Beck 2012], S.178). Dabei handelt es sich um tertiäre Medien, da beim Sender so wie beim Empfänger eine technische Einrichtung erforderlich ist (vgl. [Beck 2012], S. 180).

Nur im deutschen Sprachraum gibt es die Unterteilung von Radio und Hörfunk. Zwar werden diese zwei Worte öfters als Synonym verwendet, trotzdem werden sie auch differenziert. Unter Radio wird das Empfangsgerät verstanden, unter Hörfunk jedoch das gesamte empfangene Programm (vgl. [Klein 2012], S.20).

## 2.3 Nachricht

Eine Nachricht ist eine Folge von Zeichen oder kontinuierlichen Funktionen, die Informationen darstellen. Ihr vorrangiges Ziel ist es, diese Informationen einem Empfänger zu vermitteln (vgl. [IT Wissen 2016a]). Die Nachricht hat keinerlei Einfluss auf den Informationsgehalt der übermittelten Zeichen.

Informationen sind Bestandteil einer Nachricht und geben Auskunft darüber, wie bedeutend eine Nachricht ist. Die Informationen können sowohl textlich, grafisch als auch audiovisuell vorliegen (vgl. [IT Wissen 2016b]).

Eine Information ist der erfolgreiche Transfer von Wissen. Dabei führt dieser Transfer von neuem Wissen zu einem neueren Wissenstand beim Empfänger der Information (vgl. [Hans 1997]).

Eine zentrale Quelle für neue Informationen und somit Nachrichten sind die Nachrichtenagenturen. Nachrichtenagenturen übernehmen die wichtige Rolle der Selektionsfunktion, da sie als einer der ersten im Kommunikationsprozess auswählen, welche Nachrichten über welche Ereignisse verfasst und verbreitet werden. Nachrichtenagenturen gelten als primäre Lieferanten von Nachrichten. Aufgrund von Kostengründen kann es vorkommen, dass vollständige Beiträge unverändert in Zeitungen oder in das Rundfunkprogramm übernommen werden.

Nachrichten, die nicht durch Nachrichtenagenturen veröffentlicht wurden, gelangen nur in den Medien und somit an die Öffentlichkeit, wenn ihre Korrespondenten selbst davon erfahren (vgl. [Beck 2012], S.45).

Nachrichten in den sozialen Medien können unterschiedlich auftreten. Im weiteren Verlauf der Arbeit werden Beiträge in sozialen Medien, die öffentlich sichtbar sind, als Nachrichten bezeichnet.

## 2.4 Dokumentenorientierte Datenbank

Die dokumentenorientierten Datenbanken gehören zu den NoSQL Datenbanken. Sie speichern die Daten nicht in Tabellen, sondern in Dokumenten. Ein Dokument besteht aus einer strukturierten Zusammenstellung von Daten.

Die Daten in den Dokumenten werden mittels Key / Value – Paaren gespeichert.

Jedem Key, also dem Schlüssel, wird ein Wert zugewiesen. Werte sind in diesem Fall nicht einfache Datentypen, sondern beliebige Informationen. Bei dem Wert kann es sich auch um eine Liste handeln, die wiederum selbst weitere Informationen enthält. Eine vorgeschriebene Struktur der Daten ist nicht vorhanden. Dadurch kann in jedem Dokument eine andere Struktur vorgefunden werden. Man spricht in diesem Zusammenhang von Schemafreiheit, da für die Dokumente in der Datenbank kein Schema vorgeschrieben ist. Durch die Gestaltungsfreiheit der Dokumente entfallen bewährte Funktionen, die in relationalen Datenbanken vorhanden sind. Im Anwendungsprogramm müssen Funktionen, wie die Überprüfung des richtigen Datentyps der Datenwerte, implementiert werden.

In dokumentenorientierten Datenbanken können Daten die zusammengehören auch zusammen in einem Dokument gespeichert werden. Dadurch entfällt das Abfragen von mehreren Tabellen, wie es in den relationalen Datenbanken üblich ist. Das Speichern von Daten die zusammengehören war einer der Grundgedanken bei der Entwicklung von dokumentenorientierten Datenbanken.

Ein überwiegender Einsatz der dokumentenorientierten Datenbanken ist im Bereich von Web-Applikationen vorzufinden (vgl. [Datenbanken 2016]).

## 3 Analyse

In diesem Kapitel werden die Anwendungsszenarien einer Rundfunkanstalt und deren Umgang mit Nachrichten beleuchtet. Im **Abschnitt 3.1** wird beschrieben in welchem Rahmen die Integration der sozialen Medien stattfinden soll. Im darauffolgenden **Abschnitt 3.2** werden, basierend auf den Anwendungsszenarien, die funktionalen- und nichtfunktionalen Anforderungen an das System erläutert.

### 3.1 Szenario

Im News-Bereich einer Rundfunkanstalt wird mittels eines Newsmoduls der Eingang der Nachrichten aus verschiedensten Quellen überwacht. Zu den Quellen gehören klassische Agenturen wie die dpa oder aber auch Web-Quellen wie RSS-Feeds. Die Nachrichten liegen im NewsML-Format vor.

Mittels des Newsmoduls ist es dem Moderator möglich, einen Überblick über aktuelle, als auch über ältere Geschehnisse zu erhalten. Mittels Suchinterfaces zum Filtern von Nachrichten und der Möglichkeit mehrere parallelaufende Suchen zu starten, kann der Moderator das gewünschte Sendungsprogramm gestalten.

Das System indiziert alle eingehenden Nachrichten und speichert sie in einer Datenbank ab, wodurch das System in der Lage ist, Suchen über den Datenbestand durchzuführen. Die Entwicklungsplattform des Systems ist das .Net Framework<sup>1</sup>.

Durch eine neue Komponente soll dieses System nun durch soziale Medien erweitert werden. Die Integration der sozialen Medien erfolgt, indem die bereits vorhandenen Quellen des Newsmoduls um Nachrichten aus sozialen Medien erweitert werden. Die Arbeit wird in Zusammenarbeit mit einer Rundfunkanstalt realisiert, deren Hauptziel eine eigene Implementierung von verschiedenen sozialen Medien und der anschließenden Suchen ist.

Durch das gegebene Newsmodul der Rundfunkanstalt ergeben sich zwei Anwendungsszenarien. Diese beiden Anwendungsszenarien sollen nun durch Integration von sozialen Plattformen erweitert werden.

In Abbildung 3 werden die beiden Anwendungsszenarien abgebildet und im folgendem erläutert.

---

<sup>1</sup> <https://www.microsoft.com/net/>

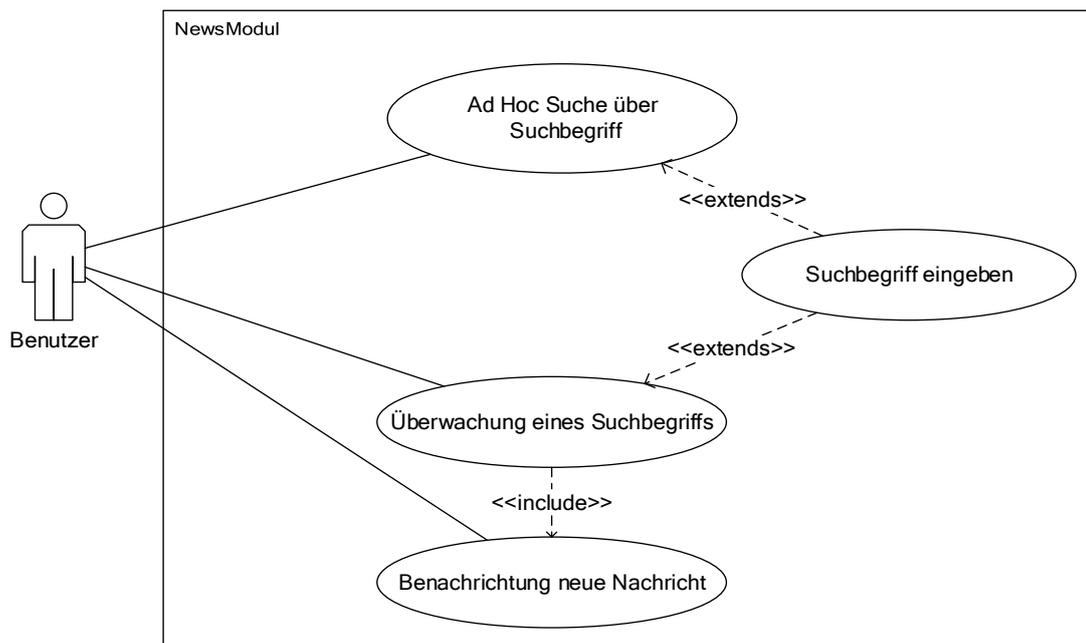


Abbildung 3 Anwendungsfälle des Newstools

### 3.1.1 Ad-Hoc-Suche

Die Ad-Hoc-Suche bietet die Möglichkeit einer schnellen Suche über die vorhandenen Nachrichten einer Rundfunkanstalt. Diese Art der Suche bietet dem Nutzer eine schnelle Suche, mit deren Hilfe der Nutzer sich über gewünschte Themen informieren und recherchieren kann.

Um diese Ad-Hoc-Suche zu starten ist es erforderlich, dass der Nutzer einen Suchbegriff eingibt.

Zusätzlich zum Suchbegriff können in einem Interface weitere Suchfilter eingestellt werden, um das Suchergebnis zu manipulieren.

Die Suchfilter bieten dem Nutzer u.a. die Möglichkeit Suchqueries zu bilden, mit denen die Volltextsuche präziser die Nachrichten filtern kann, z.B. durch das Ausschließen von Begriffen. Eine weitere Möglichkeit wäre z.B. das Angeben eines Zeitraumes, indem eine Nachricht erstellt worden sein muss.

Sind alle Einstellungen vorgenommen worden, durchsucht die darauffolgende Volltextsuche alle vorhandenen Nachrichten nach dem gewünschten Suchbegriff und Suchfiltern. Nach dem Ende der Suche wird dem Benutzer die Ergebnismenge dargestellt. Nachrichten mit einer hohen Priorität oder Wichtigkeit z.B. Eilmeldungen zu Terrorwarnungen, werden farblich hervorgehoben.

Durch die Integration der sozialen Plattformen soll es weiterhin möglich sein eine Ad-Hoc-Suche durchzuführen. Der Ablauf ist der gleiche nur mit dem Unterschied, dass die Nachrichten von den sozialen Plattformen stammen. Daraus ergeben sich Anforderungen an das System die im **Abschnitt 3.2.1** erläutert werden.

### 3.1.2 Keyword Tracking

Das Keyword Tracking bzw. die Überwachung der Nachrichten ähnelt der Ad-Hoc-Suche sehr. Auch hier wird mittels eines Suchbegriffes die Volltextsuche gestartet und die Ergebnismenge durch optionale Suchfilter manipuliert. Anders als bei der Ad-Hoc-Suche jedoch, möchte der Benutzer hier sich über ein Thema fortlaufend informieren.

Die angegebenen Suchbegriffe samt Suchfilter werden als sogenannte Suchvorlage gespeichert.

Der Benutzer erhält, genau wie bei der Ad-Hoc-Suche, eine Ergebnismenge mit den aktuellen Nachrichten. Der Unterschied zur Ad Hoc Suche liegt darin, dass man beim Eingang einer neuen Nachricht, die nach dem Start der aktuellen Suche eingetroffen ist, eine Benachrichtigung erhält.

Ein weiterer Unterschied zur Ad-Hoc-Suche ist die Tatsache, dass nach dem Start der Überwachung der Nachrichten, die Suche im Hintergrund weiterläuft. Der Hintergrundprozess wartet auf neue Nachrichten und überprüft diese anhand der zuvor eingegebenen Suchkriterien. Wenn eine Nachricht eingetroffen ist, wird diese der Ergebnismenge hinzugefügt.

Eine Benachrichtigung informiert den Nutzer über den Erhalt einer neuen Nachricht.

Auch dieser Anwendungsfall soll beim Überwachen von Suchbegriffen auf Nachrichten aus sozialen Medien zugreifen können. Die Anzahl der Nachrichten, die täglich durch die sozialen Medien eintreffen, sind um ein vielfaches höher, als in den klassischen Medien. Bei der Überwachung sind plattformspezifische Limitierungen bei der Anzahl der Datenanfragen zu berücksichtigen. Dadurch soll ein Stillstand von eintreffenden Nachrichten zu vermieden werden. Deswegen sollen die Datenanfragen einer jeden sozialen Plattform zeitlich periodisch ausgeführt werden.

## 3.2 Anforderungen

Aufgrund der im **Abschnitt 3.1** erläuterten Szenarien werden in diesem Abschnitt die funktionalen, sowie die nichtfunktionalen, Anforderungen an das System erläutert, die für die Integration von sozialen Medien in den Newsworkflow erforderlich sind.

### 3.2.1 Funktionale Anforderungen

Die Funktionalen Anforderungen beschreiben, welche Funktionen das System erfüllen muss. Diese Anforderungen ergeben sich aus den im **Abschnitt 3.1** erläuterten Anwendungsszenarien. Die funktionalen Anforderungen sind:

#### 1. Verknüpfung zu soziale Plattformen

Um Zugriff auf die Daten von sozialen Plattformen zu erhalten, ist es erforderlich, auf jeder der verwendeten Plattformen ein Konto zu besitzen.

Dem Nutzer soll es möglich sein, sich über die Software mit den sozialen Plattformen, zu denen er ein Konto besitzt, zu verbinden. Dabei soll das System die jeweiligen Zugriffsprotokolle der Plattformen verwenden, um einen reibungslosen Ablauf zu gewährleisten. Um die Anzahl der Wiederanmeldungen zu verringern, soll das System Anmeldedaten speichern.

## **2. Eingeben von Suchkriterien**

Der Nutzer kann durch die Eingabe von Suchkriterien das Suchen und Laden von Nachrichten manipulieren. Dazu zählen die Wörter, die in einer Nachricht vorhanden sein sollen oder nicht, der Zeitraum der Erstellung der Nachricht, die sozialen Medien die nach Nachrichten durchsucht werden sollen und die Aktualität der Nachricht.

Das System soll auf diese Suchkriterien entsprechend reagieren, die Suche anpassen und die Suchergebnisse sollen diese Kriterien erfüllen.

## **3. Mehrere Suchen erstellen**

Das System soll in der Lage sein, mehrere Suchen zu erstellen und gleichzeitig zu überwachen. Neben parallellaufenden Nachrichtenüberwachungen, soll es auch möglich sein die temporären Ad-Hoc-Suchen zu starten.

## **4. Laden von Nachrichten aus Sozialen Medien**

Das System soll in der Lage sein, Nachrichten aus den sozialen Plattformen runterzuladen. Das Laden von Nachrichten aus einer sozialen Plattform soll unabhängig von anderen Plattformen stattfinden.

Aufgrund der unterschiedlichen Beschaffenheit jeder sozialen Plattform kann es erforderlich sein, dass Nachrichten aus sozialen Medien vor der eigentlichen Suche schon geladen werden müssen. Weiterhin muss jede Plattform darauf achten, die Limitierung der Anzahl an Datenfragen einzuhalten.

Ist eine Suche eine Tracking Suche, soll in periodischen Abständen überprüft werden, ob neue Nachrichten vorhanden sind.

## **5. Speichern von Nachrichten und Suchen**

Um einen schnellen Zugriff auf die Nachrichten der sozialen Medien zu haben und um wiederholende Anfragen zu vermeiden, ist es erforderlich, Nachrichten in einer Datenbank zu speichern.

Des Weiteren soll das System aktive Suchen bzw. Überwachungen des Nutzers speichern, damit beim Wiederöffnen der Software diese Suchen wiederhergestellt werden und auf den aktuellen Stand gebracht werden können.

## **6. Benachrichtigung bei Nachrichteneingang**

Das System soll den Nutzer benachrichtigen, wenn in einer seiner überwachten Nachrichtensuchen mindestens eine neue Nachricht eingegangen ist.

## **7. Nachrichten anzeigen und vergleichen**

Das System soll alle gefundenen Nachrichten anzeigen. Je nach Einstellung können wichtige farblich gekennzeichnet werden. Die Nachrichten werden in umgekehrter chronologischer Reihenfolge angezeigt, damit die neusten Meldungen, z.B. in einer Liste, oben stehen. Außerdem soll eine Möglichkeit bestehen, die Nachrichten aus unterschiedlichen sozialen Medien vergleichen zu können.

## **8. Aktualisieren der Nachrichten**

Das System soll in regelmäßigen Abständen oder nach Bedarf die vorhandenen Nachrichten auf Aktualität prüfen. Sind Nachrichten geändert worden, sollen diese Änderungen auch in den vom System gespeicherten Nachrichten übernommen werden.

## **9. Löschen von nichtrelevante Nachrichten**

Das System soll Nachrichten, die nicht relevant sind, aus der Datenbank löschen können. Welche Nachrichten relevant sind und welche nicht, soll das System selbst entscheiden. Die Entscheidung des Systems basiert auf einer vom Nutzer konfigurierbaren Datei.

### **3.2.2 Nichtfunktionale Anforderungen**

Neben den funktionalen Anforderungen sind auch nichtfunktionale Anforderungen erforderlich, um die Qualität an der Software zu gewährleisten. Orientiert wurde sich dabei an die ISO 9126 (siehe [John 2016]). Im Folgenden sind die nichtfunktionalen Anforderungen aufgelistet:

#### **1. Erweiterbarkeit und Wartbarkeit**

Dem Entwickler soll es möglich sein, neue Social Media Quellen hinzufügen oder bestehende Quellen zu ändern.

Jede integrierte Social Media Quelle besitzt ein eigenes Modul. Dadurch soll die Unabhängigkeit der Social Media Quellen untereinander gewährleistet werden.

Durch die Unabhängigkeit der Social Media Quellen zu einander soll Stabilität gewährleistet werden, falls eine Social Media Quelle ausfallen sollte.

Da soziale Medien stetig im Wandel sind, neue soziale Plattformen hinzukommen und es nicht gewährleistet ist, dass soziale Plattformen bestehen bleiben, soll es dem Entwickler möglich

sein, neue Plattformen implementieren, ändern, erweitern oder entfernen zu können. Durch definierte Schnittstellen soll dies dem Entwickler ermöglicht werden.

Um den Aufwand an Wartbarkeit und Erweiterbarkeit zu reduzieren, kann auf erprobte Software zugegriffen werden. Die Entwicklungsumgebung soll die gleiche sein, wie beim vorhandenen System.

## **2. Effizienz**

Das Zeitverhalten bei einer Suche spielt eine wichtige Rolle für den Nutzer. Nach Start einer Suche sollen dem Nutzer in weniger als 5 Sekunden die ersten Nachrichten angezeigt werden. Das System ist abhängig von den Antwortzeiten der sozialen Medien. Um die Auswirkungen dieser Abhängigkeit zu minimieren, soll das Laden von Meldungen aus den sozialen Plattformen individuell angepasst werden. Durch optimierte Abfragen oder vorzeitiges Laden von Nachrichten, vor der aktuellen Suchen, sollen Ressourcen gespart werden.

Bei einer zeitlichen Limitierung der Anzahl der Datenanfragen durch die jeweilige soziale Plattform, soll jede Komponente, die zur Verfügung stehende Anzahl, bis zur nächsten Wiederaufladung optimal verwenden.

## **3. Zuverlässigkeit**

Die Zuverlässigkeit des Systems muss in mehreren Punkten gewährleistet sein. Da die Social Media Komponente abhängig von den jeweiligen sozialen Plattformen ist, muss gewährleistet werden, dass bei einer Fehlermeldung nach einer Datenanfrage das System trotzdem noch funktioniert und die Abfrage zu einem späteren Zeitpunkt nochmal ausgeführt wird.

Zu jedem Zeitpunkt soll das System wissen, wann die letzte Nachricht jeder einzelnen Plattform eingetroffen ist. Das System soll tolerant gegenüber Änderungen der Datensätze aus den sozialen Medien sein.

## **4. Benutzbarkeit**

### **a. Nutzersicht:**

Die Verständlichkeit soll trotz der Integration durch Social Media beibehalten werden.

Da sich die neue Social Media Komponente bei der Bedienbarkeit am bereits vorhandenen Newstool orientiert, soll es dem Nutzer möglich sein, sich schnell zurecht zu finden.

Neue Funktionen, wie z.B. die nun erforderliche Verbindung zu den sozialen Plattformen, sollen dem Nutzer mitgeteilt werden.

### **b. Entwicklersicht:**

Es soll dem Entwickler möglich sein, neue Social Media Quellen hinzufügen oder zu entfernen. Dies soll durch definierte Schnittstellen ermöglicht werden.

## 5. Sicherheit

Auch Sicherheitsaspekte sollten berücksichtigt werden. Durch das Speichern von Nachrichten aus den sozialen Plattformen sind Maßnahmen zum Datenschutz erforderlich.

Die benutzerspezifischen Zugriffstoken, die durch das Verknüpfen des Systems mit den sozialen Medien generiert werden, müssen sicher hinterlegt und vor Missbrauch geschützt werden. Gleiches gilt für die Nachrichten der sozialen Medien, da diese empfindliche Daten enthalten können.

# 4 Konzeption

Anhand der im **Kapitel 3** genannten Anwendungsfälle und Anforderungen an das System, wird im Folgenden das Konzept zur Realisierung der Integration von sozialen Medien, in den Newsworkflow einer Rundfunkanstalt entworfen und vorgestellt. Am Anfang des Kapitels, im **Abschnitt 4.1**, erfolgt eine Konzeption einer Systemarchitektur. Es wird darauf eingegangen, welche Eigenschaften das zu entwickelnde System hat und wie diese umgesetzt werden soll. Im **Abschnitt 4.2** wird zusätzlich eine fachliche Architektur entworfen. Anschließend erfolgt eine Beschreibung der Außenansicht sowie der Innenansicht der zu Komponenten in **Abschnitt 4.3** und **4.4**. Der **Abschnitt 4.5** erwähnt die Fremdsoftware die genutzt wird. Ein Testkonzept für das System wird in **Abschnitt 4.6** erläutert. Abschließen wird dieses Kapitel mit **Abschnitt 4.7**, welches den Umgang mit den Nachrichten aus sozialen Medien für das System beschreibt.

## 4.1 Systemarchitektur

In das bereits vorhandene Newssystem einer Rundfunkanstalt soll ein weiteres System integriert werden, das für die Nachrichten aus sozialen Medien zuständig ist.

Ziel dieses System ist die Bereitstellung von Schnittstellen zum Hinzufügen von sozialen Medien und die Realisierung der im **Abschnitt 3.1** beschriebenen Suchen. Dabei ist das neue System autark vom alten Newssystem. Das neue System wird durch eine graphische Oberfläche im Newssystem bereitgestellt.

Die graphische Oberfläche dient als Bedienungselement für den Nutzer des Systems zum Verwalten seiner Verknüpfungen der sozialen Plattformen und seiner Suchen auf den Nachrichten der sozialen Medien.

Durch die Vielzahl an verschiedenen sozialen Medien wird jedes soziale Medium durch eine eigene Komponente realisiert. Diese Komponenten sind für das Herunterladen, Speichern und Verwalten der Nachrichten im Hintergrund zuständig. Das System wird exemplarisch anhand zwei sozialen Medien konzipiert.

Im Folgenden wird, anhand von Abbildung 4, eine Übersicht über das zu entwickelnde System gegeben.

Der Nutzer bedient das System durch die graphische Oberfläche.

Die dahinterliegende Such-Komponente dient als Schnittstelle für die graphische Oberfläche und den einzelnen Komponenten der sozialen Medien.

Das Back-End (in der Abbildung nicht dargestellt) stellt eine Verbindung zu den sozialen Medien her und sorgt für die Speicherung der Nachrichten in einer Datenbank. Dies soll durch Hintergrundprozesse gesehen. Die Hintergrundprozesse der einzelnen Komponenten der

sozialen Medien laufen im Hintergrund ab und sind für den Nutzer nicht sichtbar. Die Hintergrundprozesse werden im **Abschnitt 4.1.3** weiter behandelt. Die Komponenten die für ein soziales Medium verantwortlich sind, werden im weiteren Verlauf der Arbeit auch „soziale Komponenten“ genannt.

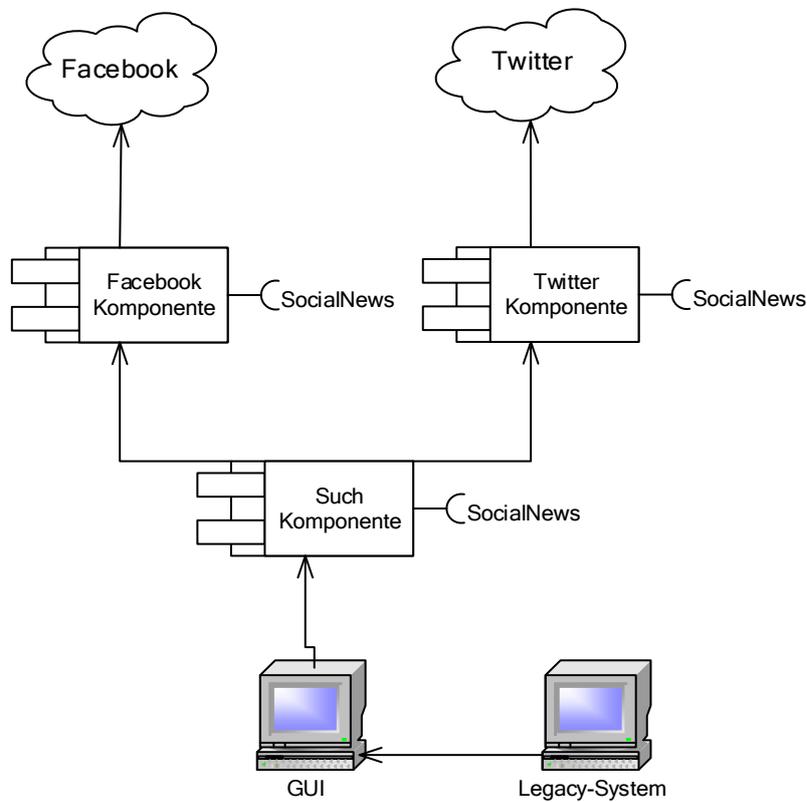


Abbildung 4 Übersicht des Systems

Die einzelnen sozialen Medien sollen in unabhängigen Komponenten realisiert werden. Die sozialen Medien agieren unabhängig voneinander und dies soll durch ihre Realisierung in unterschiedlichen Komponenten gewährleistet werden. Ziel soll eine lose Kopplung der Komponenten untereinander sein. Durch die lose Kopplung soll vermieden werden, dass sich Änderungen einer Komponente zu stark auf die anderen Komponenten auswirken. Die lose Kopplung unterstützt die Verwirklichung der in **Abschnitt 3.2.2** beschriebenen nichtfunktionalen Anforderungen der Wartbarkeit und Erweiterbarkeit, durch die daraus resultierende Stabilität. Des Weiteren wird die nichtfunktionale Anforderung der Zuverlässigkeit des Systems durch die lose Kopplung unterstützt, da sich Ausfälle oder Änderungen der sozialen Komponenten nicht auf andere auswirken.

Um diese Besonderheit des Systems zu gewährleisten, wurde das Entwurfsmuster der Fassade als Vorbild ausgewählt, welches die lose Kopplung der einzelnen Komponenten fördert. Das Fassaden-Entwurfsmuster bietet eine Schnittstelle, welches den Einsatz der Komponenten vereinfacht (vgl. [Gamma 1996], S.212).

Durch das Fassaden-Entwurfsmuster bleiben die Komponenten, die verantwortlich für die sozialen Medien sind, eigenständig. Gleichzeitig dient die Fassade als Bindungsglied der sozialen Komponenten. Weiterhin bietet das Entwurfsmuster der Fassade die Möglichkeit des Hinzufügens von weiteren sozialen Komponenten an.

Die Fassade wird durch die Such-Komponente realisiert. Sie besitzt das Wissen jeder Komponente und kann die entsprechenden Funktionen für die jeweiligen Anwendungsszenarien ausführen.

Durch das Entwurfsmuster ist es für den Nutzer nicht nötig jede Komponente einzeln anzusprechen, sondern nur die Such-Komponente.

Um eine Suche durchzuführen wird die Such-Komponente vom Nutzer benutzt. Wird eine Suche vom Nutzer gestartet, spricht die Such-Komponente die betroffenen sozialen Komponenten an und führt die entsprechenden Funktionen aus.

Die SocialNews-Komponente dient als gemeinsame Schnittstelle der Komponenten. Sie bietet die Funktionen, die die Komponenten benötigen, um die erforderten Anforderungen aus **Abschnitt 3.2** zu erfüllen.

#### 4.1.1 Visualisierung

Dieser Abschnitt gibt einen Einblick in die Benutzeroberfläche der GUI, die in der Systemarchitektur genannt wurde. Es wird nun ein WireFrame vorgestellt einer möglichen Visualisierung einer GUI-Komponente. Die Visualisierung beschränkt sich dabei auf die grafische Darstellung der Komponente.

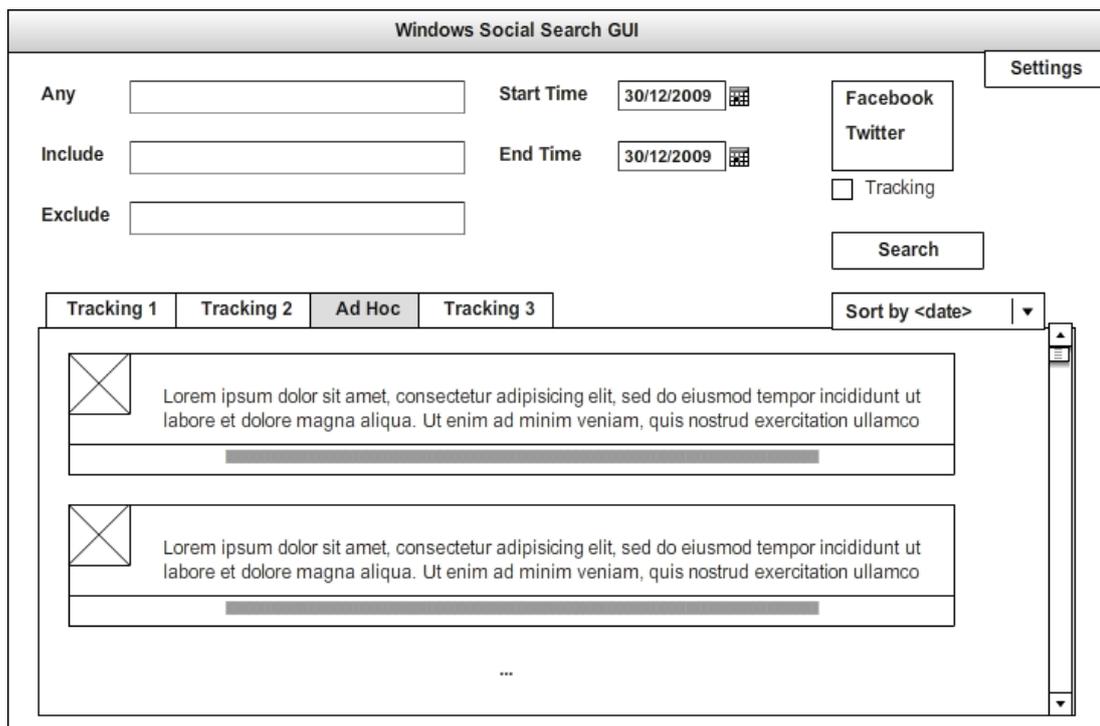


Abbildung 5 Wireframe einer möglichen grafischen Benutzeroberfläche des Systems

Die Abbildung 5 zeigt eine Möglichkeit einer grafischen Benutzeroberfläche des Systems. Die Benutzeroberfläche zeichnet sich vor allem durch die Eingabe von Suchkriterien sowie der Darstellung von Suchergebnissen aus. Dies sind die grundlegenden Funktionen die in **Abschnitt 3.2.1** gefordert wurden.

Durch den Settings-Button sollen die Verknüpfungen zu den sozialen Medien möglich sein. Das Drücken des Settings-Buttons soll eine Auflistung der verfügbaren sozialen Medien zeigen und die Möglichkeit geben, sich mit diesen zu verbinden, um Suchen auf diesen Medien zu ermöglichen.

Im Select-Fenster stehen die verfügbaren sozialen Medien zur Auswahl. Dort wird für die Suche entschieden aus welchen sozialen Medien, die Nachrichten gesucht werden sollen. Eine Mehrfachauswahl ist hier möglich.

Durch die Checkbox Tracking wird dem Nutzer die Auswahl der, in **Abschnitt 3.1** beschriebenen, Suchen ermöglicht. Wird die Checkbox nicht ausgewählt, erfolgt die Ad-Hoc-Suche.

Das Zeitfenster in dem eine Nachricht erstellt worden sein muss, wird durch die Auswahl der „Start Time“ sowie der „End Time“ entschieden.

In den Feldern rechts neben „Any“, „Include“ und „Exclude“ kommen die Begriffe, nach denen gesucht werden soll. „Any“ steht für beliebige Worte, „Include“ für enthaltene Worte und „Exclude“ für auszuschließende Worte. Nähere Details der Suche werden in **Abschnitt 4.7.4** erläutert.

Der untere Teil der Benutzeroberfläche ist für die Darstellung der Suchen und deren Suchergebnisse vorgesehen. Als Navigationselement für die Suchen eines Nutzers des Systems wurde für diese Abbildung Tabs gewählt.

Durch eine Dropdown-Liste soll es dem Nutzer möglich sein die gefundenen Nachrichten durch unterschiedliche Sortierverfahren sortieren zu können.

Eine mögliche der Darstellung der Nachricht ist in Abbildung 5 auch zu erkennen. Bei der Darstellung sollten nach Möglichkeit, der Inhalt und die Quelle des sozialen Mediums sichtbar sein. Der Inhalt ist in dieser Abbildung als Text und die Platzierung der Quelle als Quadrat mit einem X dargestellt.

Auf der Abbildung ist unter dem Inhalt der Nachricht ein grauer Block zu erkennen, dieser kann genutzt werden, um den Erstellzeitpunkt, den Updatezeitpunkt, Anzahl der „Gefällt-Mir“ oder Kommentare oder andere plattformspezifische Daten einer Nachricht anzeigen zu lassen.

Die funktionale Anforderung aus dem **Abschnitt 3.2.1**, Benachrichtigung bei Nachrichteneingang, soll durch das Hervorheben der Suchtabs erfolgen. Möglich wären hier eine Angabe der Anzahl der neuen Nachrichten und einer farblichen Hervorhebung der Suchtabs.

Eine weitere alternative Funktionalität der Benutzeroberfläche könnte die Möglichkeit einer Detailansicht der Nachrichten sein.

Die Einbindung der GUI durch das Altsystem kann sowohl als Client Anwendung oder aber als Webschnittstelle, welches eine höhere Wiederverwendbarkeit aufweist, erfolgen.

#### 4.1.2 Datenspeicherung

Nach dem Herunterladen der Nachrichten aus den sozialen Medien durch die dafür verantwortlichen Komponenten, werden die Nachrichten in einer Datenbank gespeichert. Die Nachrichten werden gespeichert, um sie auch für spätere Suchen bereitzustellen. Durch die Speicherung der Nachrichten stehen diese dem System zu Verfügung und müssen nicht erneut angefragt werden, wodurch Ressourcen gespart werden, welches in **Abschnitt 3.2.2** erfordert wird. Des Weiteren werden Nachrichten für Vorgänge der Nachrichtenbeschaffung- und Aktualisierung benötigt (siehe **Abschnitt 4.7**).

Die Speicherung von Daten erfüllt außerdem die Anforderung aus **Abschnitt 3.2.1** für die Speicherung von Nachrichten und Suchen. Neben den Nachrichten werden auch Suchen, spezielle Datensätze zum Nachrichtenerhalt und Login-Daten für das jeweilige soziale Medium gespeichert.

Für die Wahl der Art der Datenbank wird folgendes Beispiel zwei Nachrichten aus Twitter betrachtet:

Nachricht A	Nachricht B
<pre>{   "_id": "705048938659729408",   "created_time": ISODate("2016-03-02T15:15:40Z"),   "updated_time": ISODate("2016-03-02T15:15:40Z"),   "from": {     "_id": "1533348044",     "name": "Luenenet",     "follower": 67   },   "message": "Tsunami-Warnung für Padang auf Sumatra aufgehoben: Jakarta (dpa) - Nach dem starken. https://t.co/1wwjw7yxv4 https://t.co/ECmuAlnYEh",   "media": [     {       "type": "photo",       "url": "http://pbs.twimg.com/media/CcjWRivUUAwCEf.jpg"     }   ],   "url": [     "http://dlvr.it/KgCScv"   ],   "retweet_count": 0,   "favourite_count": 0,   "source": 2,   "searchId": ObjectId("56978ec44eb6da10d8adc040") }</pre>	<pre>{   "_id": "705076646869467136",   "created_time": ISODate("2016-03-02T17:05:46Z"),   "updated_time": ISODate("2016-03-02T17:05:46Z"),   "from": {     "_id": "701621977",     "name": "FlotteNews",     "follower": 604   },   "message": "#TELEKOM Panik nach starkem Erdbeben in Indonesien: Jakarta (dpa) - Ein starkes Erdbeben ... https://t.co/kcj01yEI9S #tonline #news #free",   "hashtag": [     "telekom",     "tonline",     "news",     "free"   ],   "url": [     "http://bit.ly/1RommCK"   ],   "retweet_count": 0,   "favourite_count": 0,   "source": 2,   "searchId": ObjectId("56978ec44eb6da10d8adc040") }</pre>

Tabelle 1 Vergleich zweier Nachrichten aus Twitter

Wie zu erkennen ist, fehlt in der rechten Nachricht das Feld „media“, die ein Array an Media-Objekten enthält, wie es in der linken Nachricht der Fall ist. Der linken Nachricht fehlt das Feld „Hashtag“, mit einem Array von Hashtags, die wiederum die rechte Nachricht besitzt.

Durch die Möglichkeit des Fehlens von Feldern erweist sich eine SQL-Datenbank nicht als effiziente Möglichkeit der Datenspeicherung, da durch die fehlenden Felder null-Werte in der Datenbank entstehen, die nicht weiter genutzt werden.

Nachrichten aus verschiedenen sozialen Medien weichen vom Schema ab. Auch kann sich das Schema einer Nachricht innerhalb eines sozialen Mediums weiterentwickeln. Um den Aufwand der Anpassung und Änderung von Tabellen zu vermeiden, wie es in einer SQL-Datenbank üblich ist, wird auf eine dokumentenorientierten Datenbank ausgewichen. Eine dokumentenorientierte Datenbank zeichnet sich durch die Möglichkeit der Schemafreiheit der Daten aus. Durch die Schemafreiheit der dokumentenorientierte Datenbank ist es nicht nötig für jeden Datentyp eine Tabelle anzulegen, wie es in einer SQL-Datenbank der Fall ist.

Für den weiteren Verlauf der Bachelorarbeit wurde sich für die dokumentenorientierte Datenbank MongoDB<sup>2</sup> entschieden, aufgrund der in **Abschnitt 3.2.2** beschriebenen Anforderung des Einsatzes von erprobter Software.

Alternativ können auch andere dokumentenorientierte Datenbanken benutzt werden.

### 4.1.3 Hintergrundprozesse

Eine grundlegende Funktion des Systems ist die Nachrichtengewinnung aus den sozialen Medien. Auf diese Funktion hat der Nutzer nur indirekt Einfluss. Der Einfluss beschränkt sich auf das Erstellen der Suchen. Suchen können von den sozialen Komponenten genutzt werden, um Datenanfragen an das jeweilige soziale Medium anzupassen.

Jede Komponente der sozialen Medien stellt die erforderlichen Funktionen zum Herunterladen von Nachrichten zur Verfügung. Diese Funktionen werden durch einen im Hintergrund laufenden Prozess periodisch ausgeführt. Der Prozess nutzt dabei nur die vorgeschriebenen Funktionen der einzelnen Komponenten. Durch die periodische Ausführung der angebotenen Funktionen sorgt der Prozess für einen aktuellen Stand der Nachrichten.

Nachrichten, die heruntergeladen sind, werden daraufhin in der Datenbank gespeichert und stehen dem Rest des Systems zur Verfügung.

---

<sup>2</sup> <https://www.mongodb.com/>

## 4.2 Fachliche Architektur

Der nun folgende Abschnitt soll veranschaulichen in welchem Rahmen das System zum Einsatz kommt. Das fachliche Datenmodell in Abbildung 6 soll einen Einblick zur Verwendung des Systems geben.

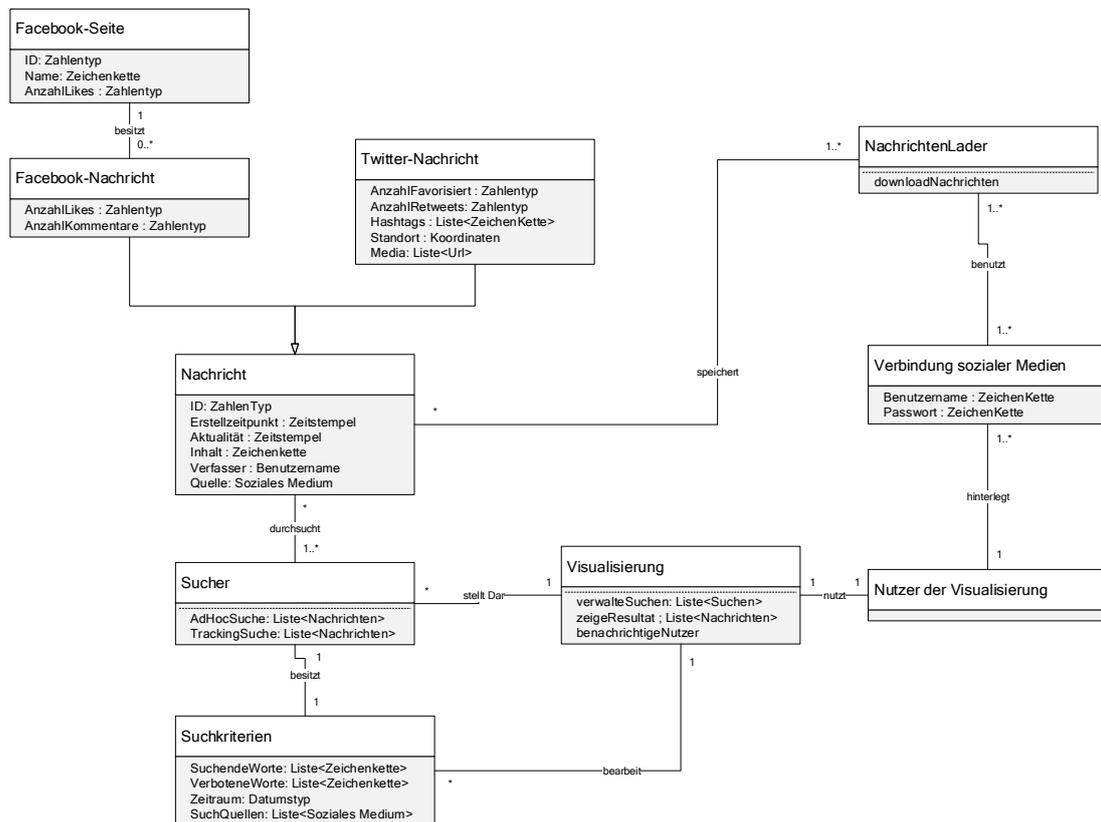


Abbildung 6 Fachliches Datenmodell des Systems

Das fachliche Datenmodell in Abbildung 6 zeigt die wesentlichen Bestandteile des Systems. Um das System nutzen zu können muss der Nutzer seine Daten zu den sozialen Medien hinterlegen. Aus den hinterlegten Daten werden Zugriffstoken für die sozialen Medien generiert. Mit diesen Zugriffstoken ist es möglich die APIs der jeweiligen sozialen Plattformen anzusprechen.

Der NachrichtenLader verwendet, die vom Nutzer hinterlegten Daten, um eine Verbindung mit den sozialen Medien zu etablieren. Diese Verbindung wird genutzt um Nachrichten aus den sozialen Medien zu erhalten und zu speichern.

Die Nachrichten aus Facebook werden über die Facebook-Seiten heruntergeladen, da die von Facebook angebotene API, das Suchen auf dem sozialen Medium nicht ermöglicht. Für die Nachrichten aus Twitter wird keine übergeordnete Quelle benötigt. Da Twitter eine

Suchfunktion mittels der Search API ermöglicht, wird diese auch verwendet um Twitter-Nachrichten für die Suchen zu erhalten.

Der Sucher durchsucht die vom NachrichtenLader bereitgestellten Nachrichten. Um für den Nutzer relevante Nachrichten zu erhalten, verwendet der Sucher Suchkriterien, die vom Nutzer über die Visualisierung eingegeben worden sind. Die in **Abschnitt 3.1.** beschriebene Ad-Hoc- und Tracking-Suche werden vom Sucher bereitgestellt.

Die Visualisierung wird für das Verwalten der Suchen verwendet. Sie soll dem Nutzer eine Übersicht der aktuellen Suchen geben. Über die Visualisierung soll der Nutzer das System benutzen, indem er neue Suchen erstellt, ändert und löscht. Zur Erstellung und Änderung zählt das Bearbeiten von Suchkriterien. Dies hat direkte Auswirkungen auf den Sucher und die daraus resultierende Suchergebnismenge.

Die Benachrichtigung, dass eine Tracking-Suche neue Nachrichten gefunden hat, erfolgt auch durch die Visualisierung.

## 4.3 Außenansicht der Komponenten

Im folgenden Abschnitt wird über die Schnittstellen der einzelnen Komponenten nach außen geschrieben und ihre Bedeutung erläutert.

### 4.3.1 SocialNews-Komponente

Die Komponente Social News dient als Schnittstelle der einzelnen sozialen Komponenten. Sie besitzt mehrere Aufgaben. Eine Aufgabe ist die Bereitstellung eines vereinheitlichten Grundschemas der Nachrichten. Dieses Grundschema bildet die Basis, um die Funktionen der Suchen in den einzelnen sozialen Medien zu realisieren. Des Weiteren bietet sie Schnittstellen zur Textsuche auf den sozialen Medien, sowie eine Schnittstellen zu den hinterlegten Nachrichten in der Datenbank.

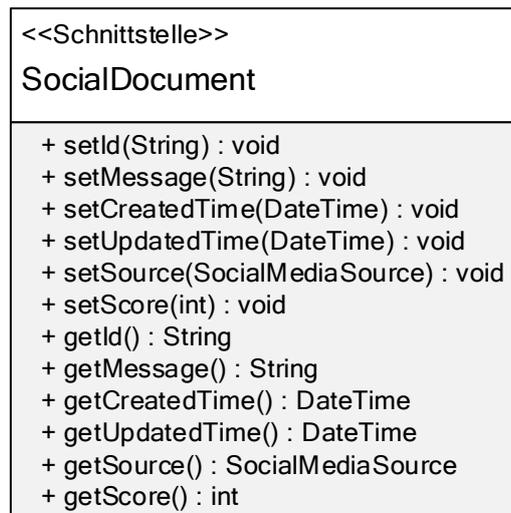


Abbildung 7 Schnittstelle `SocialDocument` aus der `SocialNews`-Komponente

Die Schnittstelle `SocialDocument`, in Abbildung 7, gibt die Grundstruktur einer Nachricht aus den sozialen Medien wieder. Da Nachrichten aus unterschiedlichen sozialen Medien auch unterschiedliche Strukturen vorweisen, ermöglicht diese Schnittstelle das Vergleichen der Nachrichten aus unterschiedlichen Quellen.

`setId()` und `getId()` setzen und erhalten die einmalige ID einer Nachricht.

`setMessage()` setzt den Text der Nachricht und der wiederum für die Suchen der Nachrichten verwendet wird. Mit `getMessage()` kann dieser erhalten werden. `setCreatedTime()` setzt den Zeitpunkt der Erstellung der Nachricht und `getCreatedTime()` gibt ihn wieder. `setUpdatedTime()` setzt den Zeitpunkt der letzten Änderung der Nachricht und `getUpdatedTime()` gibt ihn wieder. Mit der Methode `setSource()` wird das soziale Medium, aus dem die Nachricht stammt, der Nachricht eingefügt. Mit `getSource()` wird dieser Wert ausgelesen. Die Methode `setScore()` setzt den Wert einer Nachricht zum Vergleich mit anderen Nachrichte. Mittels der `getScore()` kann dieser Wert gelesen und zum Vergleichen benutzt werden.

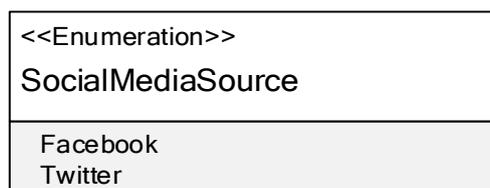


Abbildung 8 Enumeration `SocialMediaSource` aus der `SocialNews`-Komponente

Die Enumeration `SocialMediaSource`, aus Abbildung 8, beinhaltet alle vom System implementierbaren Systeme. Durch sie kann eine Nachricht einem sozialen Medium zugeordnet und die entsprechende Komponente aufgerufen werden. Die Nummerierung der Enumeration erfolgt bitweise um das Kombinieren von Enumerationen zu ermöglichen, für die

Suchen auf mehreren sozialen Plattformen. Die Enumerationen sollen um zukünftige soziale Medien erweitert werden.

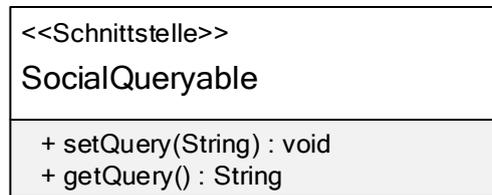


Abbildung 9 Schnittstelle `SocialQueryable` aus der `SocialNews`-Komponente

Die Schnittstelle `SocialQueryable`, aus Abbildung 9, besitzt die Methode `getQuery()` und `setQuery()`. Durch die Methode `setQuery()` ist es den Komponenten gestattet eine Suchquery zu bilden. Mit der `getQuery()` wird sie ausgelesen und benutzt damit die sozialen Medien diese verarbeiten. Dadurch können schon bei der Datenanfrage die Nachrichten gefiltert werden.

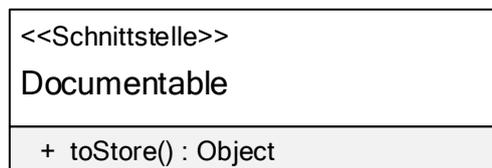


Abbildung 10 Schnittstelle `Documentable` aus der `SocialNews`-Komponente

Die Abbildung 10 stellt die Schnittstelle `Documentable` dar. Diese Schnittstelle ermöglicht das Speichern der Objekte in die Datenbank. Die Methode `toStore()` gibt das `Object` wieder, welches zur Speicherung in die Datenbank verwendet wird.

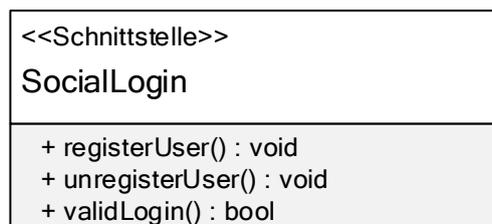


Abbildung 11 Schnittstelle `SocialLogin` aus der `SocialNews`-Komponente

`SocialLogin` ist eine weitere Schnittstelle der `SocialNews`-Komponente und wird in der Abbildung 11 dargestellt. Mit den Methoden `registerUser()` und `unregisterUser()` ist die An- bzw. Abmeldung des Systems mit einem sozialen Medium möglich. Die `validLogin()` Methode überprüft ob die Verbindung mit dem sozialen Medium noch besteht.

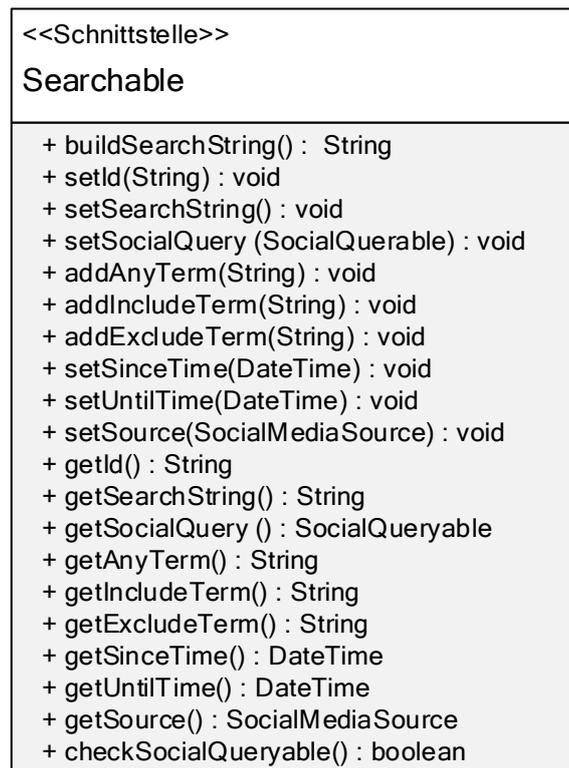
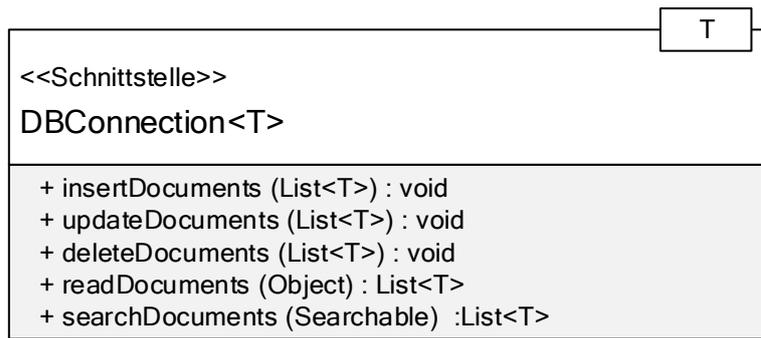


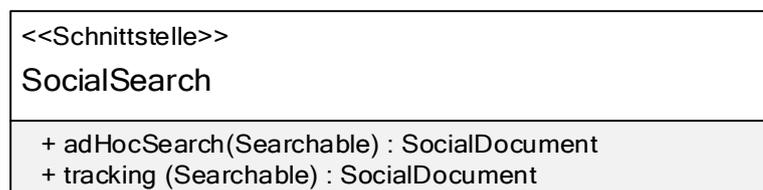
Abbildung 12 Schnittstelle `Searchable` aus der `SocialNews`-Komponente

In Abbildung 12 wird die Schnittstelle `Searchable` dargestellt. Die Schnittstelle `Searchable` ermöglicht das Suchen der Nachrichten in der Datenbank. Die Methode `getSearchString()` gibt einen String wieder, der für die Textsuche in der Datenbank benutzt werden soll. Durch die Methode `buildSearchString()` wird dieser durch die eingegebenen Suchkriterien gebildet. Die Suchkriterien werden durch die Methoden `addAnyTerm()`, `addIncludeTerm()`, `addExcludeTerm()`, `setSinceTime()`, `setUntilTime()` und `setSource()` eingefügt. Die Methoden `addAnyTerm()`, `addIncludeTerm()` und `addExcludeTerm()` fügen die Worte ein, die bei einer Suche beachtet werden sollen, in **Abschnitt 4.7.4** wird dazu näheres erläutert. Mit den Methoden `setSinceTime()` und `setUntilTime()` werden die Nachrichten auf ihren Erstellungszeitpunkt eingeschränkt. Dabei muss eine Nachricht nach dem `setSinceTime()` und vor dem `setUntilTime()` Zeitstempel erstellt worden sein. Die `setSource()` Methode gibt die sozialen Medien an die durchsucht werden sollen. Die gleichnamigen Get-Methoden liefern die in der Suche durch die gleichnamigen Set-Methoden eingefügten Werte zurück. Mit der Methode `checkSocialQueryable()` wird Auskunft darüber geben, ob die soziale Plattform das Empfangen von Nachrichten mittels einer Suchquery ermöglicht.

Abbildung 13 Schnittstelle `DBConnection` aus der `SocialNews`-Komponente

Die generische Schnittstelle `DBConnection`, aus Abbildung 13, dient als Schnittstelle zur Datenbank. Der generische Typ `T` ist dabei eine Unterklasse von `SocialDocument`. Die Methoden `insertDocuments()`, `updateDocuments()` und `deleteDocuments()` besitzen als Parameter `List<T>`. Die Übergabe einer Liste wurde gewählt, um eine Performance-Steigerung der Methoden zu erreichen. Durch die Möglichkeit der Bulk-Operation der dahinterliegenden Datenbank möglich sind. Die Methode `readDocuments()` ermöglicht das Auslesen der Nachrichten der Datenbank. Als Parameter wird ein `Object` erwartet, welches die Auswahl der zu lesenden Nachrichten beschränkt. Dieses `Object` kann, je nach eingesetzter Datenbank, durch ein datenbankspezifisches `Object` ausgetauscht werden.

Die Methode `searchDocuments()` sucht in der Datenbank nach Nachrichten, die durch `Searchable` beschrieben wurden.

Abbildung 14 Schnittstelle `SocialSearch` aus der `SocialNews`-Komponente

Die Schnittstelle `SocialSearch` aus der `SocialNews`-Komponente, wird in der Abbildung 14 dargestellt. Die Schnittstelle stellt die Methoden für das Suchen nach Nachrichten aus den sozialen Medien bereit. Sowohl die Methode `adHocSearch()`, als auch die Methode `tracking()` erwarten ein Objekt, welches die Schnittstelle `Searchable` implementiert, um daraufhin eine Liste an `SocialDocument` zurückzuliefern. Die beiden Methoden bieten die zwei gleichnamigen Suchen an, die in den Anwendungsfällen im **Abschnitt 3.1** beschrieben wurden.

### 4.3.2 Facebook-Komponente

Die Facebook-Komponente ist verantwortlich für die Nachrichten aus der sozialen Plattform Facebook. Die Komponente realisiert die Suchen, die Struktur der Nachrichten, sowie das Speichern der Facebook-Seiten und Nachrichten aus Facebook.

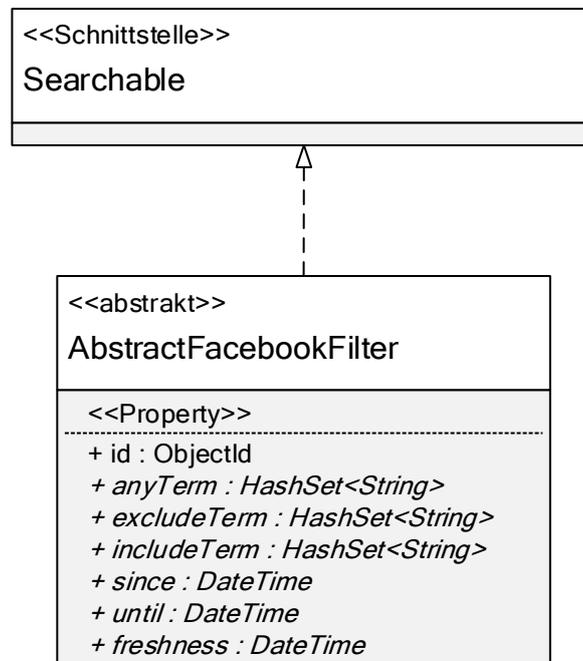
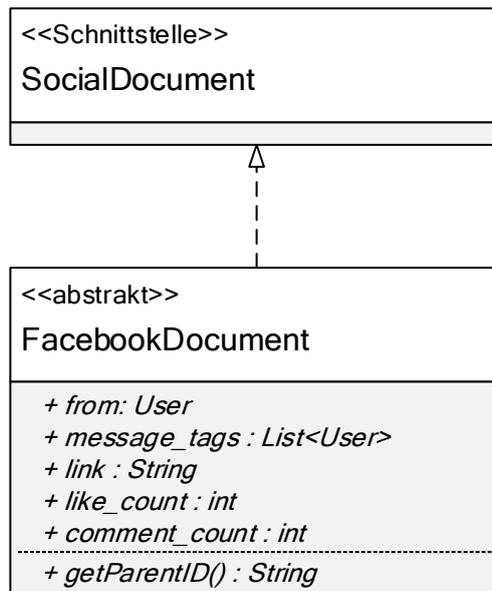


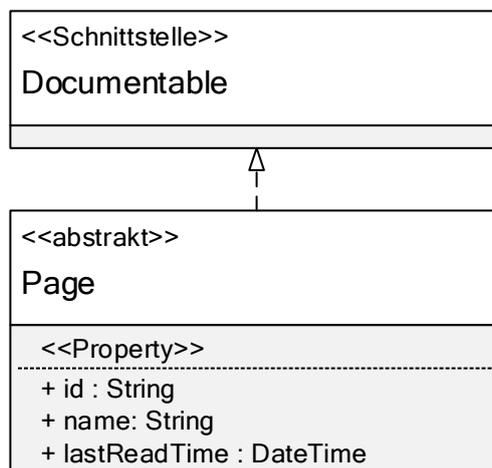
Abbildung 15 Abstrakte Klasse `AbstractFacebookFilter` aus der Facebook-Komponente

Die abstrakte Klasse `AbstractFacebookFilter`, aus Abbildung 15, bietet eine Auswahl an Suchkriterien an, mit der die Suche auf den Nachrichten aus Facebook manipuliert werden kann. Sie implementiert dabei die Schnittstelle `Searchable`.

Mit dem `AbstractFacebookFilter` ist es möglich, nach Facebook-Nachrichten in der dahinterliegenden Datenbank zu suchen. Mit `anyTerm` ist es möglich, ein Set an Worten zu übergeben, die in der Nachricht vorhanden sein sollen. Dabei muss jedoch nur mindestens eins vorkommen. Das Set `excludeTerm` enthält die Worte, die nicht in der Nachricht vorkommen dürfen. `includeTerm` beschreibt ein Set an Worten, die vorkommen müssen. `since` und `until` beschränken den Erstellungszeitraum der Nachricht und `freshness` gibt an wie aktuell die Nachricht sein soll.

Abbildung 16 Abstrakte Klasse **FacebookDocument** aus der Facebook-Komponente

Die abstrakte Klasse `FacebookDocument`, in Abbildung 16, beschreibt die Grundstruktur der Nachrichten in Facebook. Die Klasse erweitert die Schnittstelle `SocialDocument` um Facebook spezifische Inhalte. Die Property `form` gibt Auskunft über den Ersteller der Nachricht. `Message_tags` gibt die in der Nachricht erwähnten anderen `User` an. Mit `link` gibt es die Möglichkeit, den in der Nachricht enthaltenen Link auszulesen. Der `like_count` und der `comment_count` geben Auskunft, wie oft eine Nachricht geliked bzw. kommentiert wurde. Die Methode `getParentID()` gibt die ID der Nachricht oder Seite, aus der das `FacebookDocument` stammt, wieder.

Abbildung 17 Abstrakte Klasse **Page** aus der Facebook-Komponente

Die abstrakte Klasse `Page`, aus Abbildung 17, implementiert das Interface `Documentable` aus der `SocialNews`-Komponente. `Page` repräsentiert die Facebook-Seiten. Die ID von `Page`

ist die einzigartige ID aus Facebook. Die Page enthält außerdem eine Property name die den Namen der Facebook-Seite zurückliefert. Ein Zeitstempel der zuletzt gelesenen Nachricht aus der Facebook-Seite, wird durch lastReadTime realisiert.

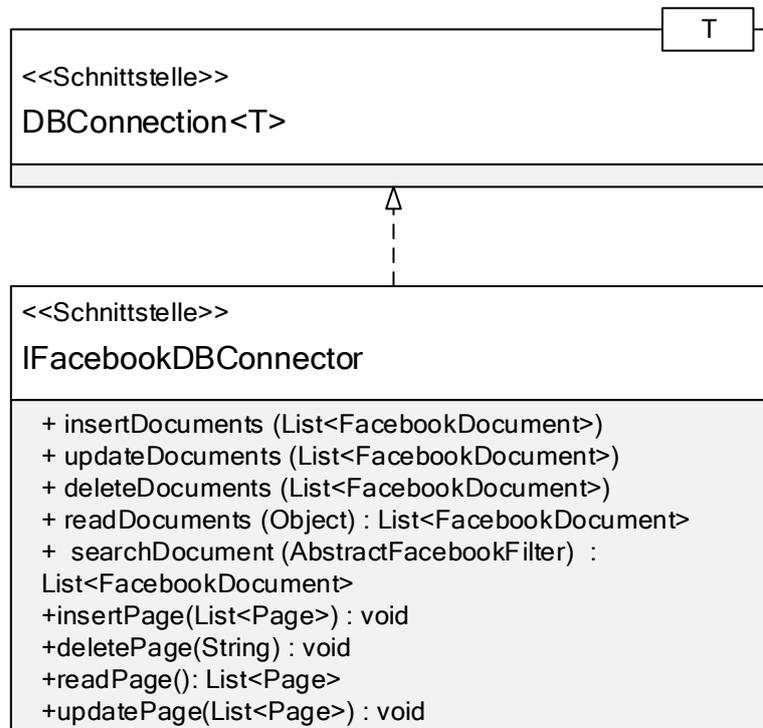


Abbildung 18 Schnittstelle `IFacebookDBConnector` aus der Facebook-Komponente

In Abbildung 18 ist die Schnittstelle `IFacebookDBConnector` zu sehen. Sie implementiert die Schnittstelle `DBConnection` aus der `SocialNews`-Komponente.

Durch das Implementieren der Schnittstelle kann `IFacebookDBConnector` die Nachrichten aus Facebook in der Datenbank verwalten.

Dabei werden für die Methoden zur Nachrichtenverwaltung innerhalb der Datenbank (`insert`-, `update`-, `delete`- und `readDocuments()`) die abstrakte Klasse `FacebookDocument` verwendet. Für die Suche nach Nachrichten, mittels der Methode `searchDocument()`, wird die abstrakte Klasse `AbstractFacebookFilter` verwendet. Außerdem erweitert sie die Schnittstelle `DBConnection` um Methoden zur Verwaltung von Facebook-Seiten, die für die Nachrichtenbeschaffung benötigt werden. Mit der Methode `insertPage()` können Facebook-Seiten in die Datenbank eingefügt werden und mit `readPages()` können alle Facebook-Seiten ausgelesen werden. Die Methode `deletePage()` können Facebook-Seiten gelöscht und mit der Methode `updatePage()` können Facebook-Seiten aktualisiert werden.

### 4.3.3 Twitter-Komponente

Für die Nachrichten aus dem sozialen Medium Twitter ist die Twitter-Komponente verantwortlich. Die Komponente realisiert die Suchen, die Struktur der Nachrichten, sowie das Speichern der Nachrichten aus Twitter.

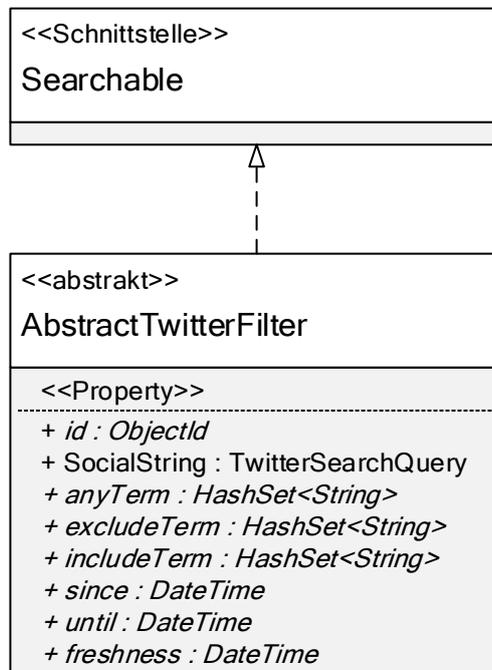
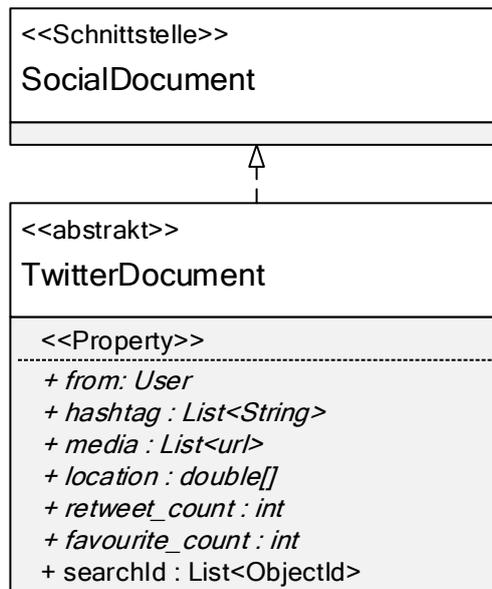


Abbildung 19 `AbstractTwitterFilter` aus der Twitter-Komponente

Durch die abstrakte Klasse `AbstractTwitterFilter`, aus Abbildung 19, ist es möglich, die Suche nach Twitter Nachrichten zu manipulieren. Dabei erfüllen die Properties `anyTerm`, `excludeTerm`, `includeTerm`, `since`, `until`, `freshness` den gleichen Zweck wie in der Facebook-Komponente. Durch die Verwendung von `TwitterSearchQuery`, durch die Property `socialString`, ist es möglich eine Query direkt an Twitter zu senden und so bereits beim Herunterladen nur relevante Nachrichten zu erhalten.

Abbildung 20 `TwitterDocument` aus der Twitter-Komponente

Die abstrakte Klasse `TwitterDocument`, in Abbildung 20, bildet die Grundstruktur einer Nachricht aus Twitter. Sie erweitert die Schnittstelle `SocialDocument` um weitere Elemente. Property `from` ist der User, der die Nachricht erstellt hat. Eine Liste der in der Nachricht enthaltenen Hashtags wird durch `hashtag` angeboten. Media enthält alle Medien der Nachricht. Die Koordinaten des Ortes, an dem die Nachricht erstellt wurde, werden mit `location` erfasst.

Eine Liste an vorhanden links in der Nachricht wird durch das Feld `media` erfasst. Auskunft über die Anzahl der Retweets und Favoriten einer Nachricht werden durch `retweet_count` bzw. `favourite_count` gegeben. Die Liste `searchId` enthält die IDs der Suchen, durch die die Nachricht empfangen wurde.

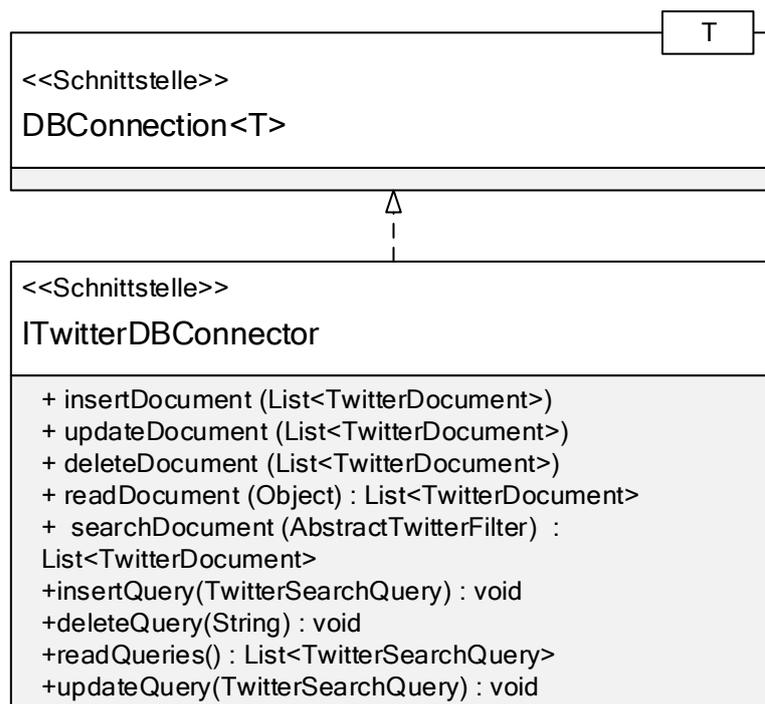


Abbildung 21 Schnittstelle `ITwitterDBConnector` aus der Twitter-Komponente

In Abbildung 21 ist die Schnittstelle `ITwitterDBConnector` zu sehen. Sie implementiert die Schnittstelle `DBConnection` aus der `SocialNews`-Komponente. Wie es die `DBConnection` Schnittstelle vorsieht, ist die `IFacebookDBConnector` Schnittstelle dadurch in der Lage Nachrichten aus Twitter in der Datenbank zu verwalten. Die Methoden zur Nachrichtenverwaltung (`insert-`, `update-`, `delete-` und `readDocuments()`) verwenden dabei die abstrakte Klasse `TwitterDocument`. Für die Methode `searchDocument()`, die zur Suche der Nachrichten innerhalb der Datenbank verwendet wird, wird die abstrakte Klasse `AbstractTwitterFilter` erwartet. Die Schnittstelle `ITwitterDBConnector` erweitert die `DBConnection` um weitere Methoden zur Verwaltung von Queries. Die Queries werden für die Nachrichtenbeschaffung über Twitter benötigt. Für die Queries wird dabei die abstrakte Klasse `TwitterSearchQuery` verwendet. Mit der Methode `readQueries()` können alle gespeicherten Queries aus der Datenbank gelesen werden. Die Methode `insertQuery()` ermöglicht das Speichern der Query in die Datenbank und die Methode `deleteQuery()` das Löschen aus der Datenbank. Die `updateQuery()`-Methode ermöglicht das aktualisieren der Query.

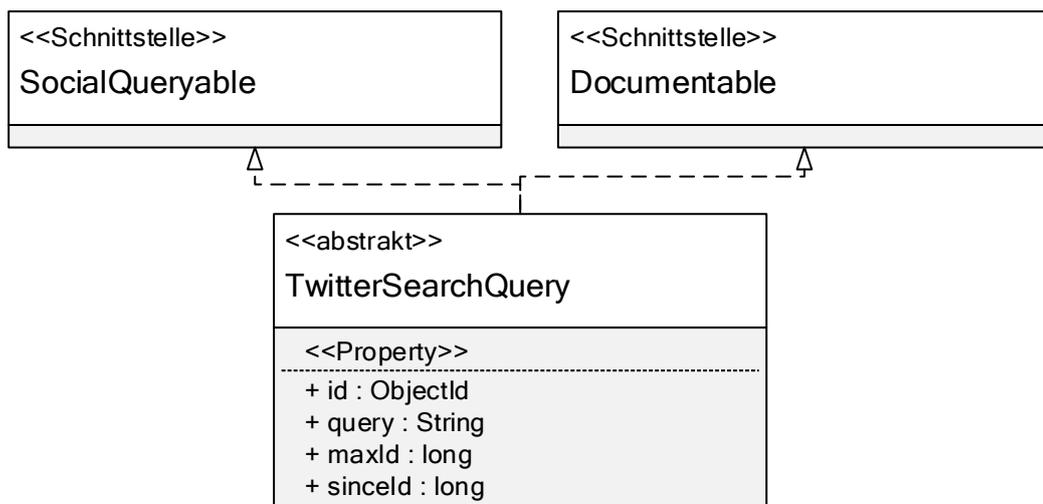


Abbildung 22 Abstrakte Klasse `TwitterSearchQuery` aus der Twitter-Komponente

Die abstrakte Klasse `TwitterSearchQuery`, in der Abbildung 22 abgebildet, repräsentiert eine Query die zur Nachrichtenbeschaffung verwendet wird. Dabei implementiert sie die Schnittstelle `Documentable`, aus der `SocialNews`-Komponente, um das Speichern der Query in einer Datenbank zu ermöglichen. Daneben implementiert sie die Schnittstelle `SocialQueryable`, ebenfalls aus der `SocialNews`-Komponente, um das Verwenden der Query auf Twitter, zur Nachrichtenbeschaffung, zu ermöglichen. Die Property `id` ist die einzigartige ID der Query, die `query` enthält die Query für die Twitter Search API. `MaxId` und `SinceId` werden benötigt um effizient Nachrichten herunterzuladen, näheres wird in **Abschnitt 4.7.3** erläutert.

## 4.4 Innenansicht der Komponenten

Der nun folgend Abschnitt behandelt die Innenansicht und den Aufbau der einzelnen Komponenten. Aufgrund der Übersichtlichkeit wurden die Methoden und Attribute in den Abbildungen nicht mitaufgenommen.

### 4.4.1 SocialNews-Komponente

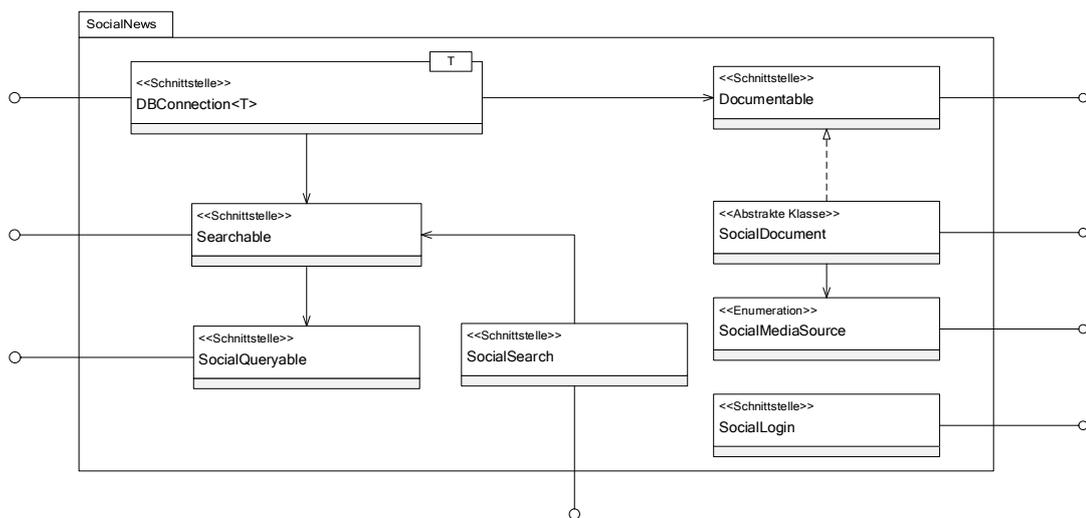


Abbildung 23 Innenansicht der Komponente SocialNews-Komponente

Die einzelnen Schnittstellen und abstrakten Klassen verhalten sich wie in **Abschnitt 4.3.1** beschrieben. In der Abbildung 23 ist zu erkennen, dass die `DBConnection` die Schnittstelle `Searchable` benutzt, um die Suche auf den Datensätzen der Datenbank Nachrichten zu ermöglichen. `SocialDocument` repräsentiert die Grundform einer Nachricht aus einem sozialen Medium und nutzt die Enumeration `SocialMediaSource`, um zu vermerken, aus welcher Quelle eine Nachricht stammt. Außerdem implementiert sie die Schnittstelle `Documentable`, die zur persistenten Speicherung der Nachrichten benötigt wird, für die Schnittstelle zur Datenbank `DBConnection`. Die Schnittstelle `SocialLogin` ermöglicht die Registrierung und Überprüfung der Verbindung zu einem sozialen Medium.



### 4.4.3 Facebook-Komponente

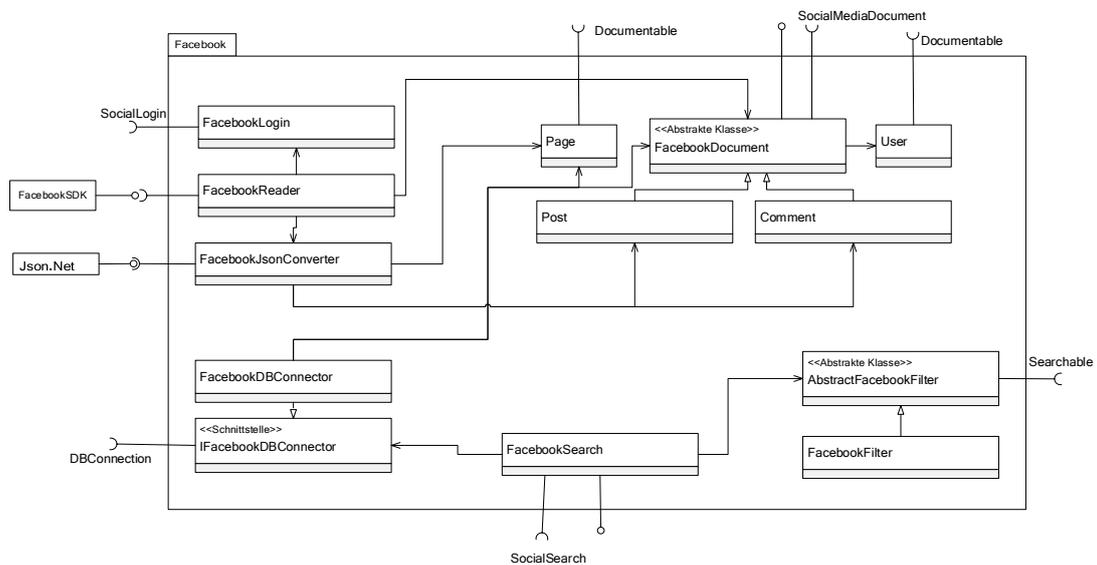


Abbildung 25 Innenansicht der Komponente für Facebook

In Abbildung 25 wird die Innenansicht der Facebook-Komponente dargestellt.

Page ist der Datentyp, der die Facebook-Seite, also die Quelle aus der die Nachrichten kommen, repräsentiert.

Die Klasse User repräsentieren einen User aus Facebook.

User wird vom FacebookDocument benutzt, um anzugeben, von wem die Nachricht stammt. Außerdem können in dem Text einer Nachricht auch andere User genannt werden, diese werden auch von FacebookDocument erfasst.

Post und Comment sind die konkreten Datentypen, die die Schnittstelle FacebookDocument implementieren. Sie stellen die Nachrichten aus Facebook dar.

FacebookLogin ermöglicht die Verbindung zum sozialen Medium Facebook und prüft ob der Nutzer über eine valide Verbindung verfügt. Der FacebookReader verwendet diese Klasse.

Für das Herunterladen der Nachrichten und der Facebook-Seiten ist die Klasse FacebookReader zuständig. Das Anfragen an die Graph API und der Erhalt der Nachrichten erfolgen über die Verwendung des FacebookSDKs.

Anschließend wird der FacebookJsonConverter genutzt, um die erhaltenen Nachrichten aus Facebook, die zu dem Zeitpunkt im JSON-Format sind, in die Klassen Post oder Comment zu konvertieren.

Der FacebookDBConnector speichert die Nachrichten aus Facebook und die Facebook-Seiten in die Datenbank.

Um eine Suche auf den Nachrichten zu starten, wird die Klasse FacebookSearch verwendet.

FacebookSearch benutzt AbstractFacebookFilter für die Suchkriterien und verwendet die Schnittstelle IFacebookDBConnector, um die Suche zu verwenden. FacebookFilter konkretisiert AbstractFacebookFilter und stellt die Suchkriterien für eine Suche auf Facebook dar.

#### 4.4.4 Twitter-Komponente

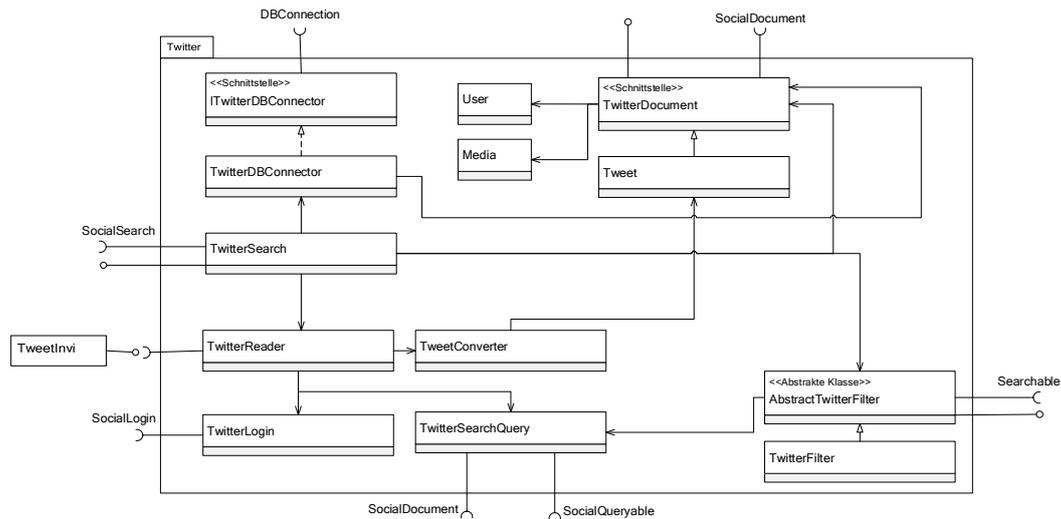


Abbildung 26 Innenansicht der Komponente für Twitter

Abbildung 26 zeigt die Innenansicht der Komponente, die für Twitter verantwortlich ist. `TwitterLogin` ermöglicht die Verbindung zum sozialen Medium Twitter und prüft ob der Nutzer über eine valide Verbindung verfügt. Der `TwitterReader` verwendet diese Klasse. Nachrichten aus Twitter werden von der Klasse `TwitterReader` heruntergeladen. `TwitterReader` verwendet `TweetInvi` und `TwitterSearchQuery` für Anfragen an die Search API und das Erhalten der Nachrichten. Die Nachrichten liegen bereits in bereitgestellten Datentypen von `TweetInvi` vor und werden mit dem `TweetConverter` in die Klasse `Tweet` konvertiert, um nach Bedarf die Tweets zu aktualisieren. Die Klasse `Tweet` konkretisiert die abstrakte Klasse `TwitterDocument`, die die Grundform der Nachrichten aus Twitter repräsentiert. `TwitterDocument` verwendet die Klasse `User`, die Auskunft über den Ersteller der Nachricht gibt. Die Klasse `Media` wird von `TwitterDocument` verwendet. Enthält eine Nachricht ein Foto oder ein Video, wird der Link zum Video von der Klasse `Media` bereitgestellt. `TwitterDBConnector` implementiert die Schnittstelle `ITwitterDBConnector` und somit die Verbindung zur Datenbank. Die Suche wird von der Klasse `TwitterSearch` realisiert. Die Klasse `TwitterSearch` verwendet sowohl den `TwitterDBConnector`, als auch die Klasse `TwitterReader`. Für die Suche nach Nachrichten wird eine Instanz von `AbstractTwitterFilter` benötigt. `TwitterFilter` erbt von der abstrakten Klasse `AbstractTwitterFilter`,

die das Interface `Searchable` implementiert und die `TwitterSearchQuery` verwendet. Der `TwitterFilter` kann somit die Suchkriterien für die Suche nach Nachrichten in der Datenbank und für die Search API bereitstellen.

## 4.5 Fremdsoftware

Im Folgenden werden die im System benutzten externen Softwares genannt und begründet aus welchen Gründen sich für diese Software entschieden wurde.

Aufgrund der in **Abschnitt 3.2.2** genannten nichtfunktionalen Anforderung der Erweiterbarkeit und Wartbarkeit wurde bei der Wahl der zu Einsatzkommendem Fremdsoftware auf die Erfüllung der Entwicklungsumgebung wertgelegt, wodurch die benutzte Fremdsoftware auch im .NET lauffähig ist.

Weiterhin dient der Einsatz von Fremdsoftware zur Reduktion von Programmieraufwand.

Bei der Verwendung der Fremdsoftware wird darauf geachtet, dass diese austauschbar bleiben, um eine Abhängigkeit zu vermeiden.

Die Komponenten zu Facebook und Twitter verwenden folgende Fremdsoftware für den Erhalt und die Verarbeitung von Nachrichten:

### Json.net

Json.net ist eine Library zum Übersetzen von JSON-Daten zu Datenobjekten.

Json.net wird in der Facebook-Komponente verwendet, um die Daten der Nachrichten die von Facebook im JSON Format bereitgestellt werden, in die jeweiligen Datenobjekte der in **Abschnitt 4.3.2** beschriebenen Datenobjekte zu konvertieren.

Alternativ können auch andere Softwarepakete verwendet werden um die JSON-Daten zu übersetzen. Es wurde sich, aufgrund der Entwicklungsumgebung, der hohen Anzahl an Benutzern und der ausführlichen Dokumentation, für JSON.NET entschieden.

### FacebookSDK

FacebookSDK.net ist eine von Facebook, in der Dokumentation der API (siehe [Facebook 2016k]), gelistete Library, mit dessen Hilfe Anfragen an die Graph API von Facebook erstellt werden können. Das Resultat der Anfragen, gibt die Library im JSON-Format wieder und steht dem System zur Weiterverarbeitung zur Verfügung.

Aufgrund des Mangels an weiteren Libraries die den Kriterien der obengenannten nichtfunktionalen Anforderungen entsprechen, wurde sich für diese Library entschieden.

Das Einbinden von anderen Libraries aus einer anderen Entwicklungsumgebung wurde hier, aufgrund der obengenannten nichtfunktionalen Anforderungen, vermieden.

### Tweetinvi

Die Software Tweetinvi ist eine von Twitter, auf der Developer Seite (siehe [Twitter 2016l]), in der Dokumentation der API, gelistet Library, mit dessen Hilfe Anfragen an die APIs von Twitter erstellt werden können. Diese Anfragen werden von der Library verarbeitet. Das Resultat der Anfragen wird in, von der Library bereitgestellten, Datenobjekten zurückgeliefert. Neben den Erhalt von Nachrichten, ermöglicht die Library außerdem die Überwachung der Limitierungen

der Datenanfragen. Dadurch wird Arbeitsaufwand sowohl bei der Realisierung als auch der Konzeptionierung gespart.

Die nichtfunktionale Anforderung der Erweiterbarkeit und Wartbarkeit fordert den Einsatz von erprobter Software. Laut Dokumentation wird diese Library bereits von Forschungsprojekten eingesetzt und erweist sich dadurch als erprobte Software.

## 4.6 Testkonzept

Um die Funktionalitäten und die Qualität des Systems zu gewährleisten, wird das System mittels geeigneter Testverfahren überprüft. Voraussetzung für die Tests ist eine stetige Verbindung zum Internet sowie zur Datenbankanwendung.

Das entwickelte System besteht aus mehreren Komponenten. Dadurch ist es erforderlich jede dieser Komponenten zu testen. Dies ist erforderlich um weitere Fehler bei der Integration dieser Komponenten zu vermeiden, die durch fehlerhafte Funktionen der Komponenten hervortreten. Zunächst werden die Komponenten in ihren einzelnen Bestandteilen mittels Komponententest bzw. Unit Test getestet.

Sie sind dafür da die Komponenten hinsichtlich der ordnungsgemäßen Funktionalitäten der Komponenten zu prüfen.

Für die sozialen Komponenten ist eine stetige Verbindung zum Internet erforderlich, um eine Verbindung zum jeweiligen sozialen Medium zu schaffen.

Im Vordergrund der Tests, der sozialen Komponenten, steht der ordnungsgemäße Erhalt der Nachrichten. Dazu zählt neben des Nachrichteneingangs auch die Überprüfung der Richtigkeit der Datentypen innerhalb der Nachrichten und das Speichern und Laden der Nachrichten.

Nach erfolgreichem Komponententest sollen die Komponenten miteinander getestet werden. Hier werden die Funktionalitäten der Komponenten im Zusammenspiel getestet. Hierbei wird überprüft ob die Komponenten, beim Einsatz andere Komponenten die erwünschten Ergebnisse und Verhaltensweisen erzielen. Dabei sollte zunächst die Integration der sozialen Komponenten getestet werden und anschließend die Suche nach den Nachrichten. Durch die Implementierung der SocialNews-Komponente, durch die sozialen Komponenten, sind bereits Teile der Integrationstests durchgeführt worden, um bei den Komponententests die Funktionalitäten der einzelnen Schnittstellen zu gewährleisten. Das Testen der Fremdsoftware zählt auch zum Integrationstest und muss bereits bei den Komponententest erfolgen, da diese beim Nachrichteneingang zum Einsatz kommen.

Anschließend soll durch einen Abgabetest das entwickelte System in einer realen Umgebung getestet werden. Dieser Test soll weitere Schwachstellen im System ausfindig machen, die durch den Einsatz in einer reale Umgebung auftreten können. Für die Umgebung kann zu diesem Zweck auch eine Testumgebung verwendet werden, die sich sehr nah an der Realität des Einsatzgebietes der Software orientiert. Für den Abgabetest sind mehrere Nutzer erforderlich, die durch die Erfahrungen in ihrem Einsatzgebiet Schwachstellen im System ausfindig machen sollen.

## 4.7 Nachrichten aus sozialen Medien

In diesem Abschnitt werden die Nachrichten der sozialen Medien betrachtet.

Es wird erklärt, wie man die Daten aus den sozialen Medien Facebook und Twitter beschafft, aktualisiert, durchsucht und löscht. Außerdem wird erwähnt, was für Limitierungen bei den Datenanfragen der jeweiligen sozialen Medien vorhanden sind.

### 4.7.1 Nachrichtenbeschaffung

Unterschiedliche Soziale Medien verlangen auch unterschiedliche Methoden um Nachrichten zu erhalten. Einige soziale Medien erlauben es direkt bei der Anfrage die Nachrichten einzugrenzen und somit für die Suche eine bereits bereinigte Menge an Nachrichten bereitzustellen.

Der Vorteil dieser Methode ist, dass dem Nutzer nur relevante Nachrichten aus diesem Medium zur Verfügung stehen. Bei späteren Bereinigungen der Nachrichten entsteht dadurch ein geringerer Aufwand, aufgrund der geringeren Datenmenge.

Ist die Verwendung von Anfragen dieser Art in einem sozialen Medium nicht möglich, müssen Nachrichten bereits im Voraus, also vor der eigentlichen Suche, heruntergeladen werden, um aus den heruntergeladenen Nachrichten eine für die Suche relevante Teilmenge zu erhalten.

#### Facebook

Facebook bietet derzeit keine Möglichkeit an, um bereits bei der Datenanfrage Nachrichten zu filtern. Um Nachrichten zu erhalten, müssen dem Nutzer die Nachrichtenquellen bekannt sein. Dazu muss der Nutzer die Facebook-ID einer Quelle wissen. Mithilfe der Graph API ist es dem Nutzer möglich nach Facebook-Seiten zu suchen und die ID zu erhalten. Die Facebook-Seiten mit ihrer ID werden gespeichert, um eine erneute Suche der ID und Seite zu vermeiden. In der folgenden Abbildung 27 wird die Nachrichtenbeschaffung aus Facebook anhand eines Aktivitätsdiagramms veranschaulicht.

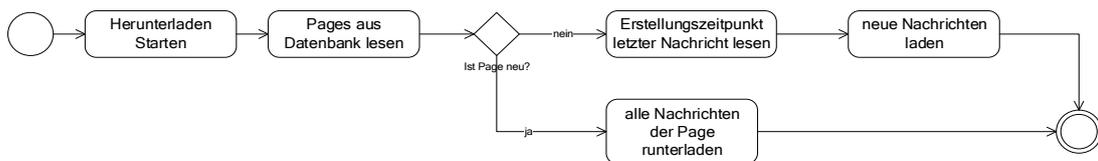


Abbildung 27 Aktivitätsdiagramm zur Nachrichtenbeschaffung aus Facebook (ein Durchlauf)

Wird ein Prozess zur Nachrichtenbeschaffung aus Facebook gestartet, müssen zunächst alle vorhandenen Facebook-Seiten aus der Datenbank gelesen werden. Die Facebook-Seiten sind aus zwei Gründen erforderlich:

1. Die IDs werden benötigt, um Nachrichten aus den Facebook-Seiten zu laden, wie oben bereits beschrieben.

2. Ein Zeitstempel, der nachträglich erstellt wird, wird benötigt, um zu erfahren, ob aus der Facebook-Seite bereits Nachrichten heruntergeladen worden sind.

Ist ein Zeitstempel vorhanden, wird dieser verwendet um mithilfe der Graph API alle Posts aus der Facebook-Seite herunterzuladen, die nach dem Zeitstempel erstellt worden sind. Anschließend werden die aktuellen Kommentare der neuen Post heruntergeladen. Ist bei der Abfrage nach der Facebook-Seite der Zeitstempel noch nicht vorhanden, wurden noch keine Nachrichten erfolgreich heruntergeladen. Daher werden alle derzeit vorhandenen Nachrichten von der Facebook-Seite heruntergeladen.

Ist der Prozess erfolgreich, wird der Zeitstempel der Facebook-Seite mit dem Erstellungszeitpunkt der letzten empfangenen Post aktualisiert. Der Erstellungszeitpunkt der Post wird als Zeitstempel gewählt, da die Graph API bei Angabe eines Zeitstempels nur Posts zurückgibt, die nach dem Zeitstempel erstellt worden sind.

Alle erhaltenen Nachrichten werden in der Datenbank gespeichert. Der Vorgang, aus Abbildung 27, wird periodisch und in regelmäßigen Abständen wiederholt. Die heruntergeladenen Nachrichten stehen, nach jedem abgeschlossenen Vorgang, den Suchen zur Verfügung.

## Twitter

Anders als Facebook bietet Twitter die Möglichkeit, mit Hilfe einer Query, die erhaltenen Nachrichten einzugrenzen. Dies wird durch Twitters Search API angeboten. Dadurch ist es nicht notwendig, Nachrichten bereits im Voraus, also vor der eigentlichen Suche, herunterzuladen. Die Search API setzt dabei nicht auf Vollständigkeit, sondern auf Relevanz und es können Nachrichten bis zu ca. 6-9 Tage aus der Vergangenheit heruntergeladen werden (vgl. [Twitter 2016i]).

Durch die Verwendung der Search API, erfolgt der Vorgang der Nachrichtenbeschaffung aus Twitter erst durch das Erstellen einer Suche. Dadurch wird die Menge an gespeicherten nichtrelevanten Nachrichten reduziert. Durch die angegebenen Suchkriterien wird eine Query für die Search API erstellt. Handelt es sich bei der Suche um eine Tracking Suche, wird die Query in der Datenbank gespeichert.

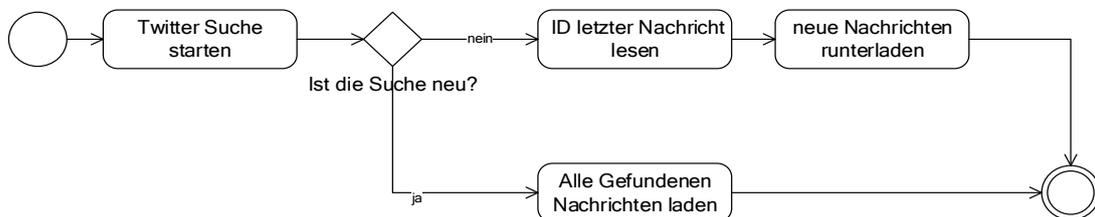


Abbildung 28 Aktivitätsdiagramm zur Nachrichtenbeschaffung aus Twitter durch Tracking Suche (ein Durchlauf)

Das Aktivitätsdiagramm in Abbildung 28 zeigt den Ablauf einer Tracking Suche auf Twitter, um Nachrichten zu erhalten. Zunächst werden die Twitter Suchen aus der Datenbank gelesen. Es wird überprüft, ob es sich bei der Suche um eine neue Suche handelt. Dies wird überprüft durch das Vorhandensein der ID der zuletzt erhaltenen Nachricht durch die gespeicherte Query der Suche.

Ist die ID vorhanden, wird die Search API mit der Kombination aus Query und der ID ausgeführt. Dies hat zur Folge, dass nur neue Nachrichten heruntergeladen werden, die nach der zuletzt erhaltenen Nachricht erstellt worden sind.

Ist die ID nicht vorhanden, wird die Query mittels der Search API ausgeführt und die erhaltenen Nachrichten aus Twitter heruntergeladen. Die Search API wird dabei verwendet, um alle bereits vergangenen Nachrichten zu erhalten, die seit der Ausführung der Search API vorhanden sind.

Nach dem erfolgreichen Herunterladen und dem Speichern der Nachrichten in der Datenbank, wird die verwendete Suche aktualisiert.

Bei der Aktualisierung der Suche wird die ID der aktuellsten eingegangenen Nachricht einer Suche gespeichert.

Dieser Vorgang wird periodisch und in regelmäßigen Abständen wiederholt.

Handelt es sich bei der Suche nur um eine Ad-Hoc-Suche, wird die Query einmalig ausgeführt um ein aktuelles Bild der Nachrichtenlage zu erhalten. Die erhaltenen Nachrichten werden nicht gespeichert, um beim Umstellen der Ad-Hoc-Suche auf eine Tracking Suche weitere Nachrichten zu erhalten, die vor der zurückgelieferten Ergebnismenge der Ad-Hoc-Suche, entstanden sind.

#### 4.7.2 Nachrichtenaktualisierung

Nachrichten aus soziale Medien können sich im Laufe der Zeit verändern. Darum müssen bereits gespeicherte Nachrichten auf ihre Aktualität überprüft werden.

Prioritäten bei der Nachrichtenaktualisierung haben die Veränderungen beim Text, da diese dafür verantwortlich sind, welche Ergebnismenge ein Nutzer bei einer Suche erhält. Neben den Texten kann es auch Veränderungen von Eigenschaften der Nachrichten geben. So können sich z.B. die Anzahl der Kommentare nach Speicherung einer Nachricht ändern. Diese nicht erfassten Nachrichten müssen erkannt und gegebenenfalls nachträglich heruntergeladen werden.

#### Facebook

Bei der Nachrichtenaktualisierung auf Facebook werden zwei Situationen betrachtet.

Zu einem wird überprüft, ob es zu bereits heruntergeladenen Nachrichten neue, noch nicht heruntergeladene, Kommentare gibt. Als zweites wird der Inhalt der Nachricht überprüft. Da Nachrichten auf Facebook bearbeitet werden können, kann sich auch der Inhalt der Nachricht ändern.

Facebook stellt bei Posts einen Zeitstempel zur Verfügung, der Auskunft über den Zeitpunkt der letzten Aktualisierung gibt. Dieser Zeitstempel wird von Facebook aktualisiert, wenn der Beitrag editiert oder kommentiert wurde (vgl. [Facebook 2016h]). Kommentare besitzen diesen Zeitstempel nicht. Die Graph API bietet keine Funktion diesen Zeitstempel abzufragen.

Um festzustellen, ob ein Post neue Kommentare hat oder bearbeitet wurde, muss der Aktualisierungszeitstempel überprüft werden. Überprüft wird, ob der bereits gespeicherte Zeitstempel des Posts älter als der Zeitstempel vom erneut heruntergeladenen Post ist.

Ist dies der Fall und die Anzahl der Kommentare ist gleichgeblieben, bedeutet dies, dass sich der Inhalt Nachricht verändert hat und dieser aktualisiert werden muss.

Ist zusätzlich auch die Anzahl der Kommentare höher, werden die neuen Kommentare heruntergeladen und gespeichert. Im Anschluss wird die aktuelle Version des Posts in der Datenbank aktualisiert.

## Twitter

Tweets auf Twitter können nicht bearbeitet werden. Dadurch kann der Inhalt einer Nachricht nicht nachträglich verändert werden. Daher ist es für die Suche nicht notwendig, Nachrichten aus Twitter nach Textänderungen zu überprüfen. Jedoch können sich die Anzahl der Retweets und „Gefällt Mir“-Angaben ändern.

Um Tweets zu aktualisieren, werden dessen Retweets verwendet. Retweets entstehen nach der Erstellung des Original Tweets. Zusätzlich enthalten Retweets den originalen Tweet, in der aktuellen Fassung zum Erstellungszeitpunkt des Retweets. Der enthaltene Tweet eines Retweets kann verwendet werden, um den originalen Tweet zu aktualisieren.

Bei der Aktualisierung eines Tweets wird dem Tweet ein Zeitstempel hinzugefügt, um zu erfassen, wann die Aktualisierung stattgefunden hat. Der Erstellungszeitpunkt des Retweets wird als Aktualisierungszeitpunkt verwendet. Ein Tweet wird nur aktualisiert, wenn der Retweet nach der letzten Aktualisierung des originalen Tweets entstand.

Die Aktualisierung der Nachrichten erfolgt in der Nachrichtenbeschaffung, da durch die Verwendung der Search API sowohl Tweets, als auch Retweets, in der Ergebnismenge vorhanden sind.

Alternativ ist es möglich, die gespeicherten Tweets erneut Herunterzuladen und diese zu aktualisieren. Je nach Höhe der Anzahl der zu aktualisierenden Tweets, kann dies zu einem erhöhten Aufwand führen. Als Aktualisierungszeitpunkt wird dann der Zeitpunkt der Ausführung des erneuten Herunterladens verwendet.

### 4.7.3 Limitierung der Datenanfragen

Um die Zahl von Datenanfragen an die angebotenen Schnittstellen zu verringern, setzen soziale Medien häufig Limitierungsverfahren ein, die die Anzahl der Zugriffe innerhalb eines vorgegebenen Zeitintervalls begrenzen.

Dies muss bei der Nachrichtenbeschaffung und -aktualisierung beachtet werden, da diese Limitierung der Datenanfragen einen direkten Einfluss auf die Nachrichtenmenge hat. Vermieden werden sollen Datenanfragen, die keine oder nur wenige Nachrichten zurückliefern. Datenanfragen, die eine höhere Anzahl an Nachrichten zurückliefern, sollten bevorzugt werden. Für alle sozialen Medien gilt, dass bei einer nicht abgeschlossenen Datenanfrage, die bereits erhaltenen Nachrichten gespeichert werden. Im Folgenden werden die Verhaltensweisen der verwendeten sozialen Medien vorgestellt.

## Facebook

Wie im **Abschnitt 2.1.1** geschrieben kann es in seltenen Fällen dazu kommen, dass die Anzahl der Graph API Aufrufe limitiert werden. Aufgrund der in **Abschnitt 3.2.2** nichtfunktionalen Anforderung beschriebenen Zuverlässigkeit des Systems, wird im weiteren Verlauf davon ausgegangen, dass eine Limitierung der API Aufrufe vorhanden ist.

Für jede Art der Abfrage gibt es eine gemeinsame Limitierung auf Facebook bei der Verwendung der Graph API. Auf Facebook wird nicht die Anzahl der Anfragen gezählt, sondern die Anzahl der verwendeten IDs. Daher versucht man bei jeder Anfrage möglichst viele Nachrichten zu erhalten. Anfragen auf multiple IDs haben einen performanten Vorteil gegenüber Anfrage einzelner IDs, da sie schnellere Antwortzeiten haben. Jedoch wird dadurch die Anzahl der Nachrichten pro ID beschränkt, weil Facebook die Größe der erhaltenen Datenmenge beschränkt. Der Vorteil einzelner ID Anfragen liegt bei der Anzahl der zu erhaltenen Nachrichten. Dadurch werden Einzelanfragen in der Nachrichtenbeschaffung und -Aktualisierung verwendet.

Nachrichten aus Facebook werden über die Facebook-Seiten erhalten wie in **Abschnitt 4.7.1** beschrieben. Mit jeder Anfrage auf einer Seite ist es möglich bis zu maximal 100 Post zu erhalten (vgl. [Facebook 2016j]). Dafür muss das Limit der zu erhaltenen Post auf 100 gesetzt werden. Um die Anzahl der zu erhaltenen Kommentare einer Nachricht pro Anfrage zu erhöhen, kann auch hier eine Anzahl an zu erhaltenen Kommentaren angegeben werden. Facebook hat hier keine Grenze, wodurch eine beliebige Anzahl genannt werden kann. Gelegentlich kann Facebook eine Fehlermeldung auswerfen, wodurch die Anzahl reduziert werden muss. Dies muss vom System berücksichtigt werden. Bei einer Fehlermeldung müssen die Anzahl der angeforderten Kommentare verringert oder die Datenanfrage abgebrochen werden.

## Twitter

Twitter besitzt für unterschiedliche API Aufrufe auch unterschiedliche Limitierungen. Um die Verwendung der Search API möglichst effizient zu gestalten wird, die von Twitter empfohlene Timeline Methode benutzt (siehe [Twitter 2016k]). Bei der Timeline Methode werden die IDs der erhaltenen Tweets verwendet, um bei der nächsten Verwendung der Search API bereits heruntergeladene Tweets auszuschließen. So werden die verwendeten Queries für die Search API optimiert.

#### 4.7.4 Suche der Nachrichten

Gesucht wird innerhalb Daten der Nachrichten aus den sozialen Medien.

Eine Suche besteht aus zwei Bestandteilen. Der erste Teil ist die Volltextsuche auf den Inhalt einer Nachricht. Der zweite Teil ist die Übereinstimmung der Kennzahlen der Nachricht mit den Suchkriterien.

Um nach Wörtern oder Begriffen in einer Nachricht zu suchen, muss vorher definiert werden, welche Daten der Nachricht dafür verwendet werden. Anschließend wird über dieses definierte Feld der Daten eine Volltextsuche vollzogen.

Nachrichten aus Facebook werden über dessen Text durchsucht. Die Nachrichten von Twitter werden sowohl über ihren Text, als auch über ihre Hashtags gesucht.

Bei der Volltextsuche kann nach einzelnen oder aber auch nach mehreren Worten gesucht werden. Es gibt folgende Unterteilungen:

- Beliebige Worte: Es können beliebige Worte angegeben von denen eines mindestens einmal in der Nachricht vorkommen soll. (OR-Beziehung)
- Enthaltene Worte: Eine beliebige Anzahl an Worten, von denen jedes angegebene Wort mindestens einmal in der Nachricht vorkommen muss. (AND-Beziehung)
- Auszuschließende Worte: Nachrichten die mindestens eines der angegebenen Worte enthält werden von dem Suchergebnis ausgeschlossen. (Not-Beziehung)

Eine Begrenzung der angegebenen Worte ist ratsam, um Fehler bei der Datenanfrage bei den einzelnen sozialen Medien zu vermeiden, obliegt aber dem Nutzer des Systems.

Dabei kann eine mögliche Reihenfolge der verwendeten Worte, die der oben genannten Liste entsprechen.

Die Begrenzung der angegebenen Worte durch Twitter erfolgt bereits bei 500 Zeichen und muss dementsprechend bei der Anfrage an Twitter berücksichtigt werden.

Die Begrenzung entfällt bei Facebook, da hier keine Queries nach Suchbegriffen ermöglicht werden.

Neben den plattformspezifischen Begrenzungen müssen auch auf die Begrenzungen der eingesetzten Datenbank geachtet werden.

Neben den Begriffen, die gesucht werden sollen, bietet das System die Möglichkeit an, weitere Kennzahlen zum Filtern der Suchergebnisse anzugeben. Die Kennzahlen werden aus den funktionalen Anforderungen „Suchkriterien eingeben“, aus dem **Abschnitt 3.1.**, entnommen. Die Nachrichten müssen die Suchkriterien erfüllen, um als Suchergebnis zurückgeliefert zu werden.

Sowohl für die Ad-Hoc-, als auch für die Tracking-Suche, wird die Suche auf den Nachrichten in der Datenbank ausgeführt. Für die Ad-Hoc-Suche kann eine zusätzliche Datenanfrage an das soziale Medium in Betracht gezogen werden, um ein „spontaneres“ Suchergebnis zu erhalten. Jedoch kann dies zu einer erhöhten Antwortzeit der Suche führen. Aufgrund der in **Abschnitt 3.2.2** beschriebenen nichtfunktionalen Anforderung der Effizienz, wird von der zusätzlichen Datenanfrage abgesehen.

Dadurch ergibt sich, dass die Ergebnisse von Ad-Hoc-Suchen auf den Datenbestand der bisher eingegangenen Nachrichten, zum Startzeitpunkt einer Suche, basieren.

Für die Tracking Suche wird dieser Datenbestand fortlaufend aktualisiert.

#### 4.7.5 Nachrichtenbereinigung

Durch die Nachrichtenbeschaffung und -Aktualisierung wird eine Vielzahl an Nachrichten in der Datenbank gespeichert. Dies führt zu einer großen Datenmenge. Eine große Datenmenge in der Datenbank kann zu Performanceeinbußen beim Lese - und Schreibzugriffen führen, so wie bei der Suche von Nachrichten.

Um dies zu vermeiden, wird der Datenbestand regelmäßig bereinigt. Das Bereinigen der Daten hat die Löschung von nichtrelevanten Nachrichten zur Folge.

Wenn eines dieser Kriterien erfüllt ist, gilt die Nachricht als nicht relevant und sollte in nächster Zeit gelöscht werden:

- Nachrichten aus Facebook werden gelöscht, wenn ...
  - ... sie 0 Gefällt Mir und 0 Kommentare haben eine Woche nach deren Erstellung.
  - ... sie nur aus Markierungen anderer User bestehen.
- Nachrichten aus Twitter werden gelöscht, wenn ...
  - ... sie 0 Retweets und 0 Kommentare haben, eine Woche nach deren Erstellung.
  - ... sie nur aus Hashtags bestehen.

Die oben genannten Zahlen der Kriterien sind nicht vorgeschrieben und dienen lediglich als Beispiel. Durch eine konfigurierbare Datei können die erforderlichen Werte bereitgestellt werden.

Bei der Nachrichtenaktualisierung kann es zusätzlich vorkommen, dass Nachrichten, die bereits in der Datenbank gespeichert wurden, nicht vom sozialen Medium gefunden werden können. Diese Nachrichten sind nicht mehr öffentlich zugänglich oder wurden gelöscht. Ist dies der Fall, sollten diese Nachrichten auch in der Datenbank gelöscht werden. Je nachdem ob dies in dem jeweiligen Ermessen des einzelnen sozialen Mediums ist.

#### 4.7.6 Zusammenfassung der Nachrichten

Die anschließende Tabelle 2 liefert einen abschließenden Überblick über die behandelten Themen aus **Abschnitt 4.7**.

	<b>Facebook</b>	<b>Twitter</b>
<b>Nachrichtenbeschaffung</b>	Speicherung der Facebook-Seiten und Überprüfung auf neuen Inhalt durch Zeitstempel.	Speicherung der Twitter-Queries und Überprüfung auf neuen Inhalt durch ID.
<b>Nachrichtenaktualisierung</b>	Durch das erneute Laden der Nachrichten und der alten Nachrichten auf neue Kommentare. Aktualisierung des Textes und Kennzahlen möglich.	Nachrichten werden durch die Retweets aktualisiert. Aktualisierung des Textes nicht möglich, jedoch der Kennzahlen.
<b>Limitierung der Datenanfragen</b>	Facebook-Seiten mit einer höheren Anzahl an neuen Post werden öfters verwendet.	Queries mit einer höheren Anzahl an neuen Nachrichten werden öfters verwendet.
<b>Suche der Nachrichten</b>	Die Suche erfolgt über den Text.	Die Suche erfolgt über den Text und den Hashtags.
<b>Nachrichtenbereinigung</b>	Durch konfigurierbare Einstellungen.	Durch konfigurierbare Einstellungen.

Tabelle 2 Überblick der Nachrichten aus sozialen Medien

# 5 Realisierung

In diesem Kapitel wird die Realisierung der in **Kapitel 4** beschriebenen Konzepte veranschaulicht. Anfangs wird dargestellt, wie weit die Konzepte realisiert worden sind und welche Abweichungen vorgenommen wurden. Anschließend wird die Arbeit und das prototypische System bewertet. Es wird eine Überprüfung der Erfüllung der aufgestellten Anforderungen aus **Abschnitt 3.2** vorgenommen. Neben der Bewertung des Systems anhand der gestellten Anforderungen, wird die zukünftige Perspektive des Systems bewertet. Zum Abschluss des Kapitels erfolgt eine methodische Abstraktion des Systems.

## 5.1 Realisierungsumfang

Im Rahmen dieser Bachelorarbeit wird neben der schriftlichen Ausarbeitung des Themas, auch eine prototypische Entwicklung des Systems vorgenommen.

Die Grundlage des Systems wurde aus den im **Kapitel 3** beschriebenen Anwendungsfällen entnommen. Durch die beschriebenen Anwendungsfälle entstanden die Anforderungen an das System, die wiederum mitverantwortlich für das **Kapitel 4** sind.

Das **Kapitel 3** ist für die Gestaltung und Ausarbeitung des Systems mit verantwortlich.

Das primäre Ziel der Realisierung ist die Nachrichtenbeschaffung und der Suche auf den Nachrichten.

Bei der Realisierung wurden die vier Komponenten aus **Kapitel 4** realisiert.

Die vier realisierten Komponenten sind die Facebook-, Twitter-, Such- und SocialNews-Komponente.

Die beiden Komponenten Facebook und Twitter realisieren die in **Abschnitt 4.7** beschriebenen Konzepte mit dem Umgang der Nachrichten, aus der jeweiligen sozialen Plattform. Um die sozialen Medien Facebook und Twitter verwenden zu können, war es nötig, auf beiden Seiten ein Entwicklerkonto anzulegen.

Die SocialNews-Komponente dient bei der Realisierung als gemeinsame Schnittstelle, um Nachrichten aus den verschiedenen sozialen Medien zu vergleichen. Sie wird von jeder realisierten Komponente implementiert.

Die in den Anwendungsfällen beschriebenen Suchen wurden durch die Such-Komponente realisiert.

Die Such-Komponente ist wie auf der Konzeption entnommen nach dem Entwurfsmuster der Fassade realisiert worden. Für die Suche auf den Nachrichten der sozialen Medien spricht die Such-Komponente die verantwortlichen Komponenten an.

### 5.1.1 Prototyp des Systems

Der Prototyp wurde in der Sprache C# in der .Net Entwicklungsumgebung realisiert. Dies schließt jede der realisierten Komponente mit ein. Für die Datenspeicherung wurde, wie im **Abschnitt 4.1.2** gefordert, eine dokumentenorientierte Datenbank eingesetzt. Hierfür wurde sich für MongoDB aus den im **Abschnitt 4.1.2** genannten Gründen entschieden. Als IDs für die Suchen wurden von MongoDB generierten ObjectIDs verwendet, um einzigartige IDs zu erhalten und doppelte IDs zu vermeiden. Bei der Datenspeicherung wurden zudem die Nachrichten, aus Facebook und Twitter, getrennt voneinander gespeichert.

Für die Volltextsuche auf den Daten in MongoDB wurde die von MongoDB bereitgestellte Funktion `$text` verwendet (siehe hierzu: [Mongo 2016]). Dabei wurden keinerlei Gewichtungen von Kriterien vorgenommen. Die Methode `buildSearchString()`, aus der Schnittstelle `Searchable`, bildet eine von MongoDB interpretierbare Query, die daraufhin die Nachrichten in der Datenbank durchsucht.

Realisiert wurden die Komponenten Facebook und Twitter, die für die Nachrichten aus den gleichnamigen sozialen Medien verantwortlich sind. Diese, in der Konzeption genannten, sozialen Komponenten verwenden außerdem die in **Abschnitt 4.5** vorgestellte Fremdsoftware. Um eine Verbindung mit den sozialen Medien herzustellen, ist eine Verknüpfung durch die jeweilige soziale Komponente erforderlich. Diese wurde durch vordefinierten Parametereingaben realisiert, um den entstehenden Aufwand durch sich wiederholende Anmeldevorgänge zu vermeiden.

Ein Grund für die Trennung der sozialen Medien in einzelne Komponenten war die in **Abschnitt 4.1** erwähnte lose Kopplung. Da das System verlangt, dass Nachrichten aus unterschiedlichen Medien vergleichbar sind, musste die Schnittstelle `SocialDocument` aus der `SocialNews`-Komponente implementiert werden. Dadurch entstand ein Kompromiss, da die Komponenten zwar unabhängig voneinander, jedoch teilweise abhängig von der Komponente `SocialNews` sind. Aufgrund dessen wurden die Schnittstellen in den Komponenten wiederum von Schnittstellen bzw. abstrakten Klassen innerhalb der realisierenden Komponente implementiert.

Die `SocialNews`-Komponente wurde, wie im **Abschnitt 4.4.1** beschrieben, realisiert und dient dem gleichen beschriebenen Zweck. Wird eine Suche ausgeführt, verwendet die Such-Komponente die bereitgestellten Schnittstellen der sozialen Komponenten, um die Suche auszuführen.

In der Facebook-Komponente wird, sowohl für die Ad-Hoc-, als auch bei der Tracking-Suche, die Suche auf den gespeicherten Nachrichten der Datenbank ausgeführt.

Um Nachrichten für die Suche zu erhalten sind bei der Facebook-Komponente Facebook-Seiten erforderlich, die beobachtet werden müssen. Diese Facebook-Seiten wurden in der Datenbank hinterlegt, um den Eingang von Nachrichten aus Facebook zu gewährleisten.

Auch bei der Twitter-Komponente wird die Tracking-Suche auf den gespeicherten Nachrichten der Datenbank ausgeführt. Dies ist erforderlich um auch Suchkriterien zu berücksichtigen die nicht Worte sind, sondern Kennzahlen. Um Nachrichten für die Tracking-Suche zu erhalten,

wurden die angegebenen Worte der Suchkriterien für die Query der Datenanfrage an Twitter verwendet, um so bereits gefilterte Nachrichten zu erhalten.

Soll eine Ad-Hoc-Suche auf dem sozialen Medium Twitter ausgeführt werden, führt die Twitter-Komponente, neben der Suche der Nachrichten auf der Datenbank, eine zusätzliche Datenanfrage aus, um einen zusätzlichen Blick auf die gegenwärtige Nachrichtenlage zu erhalten. Diese neu erhaltenen Nachrichten werden jedoch nicht gespeichert. Die Datenanfrage bei der Ad-Hoc-Suche ist, wie bei der Tracking-Suche, eine aus den Suchkriterien gebildete Query.

Die Such-Komponente, die als Fassade die sozialen Komponenten verwendet, führt die jeweiligen Suchklassen der Komponenten aus.

Um auch zu einem späteren Zeitpunkt aktuelle Daten zu erhalten, werden die Tracking-Suchen in der Datenbank gespeichert.

Um Nachrichten aus unterschiedlichen sozialen Medien zu sortieren bzw. zu vergleichen, wird die Methode `getScore()` bereitgestellt, die jedes `SocialDocument` implementieren muss. Bei der Facebook-Komponente wurden hier die „Gefällt Mir“-Angaben und Kommentare und auf Twitter die Favoriten und die Anzahl der Retweets addiert.

### 5.1.2 Abweichungen im Prototyp

Die Realisierung des Systems, aus dem **Kapitel 4**, wird auf einer Maschine durchgeführt und nicht wie in **Kapitel 4** vorgesehen, dass die Datenbank als Back-End realisiert wird.

Weiterhin wurden auf die Aspekte der Sicherheit und Datenschutz, in Hinblick auf die Datenspeicherung von sensiblen Daten, verzichtet. Da dieser Aspekt, im jetzigen Stadium des Prototyps, zu keinen zusätzlichen Nutzen dient.

Die in **Abschnitt 4.7** beschriebene Nachrichtenbeschaffung, -aktualisierung und -bereinigung sollten in Hintergrundprozesse automatisiert werden. Im Rahmen der Arbeit wurden diese Vorgänge, aufgrund des dadurch entstehenden geringen Gesamtnutzens für diese Arbeit, nicht automatisiert. Jedoch wurden die Vorgänge in Testfällen abgebildet und überprüft.

Eine Funktion, die den Nutzer mit den sozialen Medien authentifiziert, wurde durch vorgefertigte Parameterangaben realisiert, um wiederholende Anmeldeverfahren zu meiden. Jedoch ist diese Funktion durch die Einbindung der FacebookSDK und der Tweetinvi Fremdsoftware erleichtert, da diese bereits die Funktionalitäten implementiert haben und für zukünftige Implementierungen genutzt werden können.

In der Konzeption vorgesehene konfigurierbare Dateien wurden durch vordefinierte Parametereingaben realisiert.

Eine Limitierung der Datenanfragen wurde in Facebook nicht vorgenommen, da eine Limitierung der Anfragen seitens Facebook, nicht erfolgte.

Die abstrakten Klassen `Page` und `TwitterSearchQuery` wurden als Klassen implementiert.

Die Realisierung einer GUI-Komponente, zur Visualisierung von Suchkriterien und -ergebnissen, lag nicht im Kern der Arbeit, wodurch sie nicht in einer eigenständigen Komponente realisiert.

## 5.2 Test

In Rahmen der Bachelorarbeit wurden mit Unit-Test gearbeitet, um die Funktionalität des entwickelten Prototypens zu testen. Zum Testen wurde hierfür die Software NUnit<sup>3</sup> verwendet, um voneinander unabhängige Test zu kreieren. Außerdem wurden mit Logs gearbeitet, um die einzelnen Vorgänge zu überwachen. Die Software NLog<sup>4</sup> wurde dafür verwendet.

Im Vordergrund der Test standen die Funktionalitäten der Komponenten, die verantwortlich für die sozialen Medien sind. Im Kern standen dabei die Funktionalitäten die zum Erhalten und zum Suchen von Nachrichten beitragen, da dieser Kern der Arbeit sind und weiterführende Test zu einem Mehraufwand führen würden, der im Rahmen der Arbeit zu keinen weiteren Nutzen führen würde. Tests zum Gebrauch des Systems durch mehrere Nutzer erfolgten nicht.

Voraussetzung war für die Tests ist eine stetige Internetverbindung. Auch eine Verbindung zu einer Datenbank ist erforderlich, um Nachrichten sowie Suchen zu speichern bzw. zu laden.

Um Nachrichten aus Facebook zu erhalten war es zunächst notwendig, das Speichern und Laden von Facebook-Seiten zu überprüfen.

Bei den Twitter-Nachrichten ist für den Erhalt der Nachrichten erforderlich, bereits vorher eine Suche bzw. Suchkriterien anzulegen, da aus den Suchkriterien wiederum die Queries entstehen, die für die Nachrichtenerhalt notwendig sind.

Diese anfänglichen Tests sind erforderlich um weitere Test auf deren Grundlage auszuführen zu können.

Die darauffolgenden Tests befassten sich mit der Nachrichtenbeschaffung sowie der Aktualisierung der Nachrichten.

Bei der Facebook-Komponente wurde hier das Herunterladen von Nachrichten aus bereits gespeicherten Facebook-Seiten, sowie aus neu hinzugefügten Facebook-Seiten, getestet. Analog erfolgten die Tests der Verfahren für die Twitter-Komponente, mittels der Queries für die Datenanfragen.

Nach Abschluss der Test erfolgte das Testen der Suchen. Die Suchen wurden dahingehend überprüft, ob die Suchergebnisse den Kriterien einer Suche entsprechen. Die Suchergebnisse stand dem System innerhalb von 1-5 Sekunden zur Verfügung. Im Laufe der Tests wurden mehr als 2 Millionen Nachrichten empfangen.

Während des Testens der Lauffähigkeit des Systems haben sich mehrere Probleme kenntlich gemacht.

Ein Problem das bei den Tests aufgetreten ist, betrifft die Vollständigkeit der erhaltenen Nachrichten. Beim Nachrichtenerhalt wurde überprüft, ob auch tatsächlich Nachrichten empfangen wurden. Jedoch erwies es sich als schwierig zu testen, ob tatsächlich alle Daten empfangen wurden. Zwar kann eine Nachricht Auskunft über die Anzahl an Kommentaren

---

<sup>3</sup> <http://www.nunit.org/>

<sup>4</sup> <http://nlog-project.org/>

geben, jedoch ist nicht sichergestellt, ob diese tatsächlich beim Herunterladen der Nachrichten vorhanden sind. Während des Vorgangs des Prozesses könnten Nachrichten zwischenzeitlich gelöscht bzw. neu erstellt worden sein, wodurch die anfängliche Auskunft der Nachricht falsch wird. Die Wiederholbarkeit der Tests, hinsichtlich der Überprüfung der Vollständigkeit der Daten, ist nicht gegeben.

Ein weiteres Problem stellte die Qualität der Nachrichten dar.

Es ist derzeit nicht möglich die inhaltliche Relevanz der gefundenen Nachrichten einer Suche zu überprüfen. So können in der Ergebnismenge der Suchen Nachrichten sein, die nur aus den Suchbegriffen bestehen und keinerlei weiteren Informationen enthalten.

Auch können Nachrichten in unterschiedlichen Sprachen gefunden werden, wenn diese Worte enthalten, die in den Suchkriterien enthalten sein können. Dadurch können Nachrichten gefunden werden, die im Kontext der Suche nicht wünschenswert sind. Auch die Aktualität der Nachrichten ist nicht immer gegeben. Wie bereits bei der Vollständigkeit der erhaltenen Nachrichten, können Nachrichten, die gefunden werden, zum Zeitpunkt der Suche bereits gelöscht oder geändert worden sein.

Ein weiterer Punkt, der sich beim Testen bemerkbar gemacht hat, ist die Dauer einiger Prozesse. Dies betrifft das Vorgehen beim Herunterladen von Nachrichten einer neu hinzugefügten Facebook-Seite, sowie das Aktualisieren der Nachrichtenbestände, nach einem längeren Zeitraum.

Je länger der zuletzt abgeschlossene Prozess zum Herunterladen zurückliegt, desto länger dauert der Prozess zum Herunterladen von neuen Nachrichten.

Das liegt darin begründet, dass in einem längeren Zeitraum mehr Nachrichten als in einem kleineren Zeitraum entstehen können. Diese müssen wiederum durch mehrere Datenanfragen heruntergeladen werden. Dadurch entstehen weitere Verzögerungen, da das Limit der Datenanfragen verbraucht wird und bis zum nächsten Zeitpunkt der Wiederaufladung gewartet werden muss.

## 5.3 Bewertung

In diesem Kapitel wird die Realisierung der Arbeit bewertet. Zunächst werden die Anforderungen an das System begutachtet und im Anschluss wird der Prototyp bewertet. Zur Bewertung des Prototypens wird zusätzlich eine weitere Perspektive des Systems gegeben und eine methodische Abstraktion vorgenommen.

### 5.3.1 Erfüllung der Anforderungen

In diesem Kapitel werden die aus **Kapitel 3** beschriebenen Anforderungen überprüft. Es wird überprüft, welche Anforderungen erfüllt wurden. Sollten Abweichungen in der Realisierung zu den Anforderungen vorhanden sein, werden diese benannt.

## Funktionale Anforderungen

Im Folgenden wird überprüft, welche Funktionen die Realisierung des Systems erfüllt hat. Die Funktionalen Anforderungen sind aus dem **Abschnitt 3.2.1** entnommen.

- **Verknüpfung zu sozialen Plattformen**  
Diese Anforderung wurde zum Teil erfüllt.  
Das System arbeitet mit den Zugriffstoken eines Nutzers und bietet die Funktionalität sich anzumelden über die Fremdsoftware.  
Aufgrund der prototypischen Entwicklung des Systems wurden die Zugriffstoken mittels der Entwicklungstools der jeweiligen sozialen Medien bezogen und als vordefinierte Objekte benutzt.
- **Eingeben von Suchkriterien**  
Diese Anforderung wurde erfüllt.
- **Mehrere Suchen erstellen**  
Diese Anforderung wurde erfüllt.
- **Laden von Nachrichten aus sozialen Medien**  
Diese Anforderung wurde erfüllt.
- **Speichern von Nachrichten und Suchen**  
Diese Anforderung wurde erfüllt.
- **Benachrichtigung bei Nachrichteneingang**  
Diese Anforderung wurde nicht erfüllt.
- **Nachrichten anzeigen und vergleichen**  
Diese Anforderung wurde teilweise erfüllt. Das Anzeigen ist durch das Fehlen einer Realisierung einer Visualisierung nicht vorhanden. Durch die Schnittstelle `SocialDocument`, aus der `SocialNews`-Komponente, besteht jedoch die Funktionalität und somit die Möglichkeit, Nachrichten zu vergleichen.
- **Aktualisieren der Nachrichten**  
Diese Anforderung wurde erfüllt.
- **Löschen von nichtrelevante Nachrichten**  
Diese Anforderung zum Teil erfüllt. Die Löschung der Nachrichten ist möglich. Jedoch wurde die konfigurierbare Datei, die dem System beim Löschen der Nachrichten dient, mittels Parametereingaben realisiert und nicht als externe Datei.

Ein Großteil der funktionalen Anforderungen an das System wurden erfüllt. Durch den Entfall der Realisierung einer GUI-Komponente konnten Teile der funktionalen Anforderungen nicht

erfüllt werden. Durch den Visualisierungsentwurf aus **Abschnitt 4.1.1** wurden Ansätze zur Realisierung einer solchen GUI-Komponente geliefert. Die Erfüllung der funktionalen Anforderungen, die in Abhängigkeit zu einer Realisierung der GUI-Komponente stehen, wurden konzipiert und müssen durch das Realisieren einer GUI-Komponente umgesetzt werden.

### **Nichtfunktionale Anforderungen**

Die nichtfunktionalen Anforderungen die in **Kapitel 3** beschrieben wurden, beschreiben die Anforderungen an das System, die erfüllt werden sollen, um die Qualität der Software zu gewährleisten.

Die Sicherheit der Zugriffstoken und Daten erfolgt durch die Möglichkeit eines angebotenen Zugriffsschutzes der Datenbank MongoDB. Dieser schützt die Datenbank vor Zugriff von nicht autorisierten Nutzern. Weitere Maßnahmen zur Sicherheit und zum Datenschutz wurden aufgrund der prototypischen Entwicklung des Systems nicht getroffen, da sie zu keinem weiteren Nutzen für die Funktionalitäten des Systems beitragen.

Durch die prototypische Realisierung des Systems lässt sich keine Aussage über die Qualität der Benutzbarkeit treffen. Die Realisierung, der in **Kapitel 4** vorgestellten Konzeption einer Benutzeroberfläche des Systems, fordert dadurch eine nähere Überprüfung.

Die Zuverlässigkeit ist durch den Einsatz der dokumentenorientierten Datenbank gegenüber Änderungen der Datensätze gegeben. Durch das Speichern der Nachrichten ist zu dem Zeitpunkt der letzten Nachricht eines jeden sozialen Mediums bekannt.

Die Ausnutzung der Datenanfragen sieht in der Konzeption zeitlich periodische Ausführungen vor, wodurch ein regelmäßiger Nachrichteneingang ermöglicht wird. Eine Verhaltensweise bei fehlgeschlagener Datenanfrage wurde nicht realisiert. Hier muss darauf geachtet werden, aus welchem Grund der Fehler ausgelöst wurde und dementsprechend reagiert werden.

Teile der Anforderungen der Effizienz werden durch das Speichern der Nachrichten erfüllt. Dem System stehen, durch das Speichern der Nachrichten, bereits Nachrichten für die Suchen zur Verfügung. Dadurch können die Suchen unabhängig von den Hintergrundprozessen zur Nachrichtengewinnung ausgeführt werden. Dadurch wird die Antwortzeit einer Suche verringert, da diese nicht an den Antwortzeiten der sozialen Medien gebunden sind. Die Antwortzeiten für die Suchen waren innerhalb der gewünschten 5 Sekunden.

Durch gespeicherte Nachrichten lassen sich außerdem optimierte Anfragen erstellen, die Ressourcen bei der Anzahl der Datenanfragen sparen. Aus den zuletzt eingegangenen Nachrichten können Zeitstempel extrahiert werden, die dazu genutzt werden, um Anfragen zu erstellen. Nur Nachrichten, die nach diesem Zeitstempel erstellt worden sind, werden durch diese Anfragen erhalten.

Die SocialNews-Komponente ermöglicht grundsätzlich das Erweitern des Systems um weitere soziale Medien. Durch den Einsatz von Fremdsoftware bei der Realisierung des Systems wurde versucht Aufwand zu vermeiden. Hier muss weiterhin überprüft werden, inwieweit eine

tatsächliche Verringerung des Aufwandes besteht. Eine tatsächliche Bewertung der Anforderung Erweiterbarkeit und Wartbarkeit hängt von der Realisierung der weiteren Komponenten ab und muss zukünftig überprüft werden.

In der folgenden Tabelle wird eine Übersicht der Erfüllung der Anforderungen gegeben:

	Konzept	Realisierung
<b>Funktionale Anforderung</b>		
Verknüpfung zu sozialen Plattformen	Ja	Teilweise
Eingeben von Suchkriterien	Ja	Ja
Mehrere Suchen erstellen	Ja	Ja
Laden von Nachrichten aus sozialen Medien	Ja	Ja
Speichern von Nachrichten und Suchen	Ja	Ja
Benachrichtigung bei Nachrichteingang	Teilweise	Nein
Nachrichten anzeigen und vergleichen	Teilweise	Teilweise
Aktualisieren der Nachrichten	Ja	Ja
Löschen der Nachrichten	Ja	Teilweise
<b>Nichtfunktionale Anforderung.</b>		
Erweiterbarkeit und Wartbarkeit	Ja	Teilweise
Effizienz	Ja	Ja
Zuverlässigkeit	Ja	Ja
Benutzbarkeit	Ja	Teilweise
Sicherheit	Ja	Teilweise

Tabelle 3 Übersicht der Anforderungen

### 5.3.2 Bewertung des Prototyps

Die sozialen Komponenten Facebook und Twitter ermöglichen das Herunterladen von Nachrichten aus den gleichnamigen sozialen Medien.

Dabei wurde sowohl beim Konzept der Komponenten, als auch bei der Realisierung Fremdsoftware benutzt. Die Fremdsoftware FacebookSDK und Tweetinvi wurde benutzt, um die API der sozialen Medien anzusprechen und sie ermöglichen das Empfangen der Nachrichten. Eine Limitierung der Datenanfragen findet nur bei der Twitter-Komponente statt. Diese wird durch die Fremdsoftware Tweetinvi verwaltet. Bei der Facebook-Komponente wird keine Überwachung der Datenanfragen durch die FacebookSDK vorgenommen. Um die zukünftige Stabilität und Zuverlässigkeit des Systems zu gewährleisten, müssen Fehlermeldungen, die durch zu viele Datenanfragen auftreten, abgefangen werden.

Des Weiteren ist der Einsatz von Fremdsoftware kritisch zu betrachten. Durch den Einsatz dieser Fremdsoftware wird Entwicklungsaufwand gespart, jedoch gibt es mittel- bis langfristig keine Zusicherung der Lauffähigkeit der Fremdsoftware. Dadurch entsteht eine Abhängigkeit an Dritte.

Darum sollte in Zukunft auf den Einsatz von Fremdsoftware verzichtet und auf eigene Implementationen der sozialen Medien gesetzt werden. Durch den Einsatz von Fremdsoftware dieser Art entstehen Unsicherheiten bei der Zuverlässigkeit der Software. Ein Beispiel ist die Seite mit der Dokumentation der Fremdsoftware FacebookSDK<sup>5</sup>, die zum aktuellen Zeitpunkt (31.05.2016) nicht mehr erreichbar war. Durch das Fehlen der Dokumentation ist die weitere Nutzung des FacebookSDKs kritisch zu betrachten.

Mittels des Entwurfsmusters der Fassade wurde die lose Kopplung aus **Abschnitt 4.1** zwischen den sozialen Komponenten erreicht.

Um die in **Abschnitt 4.1** erwähnte lose Kopplung zu erreichen und die zur Erfüllung der Anforderung der Zuverlässigkeit des Systems aus **Abschnitt 3.2.2** führt, ist jedoch ein Kompromiss entstanden. Es besteht nur eine geringe Kopplung zwischen den einzelnen sozialen Komponenten, jedoch aber eine zwischen den sozialen Komponenten und der SocialNews- Komponente. Änderungen in den Schnittstellen der SocialNews-Komponente haben zur Folge, dass sich auch die sozialen Komponenten ändern müssen. Dieses Problem wurde gelöst, indem diese Schnittstellen wiederum von abstrakten Klassen, innerhalb der sozialen Komponenten, implementiert wurden, die dann gegebenenfalls bei Änderungen eine vorübergehende Implementation anbieten können.

Durch die Komponente der SocialNews ist es möglich, weitere soziale Medien zum System hinzuzufügen. Belegt wird dies, durch die Realisierung der Twitter und Facebook-Komponenten.

Ein Vorteil der Komponenten Facebook und Twitter ist, dass sie auch losgelöst vom System einsetzbar sind und somit weitere Einsatzmöglichkeiten denkbar sind.

Die erforderlichen Suchfunktionen sind durch den Prototypen möglich. Die Ad-Hoc-Suche ist jedoch bei den Nachrichten aus Twitter aktueller, als bei denen aus Facebook. Hauptgrund hierfür ist die angebotene Suchfunktion der Twitter Search API, die aktuelle Nachrichten bietet. Die Tracking-Suche ermöglicht das Überwachen von Nachrichten durch einer Suche. Jedoch wird diese immer auf den gesamten Datenbestand ausgeführt. Durch die Realisierung einer Visualisierungskomponente könnte das System sich die bereits erhaltenen Nachrichten merken. Hierbei ist zu beachten, dass es derzeit möglich ist, Suchen zu ändern und durch die erneute Abfrage des Datenbestandes der Datenbank sich nur relevante Nachrichten im Suchergebnis befinden. Wird sich das Suchergebnis gemerkt und die Suche wird nachträglich geändert, erfordert dies nun eine erneute Suche auf dem gesamten Datenbestand, da die Richtigkeit des gemerkten Suchergebnisses nicht mehr gegeben ist.

Für die Tracking-Suche wird sowohl bei der Facebook-, als auch bei der Twitter-Komponente, eine Suche auf der Datenbank ausgeführt. Durch den regen Austausch innerhalb der sozialen

---

<sup>5</sup> [www.facebooksdk.net](http://www.facebooksdk.net)

Medien kann jedoch nie von einem identischen Stand innerhalb der Datenbank ausgegangen werden.

Der Vergleich der Nachrichten, durch die `getScore()`-Methode aus der Schnittstelle `SocialDocument`, diente nur dem Zweck der Realisierung der Funktionalität, nicht aber der Aussagekraft der Wichtigkeit einer Nachricht.

Durch die Speicherung der Nachrichten in einer Datenbank besteht die Möglichkeit diese Nachrichten weiteren Systemen zu Verfügung zu stellen. Hier muss darauf geachtet werden nicht gegen die Richtlinien einzelner sozialen Medien zu verstoßen.

Der Prototyp muss an einigen Stellen verbessert werden. Das System ist derzeit nicht in der Lage von mehreren, unterschiedlichen Nutzern genutzt zu werden. Um dies zu erreichen müssen die Suchen einen Vermerk erhalten, wodurch sie einem Nutzer zugeordnet werden können. Weiterhin fehlen der Such-Komponente, aufgrund des Fehlens der GUI-Komponente, Funktionalitäten zum Verwalten der Nutzer und ihren Verknüpfungen zu den sozialen Medien. Des Weiteren sollten die Klassen zum Erhalten der Nachrichten, für die Hintergrundprozesse, nach außen sichtbar gemacht werden.

### 5.3.3 Zukünftige Perspektive

Abseits der Realisierung des Systems, stellt sich die Frage auf, inwiefern das System sinnvoll genutzt werden kann und welche weiteren Perspektiven sich für die Zukunft ergeben. Mithilfe des Systems ist es möglich, Nachrichten aus den sozialen Medien zu erhalten. Daraufhin lässt sich sowohl die Ad-Hoc-, als auch die Tracking-Suche, starten. Durch das Konzept und die Realisierung wurde gezeigt, dass dieses System die erforderlichen Funktionen gewährleistet. Nachrichten können heruntergeladen und gesucht werden. Das System kann darüber hinaus um weitere soziale Medien erweitert werden.

Die Nachrichten von Facebook und Twitter basieren größtenteils auf Texten. Um Videos oder Bilder zu erhalten, auch aus anderen sozialen Medien, können die beigefügten Texte, wie z.B. eine Videobeschreibung, zur Suche verwendet werden. Die Fotos und Videos können über Links neben der Nachricht bereitgestellt werden.

Aufgrund der hohen Datenmenge von erhaltenen Nachrichten, muss für eine übersichtliche Darstellung der Nachrichten gesorgt werden. Eine hohe Anzahl an gefundenen Nachrichten ist für den Nutzer nicht überschaubar. Hier sollten die Nachrichten nach einem geeigneten Sortierverfahren sortiert werden, um den Nutzer die relevanten Nachrichten seiner Suche darzustellen. Zusätzlich ist fraglich, inwiefern die gefundenen Nachrichten für den Nutzer verwendbar sind, da das System nicht auf den Inhalt einer Nachricht wertend eingeht.

Zudem muss bei der Bereitstellung der Nachrichten überprüft werden, ob Nachrichten sich zu sehr ähneln. So kann es vorkommen, dass Nachrichten sich nur in einer Kleinigkeit unterscheiden und dies von der Suche nicht berücksichtigt wird. Für den Nutzer bieten diese Nachrichten keinen Mehrwert und sollten daher gefiltert werden.

Dadurch müssen neue Anforderungen an die Nachrichten gestellt und durch geeignete Testverfahren überprüft werden.

Eine weitere Herausforderung an das System entsteht in Hinblick auf die Wahl der Richtwerte in den konfigurierbaren Dateien, die u.a. für die Nachrichtenbereinigung oder der Wahl der zu ladenden Nachrichten maßgeblich verantwortlich sind. Durch diese Manipulation des Nachrichtenbestandes der Datenbank, wird direkter Einfluss auf den Suchergebnissen und somit auf die Aussagekraft der erhaltenen Nachrichten vorgenommen. In Hinblick auf die konfigurierbaren Dateien sollten Richtwerten vordefiniert werden, um z.B. bei der Nachrichtenbereinigung nach geeigneten Werten zu reinigen. Dies können Erfahrungswerte von Nutzern bzw. Testern des Systems sein.

Eine weitere zukünftige Perspektive ist die rechtmäßige Nutzung der Software. Das System kann aus gesellschaftlichen Gesichtspunkten missbraucht werden. Die Überwachung der Nachrichten kann im eigentlichen Sinne zweckentfremdet werden. Diese Zweckentfremdung erfolgt durch das Überwachen von Quellen, die nicht dem Interesse der Allgemeinheit dienen, wie es von einer Rundfunkanstalt gewünscht ist. Das System ist in der Lage, Personen und Organisationen, innerhalb der sozialen Medien, zu überwachen. So kann z.B. durch die gezielte Angabe von Facebook-Seiten ein Portfolio von Quellen entstehen, die nicht dem eigentlichen Zwecke der Nachrichtenüberwachung, sondern des persönlichen Interesses dienen. Präventivmaßnahmen für die Nutzer der Rundfunkanstalten, wie das Lehren des ordnungsgemäßen Benutzens des Systems, sind dabei denkbar. Durch die ordnungsgemäße Benutzung kann das System zur Gestaltung von Beiträgen der Rundfunkanstalten genutzt werden. Sie ermöglicht zudem das Suchen von Themengebieten und der dazugehörigen Meinung der Öffentlichkeit.

### 5.3.4 Methodische Abstraktion

Das System ist in der Lage Nachrichten aus sozialen Medien zu durchsuchen. Die erstellte Konzeption konnte umgesetzt werden und die Lauffähigkeit wurde durch die Tests im **Abschnitt 5.2** nachgewiesen. Jedoch müssen noch weitere Tests erfolgen, um die Qualität des Systems zu gewährleisten. Weitere Herausforderungen entstehen durch den Einsatz des Systems durch mehrere Nutzer. Hier sind weitere Tests notwendig, um weitere Schwachstellen des Systems aufzuzeigen.

Auch zeigt das System, dass sich das Entwurfsmodell der Fassade für die Suchen auf den sozialen Medien bewährt hat. Die Unabhängigkeit der Komponenten untereinander wurde durch die Fassade sichergestellt. Durch die immer höhere Anzahl an sozialen Medien und der zunehmenden Komplexität dieser, ist die Fassade ein geeignetes Entwurfsmuster um weitere Komponenten hinzuzufügen.

## 6 Fazit

In diesem Kapitel wird die Arbeit zusammengefasst. Anschließend wird ein Ausblick hinsichtlich der Weiterführung der Arbeit, sowie weiteren Themengebieten, mit denen sich befassen kann, gegeben. Dieser Ausblick dient als Anreiz für weitere Entwicklungen am System.

### 6.1 Zusammenfassung

Ziel dieser Arbeit war es, ein System zu entwickeln, das Nachrichten aus sozialen Medien empfangen kann und diese anschließend mittels Suchkriterien durchsucht.

Anfangs wurde ein grundlegendes Verständnis über die in der Arbeit verwandten Themen vermittelt.

Anschließend wurde das Szenario der Arbeit beschrieben und anhand von Anwendungsfällen vertieft. Daraufhin wurden die funktionalen, sowie die nichtfunktionalen Anforderungen an das System gestellt.

Daraus entstand das in **Kapitel 4** entworfene Konzept einer Realisierung des Systems. Bei dem Entwurf des Konzepts dieses Systems wurde darauf geachtet, dass das System die gestellten Anforderungen erfüllen kann.

Es wurde ein Konzept entwickelt, das in der Lage ist, die Nachrichten von sozialen Medien zu durchsuchen. Dafür wurden exemplarisch Komponenten für die sozialen Medien Facebook und Twitter entworfen. Durch den exemplarischen Entwurf der Komponenten wurde gezeigt, dass es möglich ist weitere soziale Medien in das System zu implementieren. Dafür wurde eine zusätzliche Komponente entworfen, die als gemeinsame Schnittstelle das System erweiterbar und wartbar macht.

Das entworfene Konzept wurde anschließend in **Kapitel 5** realisiert, um festzustellen, ob das Konzept umsetzbar ist. Um festzustellen, ob das Konzept umsetzbar ist, wurde ein Prototyp des Systems entwickelt. Dabei werden die Aspekte erläutert, die in dem Prototyp realisiert bzw. nicht realisiert wurden. Anschließend erfolgte eine Bewertung des Prototypens.

Das entwickelte System ist in der Lage, Nachrichten aus unterschiedlichen sozialen Medien zu erhalten. Außerdem kann es diese Nachrichten speichern und anschließend für eine Suche bereitstellen. Dabei ist es möglich die geforderten Arten der Suchen, die in Anwendungsfällen beschrieben wurden, auf den Daten der Nachrichten zu starten. Das System ist zusätzlich in der Lage weitere soziale Medien hinzuzufügen.

Zu einer vollständigen Erfüllung aller Anforderungen kam es jedoch nicht. In der Arbeit werden jedoch weitere Möglichkeiten der Weiterentwicklung des Systems und der zukünftigen Perspektive geliefert.

## 6.2 Ausblick

Das im Rahmen der Arbeit entwickelte System bietet mehrere Möglichkeiten, um verbessert und weiterentwickelt zu werden.

Dazu zählt die Fertigstellung einer geeigneten GUI-Komponente der Sucheingabe und -ausgabe, die zwar konzeptioniert wurde, jedoch nicht in der prototypischen Entwicklung des Systems fertiggestellt wurde. Dadurch können auch nicht erfüllte funktionale Anforderungen, wie das Anzeigen von Nachrichten, erfüllt werden. Neben der Realisierung der Visualisierung, sollte auch die Such-Komponente um weitere Funktionalitäten, die durch die Realisierung der GUI-Komponente entstehen, erweitert werden.

Das System ist darauf ausgelegt unterschiedliche soziale Medien zu verwenden.

In Hinblick auf die Erweiterung des Systems, sind weitere Implementierungen von sozialen Medien in Komponenten sinnvoll. Dadurch können weitere Nachrichten für die Suche bereitgestellt werden.

Weiterhin erfordern die sozialen Komponenten des Systems eine stetige Wartung und Anpassung durch die Änderungen der jeweiligen sozialen Medien.

Durch den langfristigen Einsatz der Software sind weitere Möglichkeiten der Optimierung mit den Komponenten denkbar.

Des Weiteren können die nichtfunktionalen Anforderungen an das System verbessert werden. In Bezug auf die gespeicherten Nachrichten sind weitere Betrachtungen der Themen Datenschutz und Sicherheit sinnvoll.

Außerdem ist das System derzeit auf einem Rechner entwickelt worden. Die Konzeption sieht eine Weiterentwicklung des Systems in Richtung einer Client-Server Version vor, damit es von mehreren Nutzern genutzt werden kann, wie es bereits in **Abschnitt 5.3.2** erwähnt wurde.

Neben der Weiterentwicklung des Systems ist, ist eine weitere Auseinandersetzung mit dem Umgang der Nachrichten vorstellbar. Die gespeicherten Nachrichten ermöglichen das Anbinden von weiteren externen Systemen.

Durch eine Anbindung eines Systems für die Sentiment-Analyse, können die verfügbaren Nachrichten analysiert und bewertet werden, um z.B. Auskunft zu erhalten wie, positiv bzw. negativ, eine Nachricht aufzufassen ist.

Außerdem können die verfügbaren Nachrichten analysiert werden, die die höchste Interaktion mit den Nutzern der sozialen Medien vorweisen können.

Eine weitere zukünftige Tätigkeit könnte die genauere Betrachtung der Verknüpfung von Nachrichten aus klassischen Quellen, wie Nachrichtenagenturen und den Nachrichten aus den sozialen Medien sein. Nachrichten aus den klassischen Medien, können verwendet werden um, aus den Schlagwörtern der Nachrichten, Queries zu bilden. Die Queries könnten dann für eine Suche auf den Nachrichten der sozialen Medien verwendet werden. Ein Ansatz liefert [Ilia 2015]. Dies könnte dazu verwendet werden um z.B. aus bestehenden Nachrichten Folgenachrichten zu erhalten, die Reichweite sowie den Einfluss einer Nachricht zu bestimmen oder weiterführende Themen zu erhalten.

---

Ein anderer Punkt wäre die Analyse der klassischen Nachrichten auf ihren Einfluss in den sozialen Medien (siehe hierzu: [Tsag 2011]).

---

# Anhang A Inhalt der CD-ROM

- BA\_Siano.pdf
- SourceCode

Dieser Arbeit liegt eine CD-ROM bei. Sie enthält eine digitale Version der Arbeit und den Quellcode des prototypisch entwickelten Systems.

# Glossar

**Back-End:**

Das Back-End ist im Gegensatz zum Front-End der Teil einer Client-Server-Architektur oder eines Computersystems, der teilnehmerfern liegt. Betrachtungsmäßig liegt er näher am System. In einer Client-Server-Architektur bildet das Back-End den Server, der die Clients versorgt (vgl. [IT-Wissen 2016c]).

**JSON:**

Datenformat in Textform lesbar für Mensch und Maschine zum Datenaustausch zwischen Anwendungen.

**Like:**

Like bedeutet übersetzt, aus dem Englischen, mögen. Im Umfeld der sozialen Medien bedeutet Like das einer Person ein Beitrag gefällt. Like kann somit auch als „Gefällt Mir“ verstanden werden.

**NewsML:**

NewsML ist ein weitverbreitetes Nachrichtenformat. Es dient zum Austausch von Nachrichteninhalten, sowie ihren Metadaten.  
Die aktuellste Version ist NewsML-G2 (vgl. [IPTC 2016]).

**.Net Framework**

„.NET Framework ist eine Umgebung für die Erstellung, Weitergabe und Ausführung von Webdiensten und weiteren Anwendungen.“ (Quelle: [Microsoft 2016]).

**Retweet**

Ein erneut verschickter Tweet.

**Timeline:**

Sammlung von persönlichen Daten eines Nutzers in chronologische Reihenfolge sortiert.

**Volltextsuche:**

„Der Begriff 'Volltextsuche' bezeichnet einen Suchvorgang, bei dem der gesamte Index einer Suchmaschine nach bestimmten Begriffen durchsucht wird.“ (Quelle: ([SEO 2016])

**WireFrame:**

Skizze der Visualisierung einer GUI-Komponente.

# Abkürzungsverzeichnis

API: Application Programming Interface

GUI: Graphical User Interface

ID: Identifikator

JSON: JavaScript Object Notation

SDK: Software Development Kit

# Abbildungsverzeichnis

Abbildung 1 Marktanteile (weltweit) von Social Media Seiten (in Anlehnung an: [StatCounter 2016a]) .....	10
Abbildung 2 Marktanteile (deutschlandweit) von Social Media Seiten (in Anlehnung an: [StatCounter 2016b]).....	11
Abbildung 3 Anwendungsfälle des Newstools.....	18
Abbildung 4 Übersicht des Systems.....	25
Abbildung 5 Wireframe einer möglichen grafischen Benutzeroberfläche des Systems .....	27
Abbildung 6 Fachliches Datenmodell des Systems .....	31
Abbildung 7 Schnittstelle <code>SocialDocument</code> aus der SocialNews-Komponente.....	33
Abbildung 8 Enumeration <code>SocialMediaSource</code> aus der SocialNews-Komponente .....	33
Abbildung 9 Schnittstelle <code>SocialQueryable</code> aus der SocialNews-Komponente .....	34
Abbildung 10 Schnittstelle <code>Documentable</code> aus der SocialNews-Komponente.....	34
Abbildung 11 Schnittstelle <code>SocialLogin</code> aus der SocialNews-Komponente.....	34
Abbildung 12 Schnittstelle <code>Searchable</code> aus der SocialNews-Komponente .....	35
Abbildung 13 Schnittstelle <code>DBConnection</code> aus der SocialNews-Komponente.....	36
Abbildung 14 Schnittstelle <code>SocialSearch</code> aus der SocialNews-Komponente.....	36
Abbildung 15 Abstrakte Klasse <code>AbstractFacebookFilter</code> aus der Facebook-Komponente.....	37
Abbildung 16 Abstrakte Klasse <code>FacebookDocument</code> aus der Facebook-Komponente .....	38
Abbildung 17 Abstrakte Klasse <code>Page</code> aus der Facebook-Komponente.....	38
Abbildung 18 Schnittstelle <code>IFacebookDBConnector</code> aus der Facebook-Komponente ..	39
Abbildung 19 <code>AbstractTwitterFilter</code> aus der Twitter-Komponente .....	40
Abbildung 20 <code>TwitterDocument</code> aus der Twitter-Komponente .....	41
Abbildung 21 Schnittstelle <code>ITwitterDBConnector</code> aus der Twitter-Komponente .....	42
Abbildung 22 Abstrakte Klasse <code>TwitterSearchQuery</code> aus der Twitter-Komponente ....	43
Abbildung 23 Innenansicht der Komponente SocialNews-Komponente.....	44
Abbildung 24 Innenansicht der Such-Komponente .....	45
Abbildung 25 Innenansicht der Komponente für Facebook .....	46
Abbildung 26 Innenansicht der Komponente für Twitter .....	47
Abbildung 27 Aktivitätsdiagramm zur Nachrichtenbeschaffung aus Facebook (ein Durchlauf) .....	50
Abbildung 28 Aktivitätsdiagramm zur Nachrichtenbeschaffung aus Twitter durch Tracking Suche (ein Durchlauf) .....	51

---

# Tabellenverzeichnis

Tabelle 1 Vergleich zweier Nachrichten aus Twitter.....	29
Tabelle 2 Überblick der Nachrichten aus sozialen Medien.....	57
Tabelle 3 Übersicht der Anforderungen .....	65

# Literaturverzeichnis

[Beck 2012] Beck, Klaus: Das Mediensystem Deutschlands. Strukturen, Märkte, Regulierung. Wiesbaden, s.l.: Imprint VS Verlag für Sozialwissenschaften, 2012.

[Couchbase 2016] Couchbase: Warum NoSQL. 2016. -URL  
<http://www.couchbase.com/de/why-nosql/nosql-database> - Abruf: 05.03.2016

[Datenbank 2016] Datenbank Online Lexikon. 2016. -URL  
[http://wikis.gm.fh-koeln.de/wiki\\_db/Datenbanken/DokumentenorientierteDatenbank](http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/DokumentenorientierteDatenbank)  
- Abruf 22.03.2016

[Facebook 2016a] Facebook: Company Info. 2016. – URL  
<https://newsroom.fb.com/company-info/>. – Abruf: 07.01.2016

[Facebook 2016b] Facebook: Glossary of Terms. 2016. – URL  
<https://www.facebook.com/help/219443701509174/>. – Abruf: 07.01.2016

[Facebook 2016c] Facebook: Sharing. 2016. – URL  
<https://www.facebook.com/help/418076994900119/>. – Abruf: 07.01.2016

[Facebook 2016d] Facebook: Like. 2016. – URL  
<https://www.facebook.com/help/452446998120360/>. – Abruf: 07.01.2016

[Facebook 2016e] Facebook: Group Basics. 2016. – URL  
<https://www.facebook.com/help/162866443847527/>. – Abruf: 07.01.2016

[Facebook 2016f] Facebook: Pages Basics. 2016. – URL  
<https://www.facebook.com/help/281592001947683/>. – Abruf: 08.01.2016

[Facebook 2016g] Facebook: Graph Api Overview. 2016. – URL  
<https://developers.facebook.com/docs/graph-api/overview/>. – Abruf: 07.01.2016

[Facebook 2016h] Facebook: Post. 2016. -URL  
<https://developers.facebook.com/docs/graph-api/reference/v2.5/post> -Abruf 23.02.2016

[Facebook 2016i] Facebook: Rate Limiting on the Graph API 2016 -URL  
<https://developers.facebook.com/docs/graph-api/advanced/rate-limiting> - Abruf 24.02.2016

- [Facebook 2016j] Facebook: `{page-id}/feed` 2016. -URL <https://developers.facebook.com/docs/graph-api/reference/v2.5/page/feed> - Abruf 24.02.2016
- [Facebook 2016k] Facebook: APIs and SDKs 2016. -URL <https://developers.facebook.com/docs/apis-and-sdks#third-party-sdks> -Abruf 24.04.2016
- [Gamma 1996] Gamma, Erich / Helm, Richard / Johnson, Ralph / Vlissides, John: Entwurfsmuster. Elemente wiederverwendbarer objektorientierter Software. München: Addison-Wesley, 1996.
- [Hans 1997] Manecke, Hans-Jürgen / Seeger, Thomas: Zur Entwicklung der Information und Dokumentation in Deutschland. In: Buder/Rehfeld/Seeger/Strauch (Hrsg., 1997): Grundlagen der praktischen Information und Dokumentation. München et al. K.G. Saur, (1997) S.16-60
- [Iliia 2015] Iliakopoulou, K. / Papadopoulos, S. / Kompatsiaris, Y.: News-oriented multimedia search over multiple social networks. Content-Based Multimedia Indexing (CBMI), 2015 13th International Workshop on. Prague. 2015. S. 1-6.
- [IPTC 2016] NewsML. 2016  
-URL <https://iptc.org/standards/newsml-g2/>  
-Abruf 20.01.2016
- [IT Wissen 2016a] ITWissen : Nachricht. 2016  
-URL <http://www.itwissen.info/definition/lexikon/message-Nachricht-MSG.html>  
-Abruf 19.01.2016
- [IT Wissen 2016b] ITWissen : Information. 2016  
-URL <http://www.itwissen.info/definition/lexikon/Information-information.html>  
-Abruf 19.01.2016
- [IT Wissen 2016c] ITWissen: Back-End. 2016  
-URL <http://www.itwissen.info/definition/lexikon/Back-End-back-end.html>  
-Abruf 02.02.2016
- [John 2016] ISO 9126 und ISO 25010. 2016  
-URL <https://www.johner-institut.de/blog/iec-62304-medizinische-software/iso-9126-und-iso-25010/> -Abruf 22.01.2016
- [Klein 2012] Kleinstauber, Hans J./ Eichmann, Ralph: Radio. Eine Einführung. Wiesbaden: VS Verlag für Sozialwissenschaften / Springer Fachmedien Wiesbaden GmbH Wiesbaden, 2012.
- [Microsoft 2016] msdn: Häufig gestellte Fragen zu Microsoft .NET Framework. 2016.  
-URL <https://msdn.microsoft.com/de-de/library/ms973850.aspx#> - Abruf 14.03.2016

- [Mongo 2016] docs.mongodb.org: \$text. 2016. -URL  
<https://docs.mongodb.org/manual/reference/operator/query/text/> -Abruf 21.04.2016
- [NEON 2013] 25. NEON-Plenum, 18. Februar 2013, Frankfurt am Main 2013.  
-URL [http://bvm.org/fileadmin/images/NEON/25\\_Plenum/2013-02-18\\_NEON\\_Vortrag\\_Zahn.pdf](http://bvm.org/fileadmin/images/NEON/25_Plenum/2013-02-18_NEON_Vortrag_Zahn.pdf) -Abruf 12.06.2016
- [Rund 2016] die-mediananstalt.de: Rundfunkstaatsvertrag. 2016 – URL  
[http://www.die-medienanstalten.de/fileadmin/Download/Rechtsgrundlagen/Gesetze\\_aktuell/RStV\\_18.pdf](http://www.die-medienanstalten.de/fileadmin/Download/Rechtsgrundlagen/Gesetze_aktuell/RStV_18.pdf)  
-Abruf 02.02.2016
- [Schmidt 2013] Schmidt, Jan-Hinrik: Social Media. Wiesbaden: Springer VS, 2013.
- [SEO 2016] seo-united.de: Volltextsuche. 2016 -URL  
<http://www.seo-united.de/glossar/volltextsuche/> -Abruf 14.03.2016
- [StatCounter 2016a] StatCounter: Global Stats (worldwide). 2016. - URL  
[http://gs.statcounter.com/#social\\_media-ww-monthly-201501-201507-bar](http://gs.statcounter.com/#social_media-ww-monthly-201501-201507-bar).  
-Abruf: 30.01.2016
- [StatCounter 2016b] StatCounter: Global Stats (germany). 2016 – URL  
[http://gs.statcounter.com/#social\\_media-DE-monthly-201501-201507-bar](http://gs.statcounter.com/#social_media-DE-monthly-201501-201507-bar).  
-Abruf: 30.01.2016
- [Tsag 2011] Tsagakias, Manos / de Rijke, Maarten / Weerkamp, Wouter.: Linking online news and social media. In Proceedings of the fourth ACM international conference on Web search and data mining (WSDM '11). ACM. New York. NY. USA. S.565-574. 2011.
- [Twitter 2016a] Twitter: Unternehmen. 2016. – URL  
<https://about.twitter.com/de/company/>. – Abruf: 31.01.2016
- [Twitter 2016b] Twitter: Twitter Meilensteine. 2016. – URL  
<https://about.twitter.com/de/company/press/milestones/>. – Abruf: 31.01.2016
- [Twitter 2016c] Twitter: Erste Schritte auf Twitter. 2016. – URL  
[https://support.twitter.com/articles/324311#](https://support.twitter.com/articles/324311#/). – Abruf: 07.01.2016
- [Twitter 2016d] Twitter: Was sind Hashtags(# Symbole)?. 2016. – URL  
[https://support.twitter.com/articles/314917#](https://support.twitter.com/articles/314917#/). – Abruf: 07.01.2016
- [Twitter 2016e] Twitter: Was ist eine Twitter Timeline?. 2016. – URL  
<https://support.twitter.com/articles/495848#> – Abruf: 09.01.2016

- [Twitter 2016f] Twitter: FAQs zu Retweets (RT). 2016. – URL  
<https://support.twitter.com/articles/104996#> – Abruf: 09.01.2016
- [Twitter 2016g] Twitter: Trends. 2016. – URL  
<https://support.twitter.com/articles/317695#> – Abruf: 09.01.2016
- [Twitter 2016h] Twitter: Rest API. 2016. – URL  
<https://dev.twitter.com/rest/public> – Abruf: 09.01.2016
- [Twitter 2016i] Twitter: Search API. 2016. – URL  
<https://dev.twitter.com/rest/public/search> – Abruf: 23.02.2016
- [Twitter 2016j] Twitter: Streaming API. 2016. – URL  
<https://dev.twitter.com/streaming/overview> – Abruf: 09.01.2016
- [Twitter 2016k] Twitter: Working with Timelines. 2016. – URL  
<https://dev.twitter.com/rest/public/timelines> – Abruf: 23.02.2016
- [Twitter 2016l] Twitter: Twitter Libraries. 2016. – URL  
<https://dev.twitter.com/overview/api/twitter-libraries> – Abruf: 24.04.2016
- [Werner 2013] Werner, M.: Nachrichtentechnik: Eine Einführung für alle Studiengänge:  
Vieweg+Teubner Verlag. (2013) S.2

---

## **Versicherung über Selbstständigkeit**

*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

*Hamburg, den 15.06.2016*

\_\_\_\_\_