



NEWS MINER

Gruppe 1

Software-Entwicklungspraktikum (SEP)
Sommersemester 2013

Systemspezifikation

Auftraggeber
Technische Universität Braunschweig
Institut für Informationssysteme
Prof. Dr. Wolf-Tilo Balke
Mühlenpfordstraße 23
38106 Braunschweig

Betreuer: Philipp Wille

Auftragnehmer:

Name	E-Mail-Adresse
Maria Joanna Born	maria.born@tu-braunschweig.de
Arne Brüsch	a.bruesch@tu-braunschweig.de
Jana Sarah Riquel	j.riquel@tu-braunschweig.de
Jennifer Sieg	jennifer.sieg@tu-braunschweig.de
Viviane Werner	v.werner@tu-braunschweig.de

Braunschweig, 15. Mai 2013

Versionsübersicht

Version	Datum	Autor	Status	Kommentar
0.1	01.05.2013	Maria Born	in Bearbeitung	Einleitung bearbeiten
0.2	02.05.2013	Jennifer Sieg	in Bearbeitung	Kapitel 2, 2.1, 2.2, 2.4, 2.5 bearbeiten
0.3	02.05.2013	Arne Brüsch	in Bearbeitung	Kriterienerfüllung bearbeiten
0.4	03.05.2013	Jana Riquel	in Bearbeitung	Softwarearchitektur bearbeiten
0.5	05.05.2013	Viviane Werner	in Bearbeitung	Kapitel 2.3, 2.6, 2.7, 2.8, 2.9 bearbeiten
0.6	06.05.2013	Maria Born, Arne Brüsch, Jana Riquel, Jennifer Sieg, Viviane Werner	in Bearbeitung	Korrekturen einarbeiten
0.7	06.05.2013	Maria Born	in Bearbeitung	Glossar bearbeiten
0.8	10.05.2013	Maria Born, Jennifer Sieg, Viviane Werner	in Bearbeitung	Fehler ausbessern
0.9	13.05.2013	Maria Born, Arne Brüsch, Jana Riquel, Jennifer Sieg, Viviane Werner	abgenommen	Abgabe an Betreuer
1.0	15.05.2013	Maria Born, Arne Brüsch, Jana Riquel, Jennifer Sieg, Viviane Werner	abgenommen	fertig gestellt

Inhaltsverzeichnis

1	Einleitung	5
1.1	Projektdetails	7
1.1.1	Registrieren und Anmelden	7
1.1.2	Nachrichtenquellen hinzufügen	9
2	Analyse der Produktfunktionen	11
2.1	Analyse von Funktionalität /F100/: Registrieren	11
2.2	Analyse von Funktionalität /F200/: Anmelden	11
2.3	Analyse von Funktionalität /F300/: Abmelden	13
2.4	Analyse von Funktionalität /F400/: Themengebiete auswählen	13
2.5	Analyse von Funktionalität /F500/: Präferenzen angeben	14
2.6	Analyse von Funktionalität /F600/: Trend folgen	14
2.7	Analyse von Funktionalität /F700/: Trend nicht mehr folgen	15
2.8	Analyse von Funktionalität /F800/: Nachrichtenquelle hinzufügen	16
2.9	Analyse von Funktionalität /F900/: Nachrichtenquelle löschen	16
3	Resultierende Softwarearchitektur	19
3.1	Komponentenspezifikation	19
3.2	Schnittstellenspezifikation	21
3.3	Protokolle für die Benutzung der Komponenten	22
4	Erfüllung der Kriterien	25
4.1	Musskriterien	25
4.2	Sollkriterien	26
4.3	Kannkriterien	26
4.4	Abgrenzungskriterien	26
5	Glossar	27

Abbildungsverzeichnis

1.1	Übersicht über <i>News Miner</i>	6
1.2	Übersicht über das Registrieren und Anmelden	8
1.3	Übersicht über das Hinzufügen einer Nachrichtenquelle	10
2.1	F100/ Registrieren	12
2.2	F200/ Anmelden	12
2.3	Sequenzdiagramm /F300/ Abmelden	13
2.4	F400/ Themengebiete auswählen	14
2.5	F500/ Präferenzen angeben	15
2.6	F600/ Trend folgen	16
2.7	F700/ Trend nicht mehr folgen	17
2.8	F800/ Nachrichtenquelle hinzufügen	18
2.9	Sequenzdiagramm /F900/ Nachrichtenquelle löschen	18
3.1	Aufbau von <i>NewsMiner</i>	20
3.2	Ablauf des <i>Twittercrawlers</i> $\langle C20 \rangle$	23
3.3	Darstellung des <i>RSSGrabbers</i> $\langle C30 \rangle$	24

1 Einleitung

News Miner wird als webbasierte Client-Server Anwendung realisiert, d.h. der Nutzer kann per Webbrowser auf das System zugreifen. Er hat dann die Möglichkeit, RSS-Feeds zu aktuellen Trends zu lesen. Der Server ist dabei für die Ausführung des Programms verantwortlich.

Außerdem verwenden wir eine 3-Schichten-Architektur auf der Serverseite. Die Benutzerschnittstelle ist dabei die grafische Oberfläche. Hierüber kann der Nutzer mit dem System kommunizieren, wenn er zum Beispiel eigene Nachrichtenquellen angibt. Zudem werden ihm hier die Ergebnisse, die der Kern von *News Miner* ausgibt, präsentiert.

Dieser Kern wird in der Systemschicht realisiert und teilt sich in verschiedene Komponenten auf. Ziel der Systemschicht ist es, die vom Nutzer gewählten Nachrichten nach ihrer Aktualität zu beurteilen. Grundlage dieser Bewertung ist die Trendextraktion aus Twitterdaten. So werden in der Systemschicht die Nachrichtenartikel ausgewählt, die sich mit aktuellen Trends befassen. Dazu werden außerdem RSS-Feeds und Tweets gesammelt und gespeichert, da diese die Grundlage bilden.

Auch die Persistenzschicht besteht im Wesentlichen aus zwei Komponenten. Zum einen werden die Tweets und Feeds auf dem Filesystem abgelegt. Hier werden sie durch Lucene indiziert, um eine performante Volltextsuche zu gewährleisten. Zum Anderen werden die Nutzerdaten in einer Derby-Datenbank abgelegt, welche vor allem den Datenschutz gewährleistet, da Derby die Verschlüsselung der Kommunikation mit der Datenbank und der Datenbank Files selbst unterstützt. Die Persistenzschicht sichert also sowohl den Zugriff auf die von Lucene indizierten Daten, als auch den Datenbankzugriff.

Die Applikation wird in Java, Scala und HTML/CSS programmiert. Des Weiteren verwenden wir Play als Webframework sowie zur Nutzerverwaltung.

Im folgenden Statechart in Abbildung 1.1 werden die vom Nutzer sichtbaren Seiten als Zustände modelliert. Daraus ist ersichtlich, wie sich der Nutzer innerhalb der Benutzeroberfläche von einer Seite zu einer anderen bewegen kann. Hierbei werden alle Funktionen mit ihrem deutschen Namen aus dem Pflichtenheft und alle Schaltflächen mit ihrem englischen Namen, also so ,wie sie dem Nutzer angezeigt werden, bezeichnet.

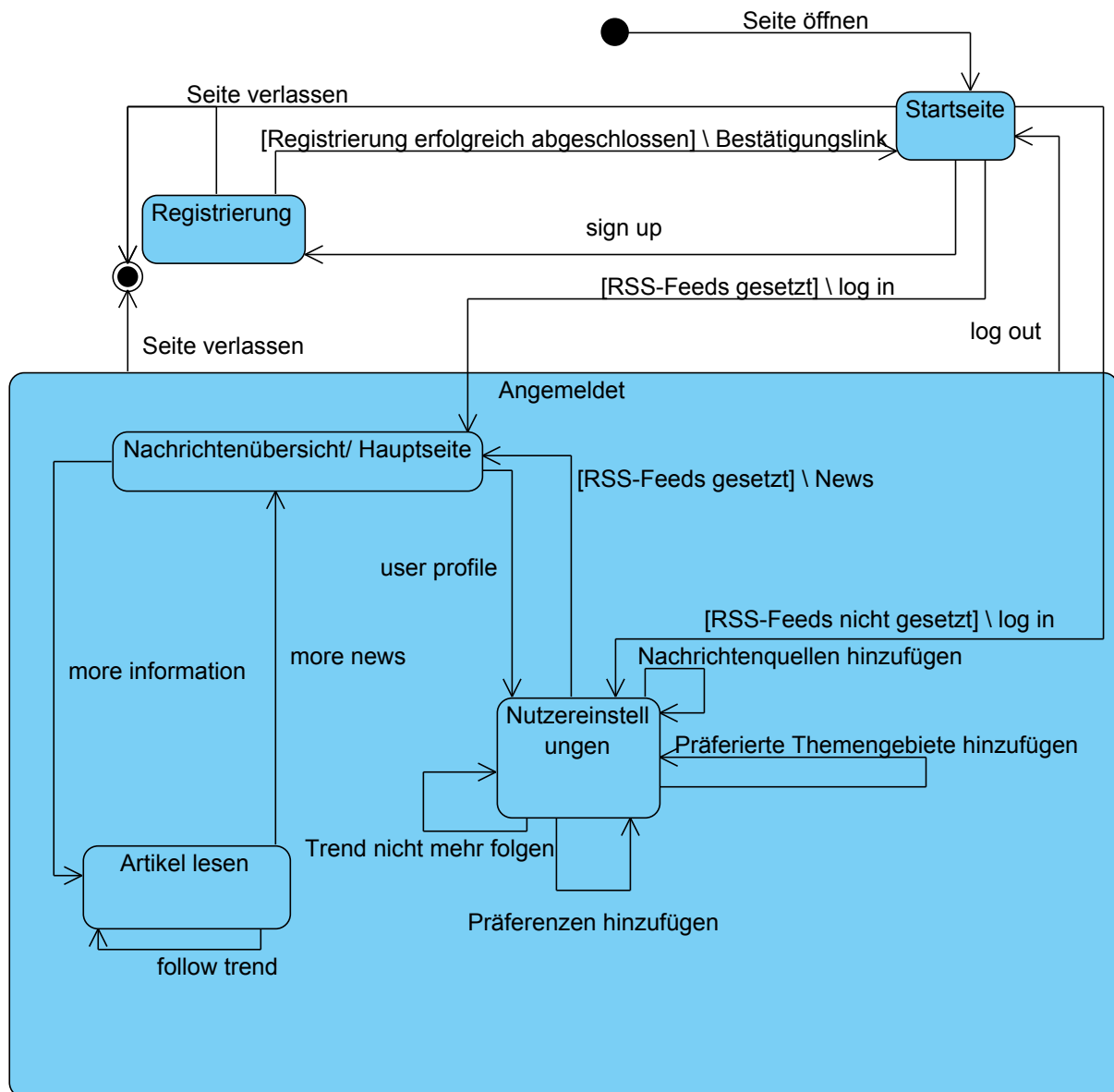


Abbildung 1.1: Übersicht über *News Miner*

1.1 Projektdetails

Um wichtige Aspekte der Anwendung *News Miner* genauer zu erläutern, werden in diesem Kapitel verschiedene Funktionen der Anwendung aufgeführt und beschrieben. In den folgenden Unterkapiteln sind die Funktionen des Registrierens, des Anmeldevorgangs und des Hinzufügens einer Nachrichtenquelle genauer erklärt.

1.1.1 Registrieren und Anmelden

Das folgende Aktivitätsdiagramm in der Abbildung 1.2 beschreibt den Workflow bei der Registrierung und Anmeldung eines Nutzers. Will sich ein Nutzer registrieren, so muss er zunächst auf der Startseite den “create an account”-Button anklicken. Dadurch wird ihm vom System die Seite angezeigt, auf der er seine Daten in Textfelder eintragen kann. Anschließend muss er diese durch einen Klick auf “sign up” bestätigen. Dies bewirkt, dass das System seine Daten in der Datenbank hinterlegt und eine Bestätigungsmail versendet. In dieser ist ein Bestätigungslink enthalten. Sobald der Nutzer diesem gefolgt ist, ist seine Registrierung abgeschlossen. Nun kann er sich anmelden. Dabei wird zunächst überprüft, ob er bereits Nachrichtenquellen angegeben hat. Wenn er sich das erste Mal angemeldet hat, ist dies natürlich nicht der Fall. Dann wird ihm vom System zunächst das Nutzerprofil angezeigt, verbunden mit einer Aufforderung, mindestens eine Nachrichtenquelle anzugeben. Hat er dies getan, so kann er durch einen Klick auf “News” in der Navigationsleiste auf die Nachrichtenübersichtsseite gelangen. Hier werden ihm die Nachrichten seiner Nachrichtenquellen zu den aktuellen Trends angezeigt.

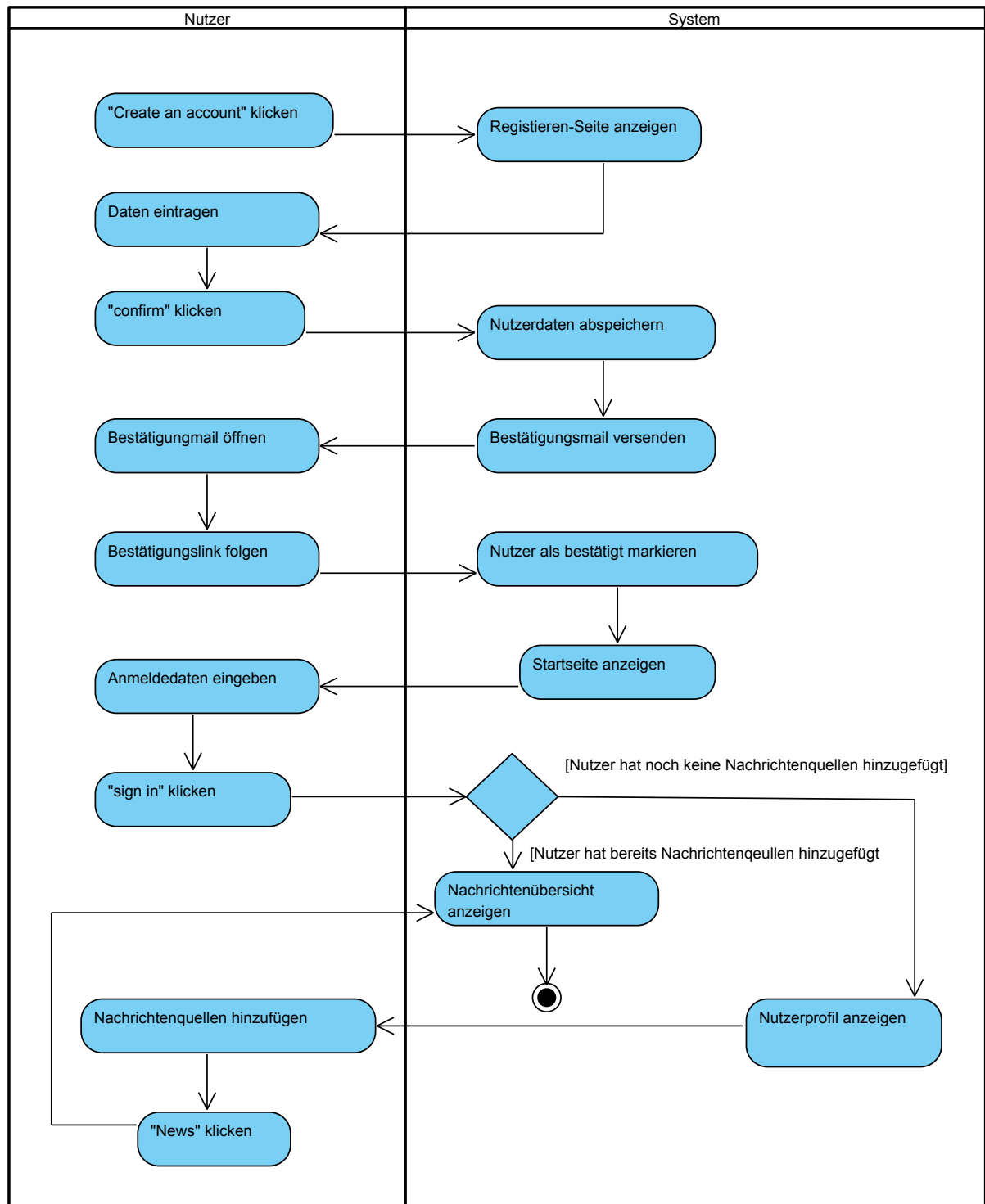


Abbildung 1.2: Übersicht über das Registrieren und Anmelden

1.1.2 Nachrichtenquellen hinzufügen

Das Aktivitätsdiagramm in Abbildung 1.3 wird der Workflow beim Hinzufügen einer Nachrichtenquelle beschrieben. Hierbei wird vorausgesetzt, dass sich der Nutzer bereits registriert und angemeldet hat. Will der Nutzer Nachrichtenquellen hinzufügen, muss er zunächst auf das Nutzerprofil gehen. Das Aktivitätsdiagramm nimmt an, dass ihm diese Seite bereits angezeigt wird. Hier sind unter der Überschrift “RSS-Feeds” mehrere Textfelder zu sehen. In eines dieser Felder muss der Nutzer die URL seines gewünschten RSS-Feeds eintragen und anschließend den “confirm”-Button klicken. Hat er dies getan, wird der Feed vom System überprüft. Findet das System unter der angegebenen URL keinen RSS-Feed, so ist dies auf einen Eingabefehler zurückzuführen. Deshalb wird dem Nutzer eine Fehlermeldung ausgegeben und er muss die Eingabe erneut bearbeiten. Findet das System an der angegebenen URL einen RSS-Feed, so extrahiert es den Titel. Danach speichert er die URL und den Titel des RSS-Feeds ab und ordnet sie dem Nutzer zu. Nun wird dem Nutzer der Titel anstatt des Textfeldes angezeigt. Anstelle des “confirm”-Buttons steht nun der Button “change title” dar. Sollte der Nutzer mit dem Titel nicht zufrieden sein, kann er diesen Button anklicken und den Titel ändern. Dieser wird vom System ebenfalls abgespeichert. Des Weiteren hat der Nutzer die Möglichkeit, weitere Nachrichtenquellen hinzuzufügen. Falls dies sein Wunsch ist, so wird der gesamte Workflow erneut durchlaufen, ansonsten ist der Workflow beendet und der Nutzer kann andere Funktionen von *News Miner* nutzen.

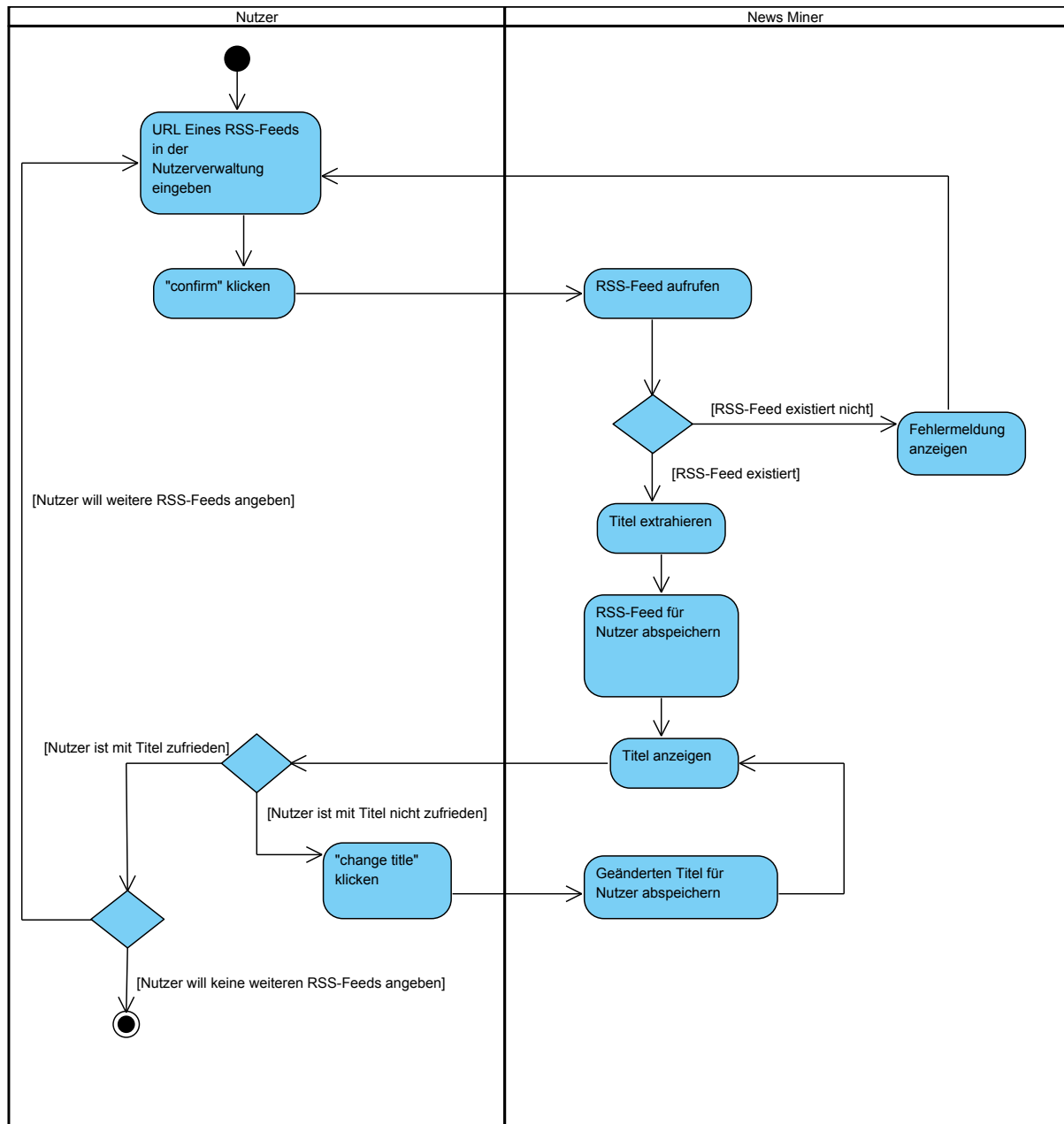


Abbildung 1.3: Übersicht über das Hinzufügen einer Nachrichtenquelle

2 Analyse der Produktfunktionen

Im folgenden Kapitel werden die Produktfunktionen sowie ggf. die nicht-funktionalen Anforderungen aus dem Pflichtenheft mittels Sequenzdiagrammen modelliert. Dadurch sollen die Interaktionen der jeweiligen Funktionen mit den verschiedenen Komponenten von *News Miner* verdeutlicht werden. Zusätzlich werden die Vorgänge in den Abbildungen näher erläutert. Dies stellt insgesamt die Basis dar, um später eine geeignete Architektur festlegen zu können.

2.1 Analyse von Funktionalität /F100/: Registrieren

Die Funktion “Registrieren” ist essentiell, um die Webanwendung *News Miner* verwenden zu können, muss dafür aber auch nur einmal pro Benutzer durchgeführt werden. Denn mit dieser Funktion erstellt der Benutzer ein Nutzerkonto. Dieses wiederum wird benötigt, um die individuellen Angaben und die persönlichen Daten zu speichern und dem Nutzer zu einem späteren Zeitpunkt jederzeit wieder zuzuordnen, worauf letztlich weitere Funktionen aufbauen.

Im folgenden Sequenzdiagramm in Abbildung 2.1 werden nun die Komponenten dargestellt, die an diesem Prozess beteiligt sind.

Zunächst verdeutlicht die Abbildung, wie der Nutzer seine persönlichen Daten, also eine E-mail-Adresse, ein Passwort und einen Nutzernamen, in der *NewsMinerGUI* angibt. Diese wiederum gibt sie an das *System* weiter, von wo aus sie dann an den *NewsMinerStorage* übergeben und dort letztlich abgespeichert werden.

Nachdem dieser Prozess erfolgreich war, wird eine Bestätigungsmail von der *NewsMinerGUI* an die angegebene E-mail-Adresse versendet. Durch Anklicken des Links in der versandten E-mail seitens des Nutzers, wird dieser von der *NewsMinerGUI* auf die Startseite von *News Miner* weitergeleitet. Damit ist die Registrierung erfolgreich abgeschlossen.

2.2 Analyse von Funktionalität /F200/: Anmelden

In dem Sequenzdiagramm in Abbildung 2.2 wird die Funktionalität “Anmelden” näher dargestellt.

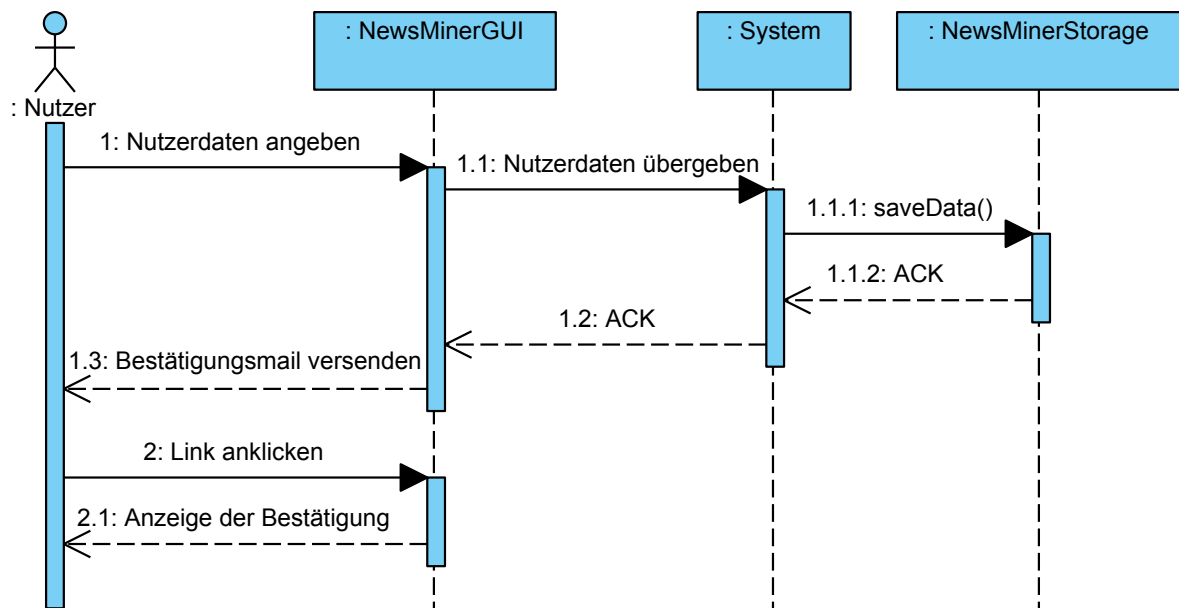


Abbildung 2.1: F100/ Registrieren

Diese Funktion wird benötigt, damit der Nutzer auf sein vorher erstelltes Nutzerkonto zugreifen kann, um die Funktionen von *News Miner* verwenden zu können.

In der Abbildung gibt der Benutzer als Erstes die benötigten Daten, also die E-mail-Adresse und das Passwort, in der *NewsMinerGUI* an. Diese wiederum werden über das *System* an den *NewsMinerStorage* übergeben, wo die eingegebenen Angaben mit den dort abgespeicherten Daten abgeglichen werden. Wenn dies erfolgreich war, leitet der *NewsMinerStorage* eine positive Rückmeldung an das System weiter, welche dann an die *NewsMinerGUI* weitergegeben wird. Anschließend wird dem Benutzer die Hauptseite von *News Miner* angezeigt.

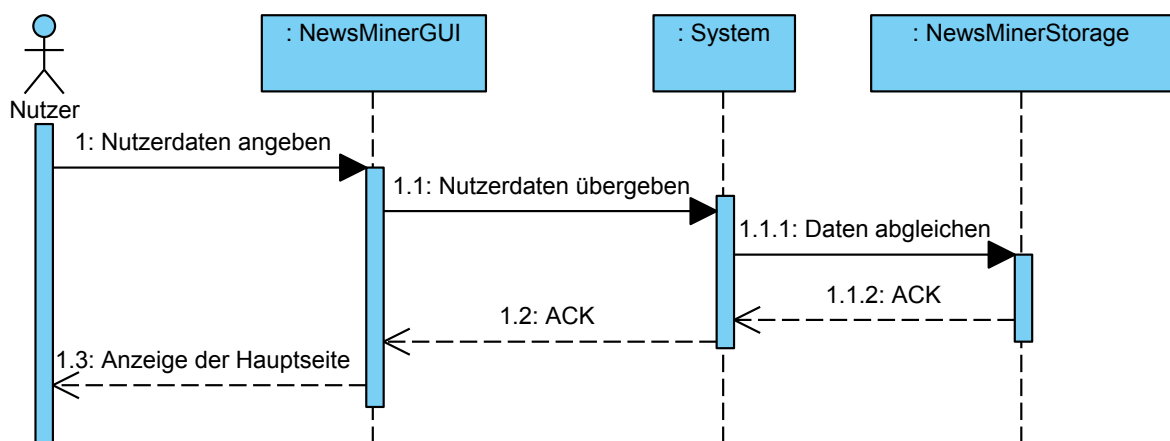


Abbildung 2.2: F200/ Anmelden

2.3 Analyse von Funktionalität /F300/: Abmelden

Im Sequenzdiagramm in Abbildung 2.3 wird die Funktionalität “Abmelden” genauer dargestellt. Diese Funktion wird benötigt, damit sich der Nutzer aus seinem Nutzerkonto wieder abmelden kann, wenn er die Seite verlassen möchte.

In der Abbildung wird dargestellt, dass der Benutzer auf den “Logout”-Button in der *NewsMinerGUI* klickt, welche anschließend dem Nutzer eine Bestätigung anzeigt, dass er sich erfolgreich abgemeldet hat.

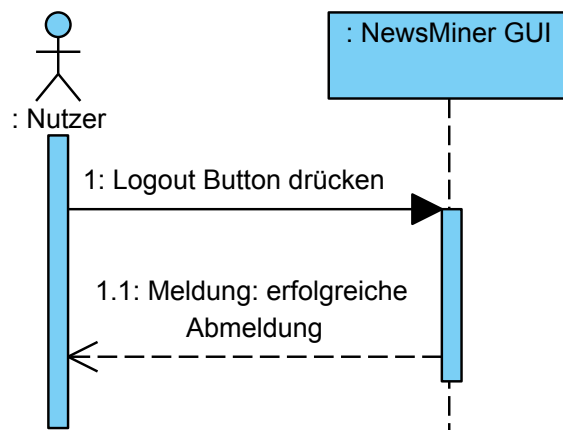


Abbildung 2.3: Sequenzdiagramm /F300/ Abmelden

2.4 Analyse von Funktionalität /F400/: Themengebiete auswählen

Die Funktion “Themengebiete auswählen” ist ein Zusatz, um dem Nutzer eine individuelle, seinen persönlichen Interessen entsprechende Auswahl an Nachrichtenartikeln zu bieten.

Dies wird in Abbildung 2.4 in dem Sequenzdiagramm verdeutlicht.

Als Erstes wählt der Nutzer ein Thema über die *NewsMinerGUI* aus. Dies wird über das *System* an den *NewsMinerStorage* übergeben und dort in der Datenbank abgespeichert, sodass die Daten dem Nutzer jederzeit zugeordnet werden können. Anschließend fordert das *System* Nachrichtenartikel zum jeweiligen ausgewählten Thema aus dem *NewsMinerStorage* an. Nun übergibt das *System* die ausgewählten Nachrichtenartikel an die *NewsMinerGUI*. Diese wiederum ist dafür zuständig, die Nachrichten dem Nutzer anzuzeigen.

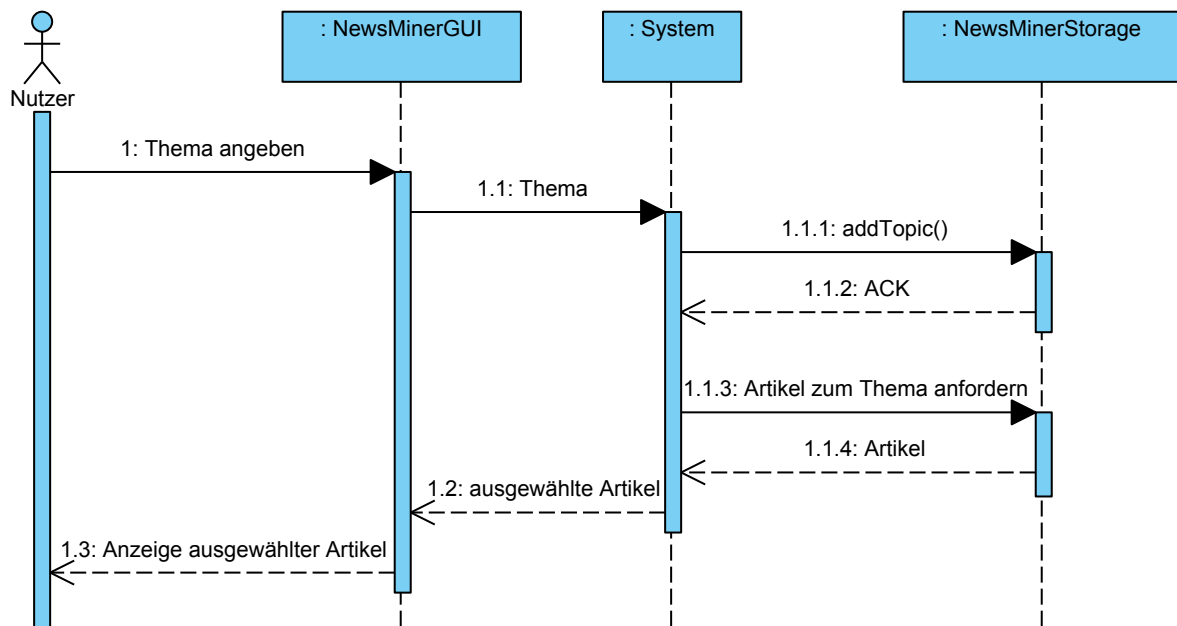


Abbildung 2.4: F400/ Themengebiete auswählen

2.5 Analyse von Funktionalität /F500/: Präferenzen angeben

“Präferenzen angeben” ist eine Funktionalität, die die Hauptaufgabe von *News Miner* erweitert, um dem Nutzer mehr Vielfalt zu bieten. Es steht ihm vollkommen frei, diese zu verwenden. In dem folgenden Sequenzdiagramm in Abbildung 2.5 wird die Funktion bildlich dargestellt.

Zunächst ist in der Abbildung dargestellt, wie der Nutzer in der *NewsMinerGUI* eine persönliche Präferenz angibt. Diese wird dann über das *System* im *NewsMinerStorage* abgespeichert, damit sie dem Nutzer jederzeit wieder zugeordnet werden kann. Danach fordert dann das *System* vom *NewsMinerStorage* die abgespeicherten Nachrichtenartikel, die einen inhaltlichen Bezug zu der angegebenen Präferenz aufweisen, an. Die ausgewählten Nachrichten werden anschließend wieder von dem *System* an die *NewsMinerGUI* übergeben und dem Nutzer angezeigt.

2.6 Analyse von Funktionalität /F600/: Trend folgen

“Trend folgen” ist eine Funktion, die *News Miner* erweitert. Sie soll dem Nutzer die Möglichkeit geben, favorisierten Trends zu folgen, um weiterhin darüber informiert zu werden. In Sequenzdiagramm in Abbildung 2.6 wird diese Funktion bildlich dargestellt.

Um diese Funktion nutzen zu können, muss der Benutzer als erstes auf den “Trend folgen”-Button in der *NewsMinerGUI* klicken. Die Informationen über den ausgewählten Nachrichtenartikel

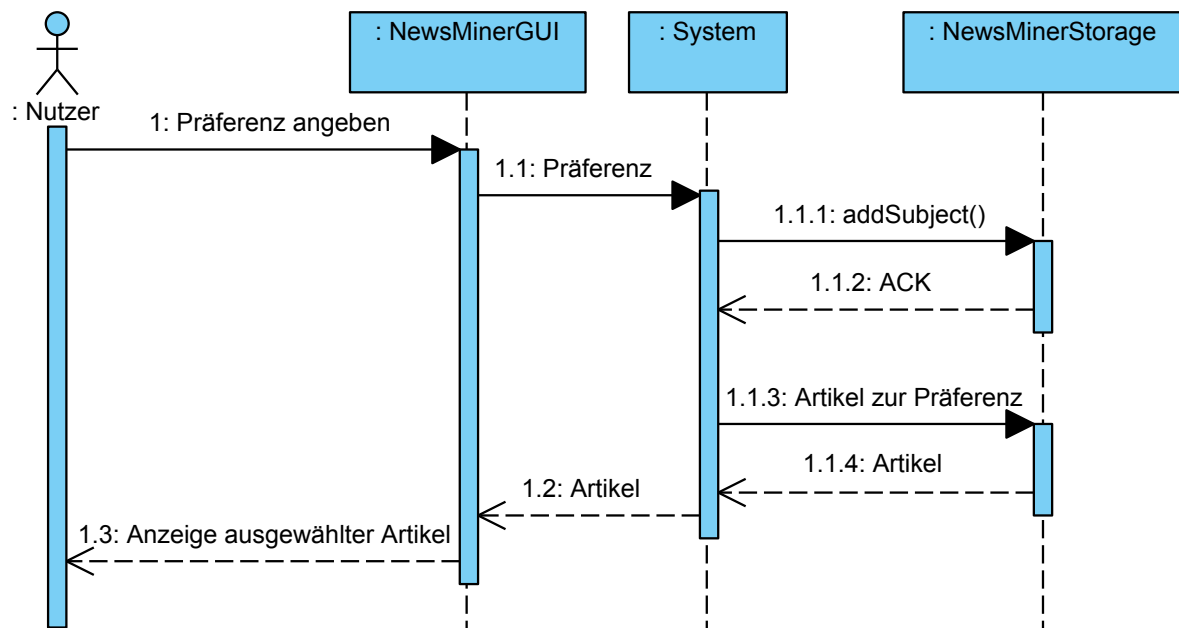


Abbildung 2.5: F500/ Präferenzen angeben

werden anschließend an das *System* weitergegeben und im *NewsMinerStorage* abgespeichert. Im nächsten Schritt erfolgt von dort ausgehend eine Bestätigung an das *System* und dann an die *NewsMinerGUI*, welche dem Nutzer den ausgewählten Trend anzeigt.

2.7 Analyse von Funktionalität /F700/: Trend nicht mehr folgen

Die Funktion “Trend nicht mehr folgen”, ist ebenfalls eine Erweiterung von *News Miner*. Dem Nutzer wird es mit dieser Funktion ermöglicht, einem Trend, dem er bisher gefolgt ist, wieder aus seinem Nutzerprofil zu löschen.

Dies wird in dem Sequenzdiagramm in Abbildung 2.7 bildlich veranschaulicht.

Der Nutzer kann in der *NewsMinerGUI* einen Trend abwählen, dem er bisher gefolgt ist. Diese Information wird an das *System* und von da aus an den *NewsMinerStorage* weitergeleitet und dort abgespeichert, also aus diesem gelöscht. Nach diesem Vorgang wird von dem *NewsMinerStorage* über das *System* eine Bestätigung an die *NewsMinerGUI* gegeben, welche dem Nutzer eine Bestätigung darüber anzeigt, dass dieser ab sofort dem ausgewählten Trend nicht mehr folgt.

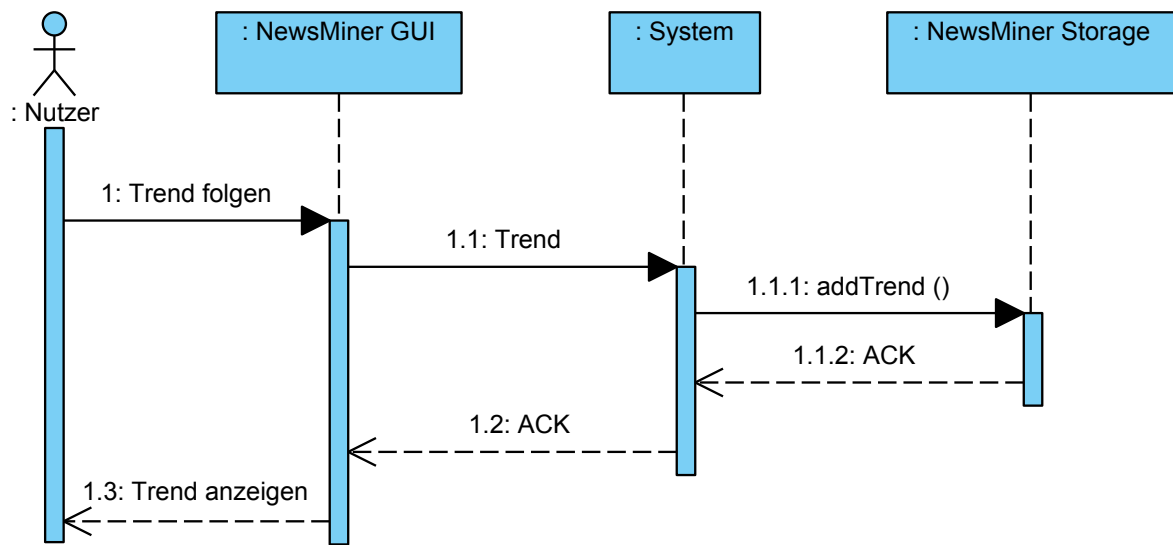


Abbildung 2.6: F600/ Trend folgen

2.8 Analyse von Funktionalität /F800/: Nachrichtenquelle hinzufügen

“Nachrichtenquelle hinzufügen” gehört zu den erweiternden Funktionen der Anwendung *News Miner*, die der Nutzer mindestens einmal ausführen muss, um Nachrichten zu erhalten. Diese Funktion und die beteiligten Komponenten werden im Sequenzdiagramm in Abbildung 2.8 dargestellt.

In der *NewsMinerGUI* muss zunächst eine Nachrichtenquelle ausgewählt werden. Diese Nachrichtenquelle wird anschließend über das *System* in den *NewsMinerStorage* eingepflegt. Falls diese Nachrichtenquelle schon im *System* eingetragen ist, wird sie nur im Nutzerprofil gespeichert. Nach der erfolgreichen Speicherung wird über das *System* eine Bestätigung an die *NewsMinerGUI* geleitet, welche dem Nutzer eine Meldung anzeigt, dass die Nachrichtenquelle gespeichert wurde.

2.9 Analyse von Funktionalität /F900/: Nachrichtenquelle löschen

News Miner bietet dem Nutzer die Möglichkeit, ausgewählte Nachrichtenquellen wieder aus dem Nutzerprofil zu löschen, wenn er keine Nachrichtenartikel mehr von dieser Quelle erhalten möchte.

Im folgenden Sequenzdiagramm in Abbildung 2.9 wird diese Funktion bildlich erläutert.

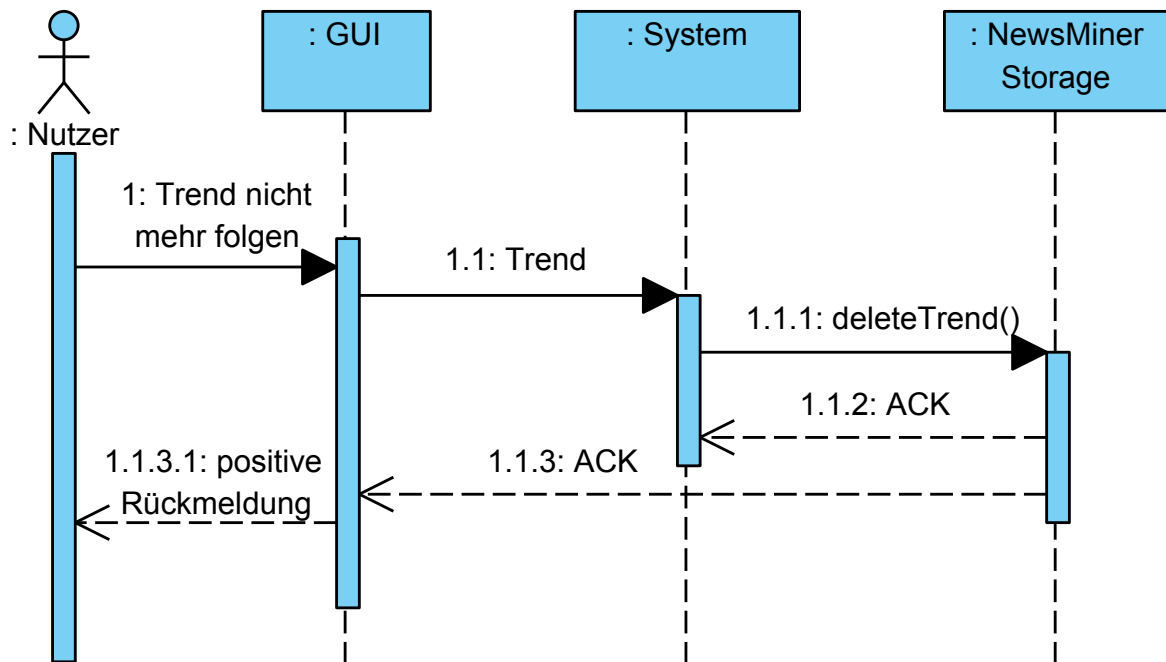


Abbildung 2.7: F700/ Trend nicht mehr folgen

Um eine bisher im Nutzerprofil gespeicherte Nachrichtenquelle wieder zu löschen, muss der Nutzer in der *NewsMinerGUI* diese gewünschte Nachrichtenquelle auswählen. Diese Information wird daraufhin über das *System* an den *NewsMinerStorage* geleitet, in dem diese Nachrichtenquelle aus dem Nutzerprofil gelöscht wird. Nach der erfolgreichen Entfernung dieser Quelle wird eine Bestätigung über das *System* an die *NewsMinerGUI* geleitet, welche dem Nutzer mitteilt, dass die Nachrichtenquelle gelöscht wurde.

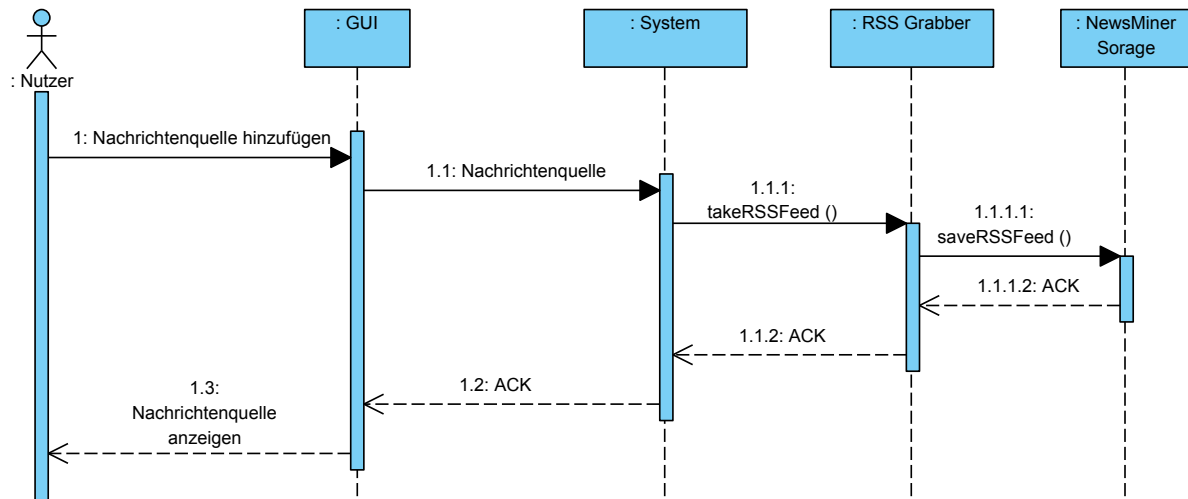


Abbildung 2.8: F800/ Nachrichtenquelle hinzufügen

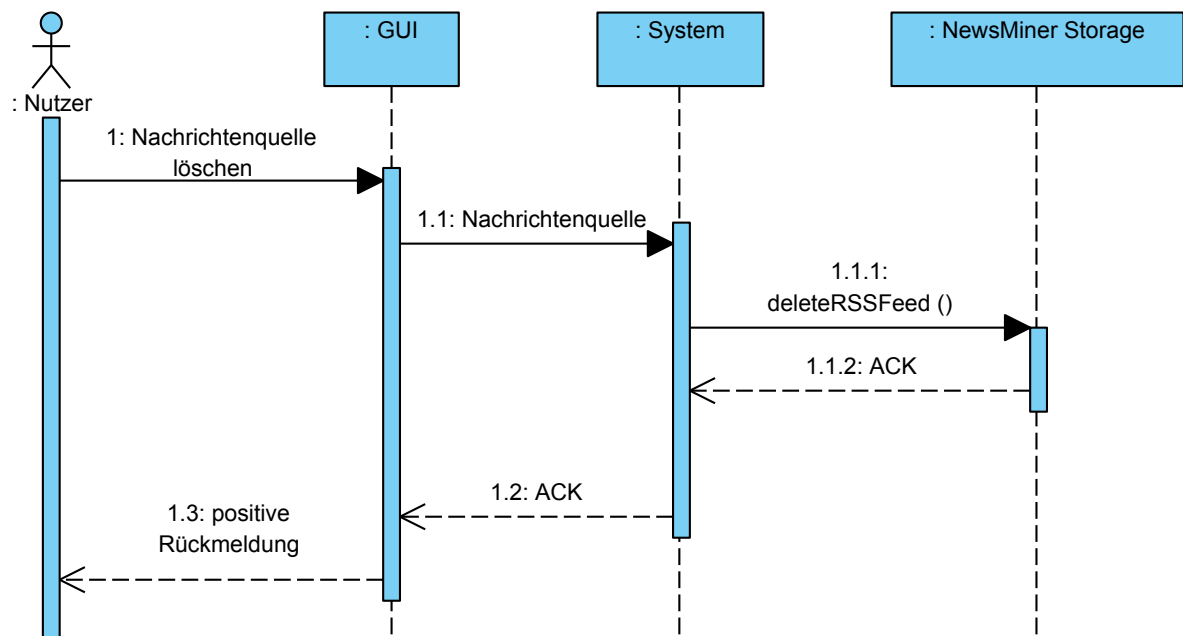


Abbildung 2.9: Sequenzdiagramm /F900/ Nachrichtenquelle löschen

3 Resultierende Softwarearchitektur

Der *News Miner* wird durch eine Vielzahl von Subsystemen, die durch einzelne Komponenten realisiert werden, implementiert. Dies wird in dem Komponentendiagramm in Abbildung 3.1 bildlich dargestellt. Die für den Nutzer sichtbare Komponente ist die *NewsMinerGUI*. Auf seiner grafischen Oberfläche ist es ihm möglich, sein persönliches Nutzerprofil zu erstellen. Im Gegenzug stellt die *NewsMinerGUI* ihm nun gefilterte Artikel dar. Dieser Vorgang wird durch die Komponenten des eigentlichen Systems realisiert. Dies sind der *TwitterCrawler*, der *RSS-Grabber*, der *TrendExtractor* und der *NewsAggregator*. Diese Subsysteme arbeiten alle zusammen, um aus den auf Twitter veröffentlichten Tweets und verschiedensten RSS-Feeds aktuelle Artikel zu extrahieren. Um diese Vorgänge umsetzen zu können, müssen verschiedene Informationen (zwischen-)gespeichert werden. Hierfür ist der *NewsMinerStorage* zuständig, welcher die gesamte Speicherverwaltung von *News Miner* übernimmt und somit auch sämtliche Nutzerprofile abspeichert. All diese Komponenten sind für sich abgeschlossene Subsysteme, die erst in ihrer Verknüpfung die vollständige Funktionalität von *NewsMiner* bilden.

3.1 Komponentenspezifikation

Komponente $\langle C10 \rangle$: NewsMinerGUI

Die *NewsMinerGUI* hat die Aufgabe, eine Verbindung zum Nutzer herzustellen. Der Nutzer kann auf der von der *NewsMinerGUI* zur Verfügung gestellten grafischen Oberfläche sämtliche persönlichen Nutzereinstellungen tätigen, wie die Registrierung bei *News Miner*, die Anmeldung und die Bearbeitung des Nutzerprofils. Die *NewsMinerGUI* übernimmt somit die Nutzerverwaltung und speichert alle relevanten Daten im *NewsMinerStorage* ab. Des Weiteren stellt die *NewsMinerGUI* dann individuell für jeden Nutzer passende aktuelle Nachrichtenartikel, welche sie aus dem *NewsMinerStorage* erhält, grafisch dar.

Komponente $\langle C20 \rangle$: TwitterCrawler

In dem Statechart in Abbildung 3.2 wird der *TwitterCrawler* näher dargestellt. Er ist dafür zuständig, dass auf Twitter neu veröffentlichte Tweets kontinuierlich gespeichert werden. Er liest somit ständig den Twitter-Stream mit und speichert in Intervallen Tweets im *NewsMinerStorage* ab. Hierbei filtert er zudem die Tweets nach ihrer verwendeten Sprache und wählt nur die

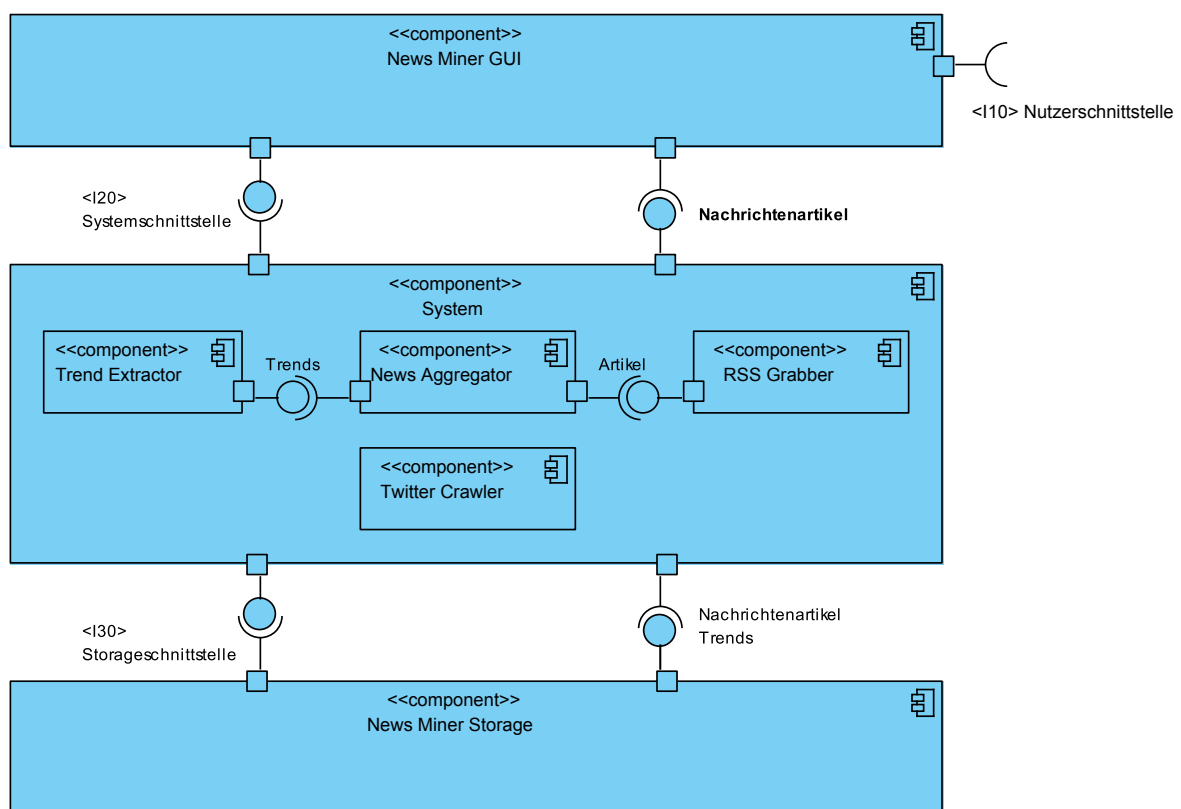


Abbildung 3.1: Aufbau von *NewsMiner*

Englischen aus. Gespeichert werden dann diese Tweets mit all ihren Metadaten vom FileMaker in einer Textdatei, um ihre Herkunft auch zu einem späteren Zeitpunkt nachvollziehen zu können.

Komponente $\langle C30 \rangle$: RSSGrabber

Der *RSSGrabber* wird in dem Statechart in Abbildung 3.3 verdeutlicht. Dieser holt regelmäßig RSS-Feeds und lädt neu dazugekommene Artikel nach. Die aus dem Internet geladenen Nachrichtenartikel werden nun mit Überschrift und Volltext im *NewsMinerStorage* gespeichert. Für den Fall, dass kein Volltext zu einem Artikel existiert, wird dieser nicht weiter verwendet.

Komponente $\langle C40 \rangle$: TrendExtractor

Der *TrendExtractor* erhält die vom *TwitterCrawler* gesammelten Tweets und durchsucht diese mit Hilfe eines Suchalgorithmus auf auffällig häufig auftauchende Wörter. Diese gefunden Trends werden nun mit dem Zeitpunkt ihres Auftretens und den passenden Schlagwörtern im *NewsMinerStorage* abgespeichert.

Komponente $\langle C50 \rangle$: NewsAggregator

Der *NewsAggregator* erhält die Nachrichtenartikel, die vom *RSSGrabber* geladen wurden, und die Trends, die der *TrendExtractor* identifiziert hat. Nun ist es die Aufgabe des *NewsAggregator*, die Artikel mit den Trends in Verbindung zu bringen. Hierfür werden die Artikel auf die Schlagwörter der Tweets, die die Trends behandeln, durchsucht. Diese Paarungen werden dann im *NewsMinerStorage* abgespeichert und können auf der Oberfläche der *NewsMinerGUI* dargestellt werden.

Komponente $\langle C60 \rangle$: NewsMinerStorage

Der *NewsMinerStorage* übernimmt die gesamte Speicherverwaltung des *News Miner*. Er speichert sämtliche Daten, die in den einzelnen Komponenten des Systems entstanden sind, ab und stellt ihnen diese zu einem späteren Zeitpunkt zur Weiterverarbeitung zur Verfügung. Somit werden im *NewsMinerStorage* Nutzerdaten, Tweets, Trends und Nachrichtenartikel verwaltet.

3.2 Schnittstellenspezifikation

Schnittstelle $\langle I10 \rangle$: Nutzerschnittstelle

Operation	Beschreibung
input()	Der Nutzer gibt über die Tastatur Eingaben an den <i>News Miner</i> . Die <i>NewsMinerGUI</i> arbeitet dann mit diesen Daten weiter.

Schnittstelle $\langle I20 \rangle$: Systemschnittstelle

Operation	Beschreibung
addUser()	Der Nutzer registriert sich neu bei <i>News Miner</i> . Seine Daten werden an die <i>NewsMinerGUI</i> übergeben, ein neues Nutzerprofil wird angelegt und im <i>NewsMinerStorage</i> abgelegt.
signInUser()	Der Nutzer meldet sich nach erfolgreicher Registrierung bei <i>News Miner</i> an. Seine Angaben werden an die <i>NewsMinerGUI</i> übergeben und überprüft.
addRSSFeed()	Der Nutzer fügt neue Nachrichtenquellen hinzu. Hier wird die URL vom gewünschten RSS-Feed an die <i>NewsMinerGUI</i> übergeben, welche dann den jeweiligen RSS-Feed zum zugehörigen Nutzerprofil hinzufügt.
addTopic()	Der Nutzer fügt ein präferiertes Themengebiet zu seinem Nutzerprofil in der <i>NewsMinerGUI</i> hinzu. Dies wird im <i>NewsMinerStorage</i> abgelegt und es können dem Nutzer fortan Nachrichtenartikel zu diesem Themengebiet angezeigt werden.
saveTrend()	Der Nutzer folgt einem bestimmten Artikel. Dieses wird der <i>NewsMinerGUI</i> übergeben und im <i>NewsMinerStorage</i> abgelegt. Zu dieser Neuigkeit werden fortan präferiert Artikel angezeigt.

Schnittstelle $\langle I30 \rangle$: Storageschnittstelle

addSubject()	Der Nutzer fügt ein präferiertes Thema zu seinem Nutzerprofil hinzu. Dies wird im <i>NewsMinerStorage</i> abgelegt und es können dem Nutzer fortan Nachrichtenartikel zu diesem Thema angezeigt werden.
saveRSS(String)	Es werden die vom <i>RSSGrabber</i> geladenen Nachrichtenartikel im <i>NewsMinerStorage</i> abgespeichert.
saveTweet()	Hier werden, die vom <i>TwitterCrawler</i> mitgelesenen Tweets von Twitter im <i>NewsMinerStorage</i> abgelegt.
addTrends()	Die vom <i>TrendExtractor</i> gefilterten Trends werden im <i>NewsMinerStorage</i> abgespeichert.
saveNews()	Nachdem der <i>NewsAggregator</i> zu jedem Trend einen zugehörigen Nachrichtenartikel gefunden hat wird zu den beiden zugehörigen Quellen im <i>NewsMinerStorage</i> ein Verweis abgespeichert.

3.3 Protokolle für die Benutzung der Komponenten

Die für den *News Miner* verwendeten Komponenten *TwitterCrawler* und *RSSGrabber* sind auch in einem anderen Zusammenhang gut einsetzbar, also zur Wiederverwendung geeignet. Der *TwitterCrawler* sorgt dafür, dass permanent Tweets abgespeichert werden. Diese können auch für

andere Zwecke, wie zum Beispiel Forschungen in den Sozialwissenschaften, verwendet werden. Dasselbe gilt auch für den *RSSGrabber*, auch die von ihm gespeicherten Nachrichtenartikel können in einem anderen Zusammenhang sinnvoll genutzt werden. Sie sind zudem auch in andere Softwareprodukte einfach einzubinden, da sie eigenständig arbeiten und keine weiteren Abhängigkeiten haben, somit sind sie unabhängige Subsysteme.

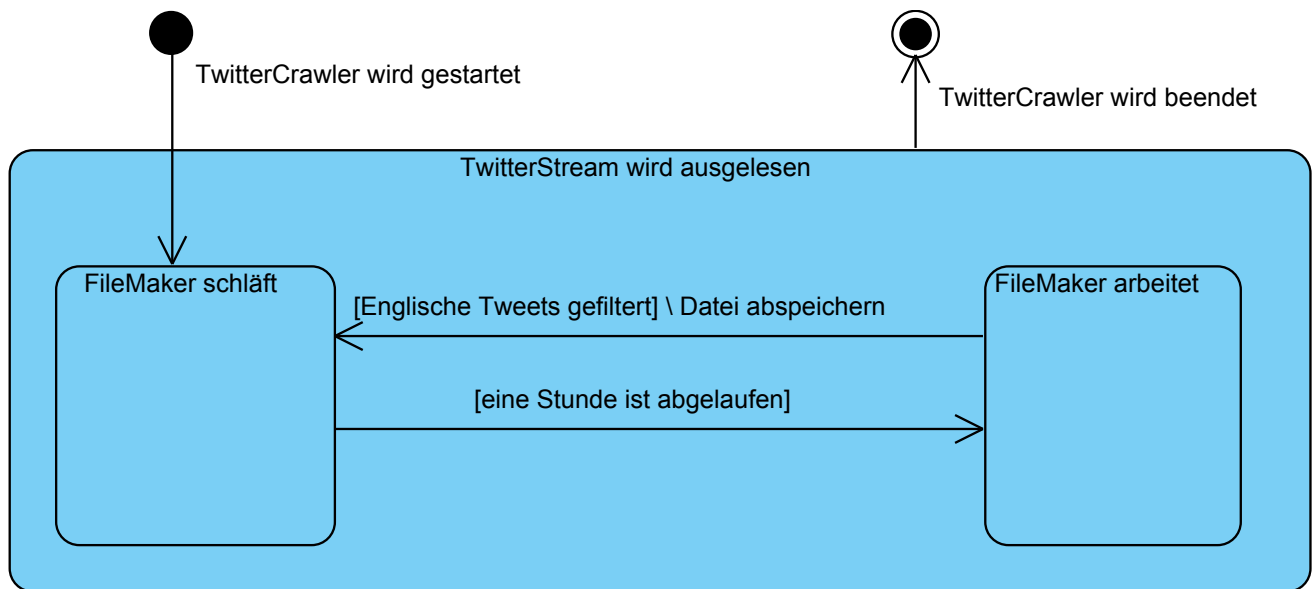


Abbildung 3.2: Ablauf des *Twittercrawlers* $\langle C20 \rangle$

Die anderen Komponenten des Systems - der *TrendExtractor* und der *NewsAggregator* - sind nicht zur Weiterverwendung geeignet. Sie sind sehr stark von den anderen Komponenten abhängig und genau auf den *News Miner* zugeschnitten. Somit ist die Einbindung in ein anderes Produkt sehr umständlich und somit wenig sinnvoll. Dasselbe gilt für die *NewsMinerGUI*, also ihr grafische Oberfläche, und den *NewsMinerStorage*, welches für die Speicherverwaltung zuständig ist. Sie sind explizit für das System von *News Miner* entwickelt worden und nur schwer auf andere Systeme zu übertragen.

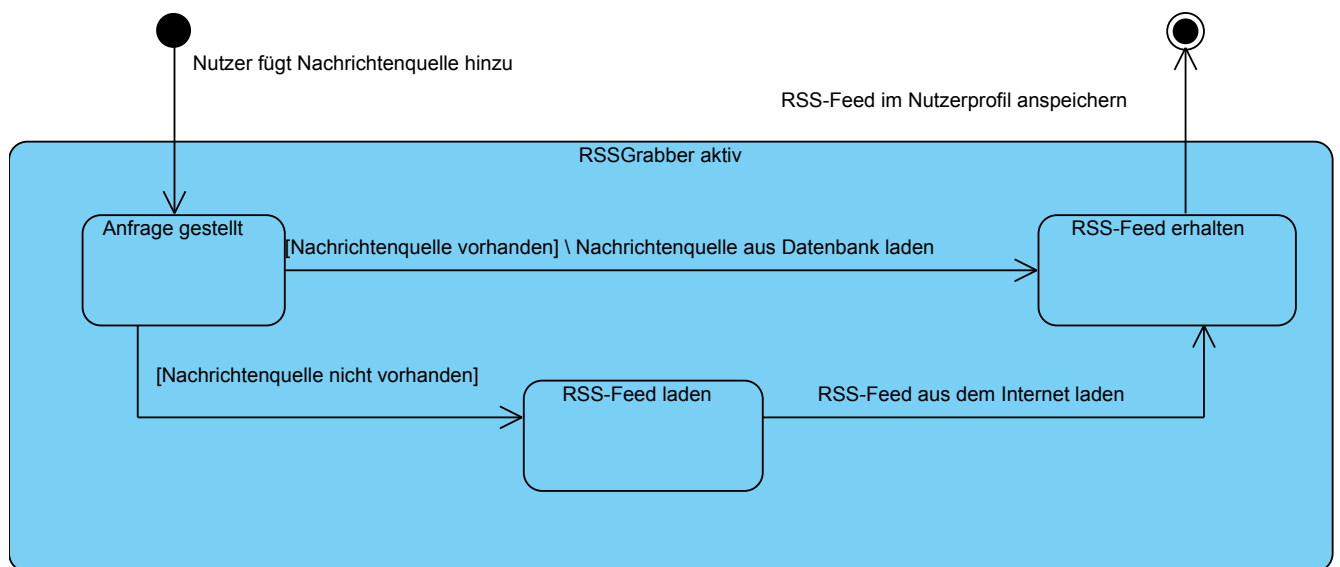


Abbildung 3.3: Darstellung des *RSSGrabbers* ⟨C30⟩

4 Erfüllung der Kriterien

4.1 Musskriterien

$\langle RM1 \rangle$ RSS-Feeds empfangen

Die *RSSGrabber*-Komponente $\langle C30 \rangle$ von *NewsMiner* empfängt laufend RSS-Feeds und speichert diese auf dem Kundensystem zwischen. Es muss dabei genügend Speicherplatz vorhanden sein. Außerdem müssen alte Artikel wieder gelöscht werden, damit sie nicht versehentlich als neu gewertet werden.

$\langle RM2 \rangle$ Twitter Stream lesen

Analog zum *RSSGrabber* $\langle C30 \rangle$ liest der *TwitterCrawler* $\langle C20 \rangle$ fortlaufend den Sample-Stream der Twitter-API mit und puffert diesen auf dem Kundensystem. Es ist dabei wichtig, dass alte Twitter-Daten (=Tweets) gelöscht werden, da sonst sehr viel Speicherplatz verschwendet wird. Zudem wurden bereits Trends extrahiert, weshalb die Tweets nicht länger benötigt werden.

$\langle RM3 \rangle$ Trends extrahieren

Das Extrahieren von Trends entfällt auf den *TrendExtractor* $\langle C40 \rangle$. Dieser benötigt den *TwitterCrawler* $\langle C20 \rangle$, um aus dessen gepufferten Daten Trends extrahieren zu können. Eine zentrale Speicherverwaltung über den *NewsMinerStorage* $\langle C60 \rangle$ ist hier notwendig, um Konflikte zu vermeiden.

$\langle RM4 \rangle$ Unter Verwendung der Trends RSS-Nachrichten filtern

Der *NewsAggregator* $\langle C50 \rangle$ filtert RSS-Nachrichten mit Trends, die er aus dem *NewsMinerStorage* $\langle C60 \rangle$ bezieht. Er benötigt den *RSSGrabber* $\langle C30 \rangle$ und *TrendExtractor* $\langle C40 \rangle$, um arbeiten zu können.

$\langle RM5 \rangle$ gefilterte Nachrichten in ansprechender Form darstellen

Diese Aufgabe entfällt auf die *NewsMinerGUI* $\langle C10 \rangle$, ein Webserver, der auf dem Play-Framework aufbauen wird.

$\langle RM6 \rangle$ Nutzerverwaltung

Die notwendige Nutzerverwaltung wird die *NewsMinerGUI* $\langle C10 \rangle$ übernehmen. Hierbei ist natürlich besonders auf Datensicherheit zu achten.

4.2 Sollkriterien

⟨RS1⟩ Die grafische Benutzeroberfläche soll übersichtlich gestaltet und intuitiv bedienbar sein.

Die *NewsMinerGUI* ⟨C10⟩ muss von Anfang an nicht nur von den Entwicklern, sondern immer wieder von Testpersonen beurteilt werden, um hohe Intuitivität zu erreichen.

4.3 Kannkriterien

⟨RC1⟩ Trends folgen

⟨RC2⟩ Präferenzen angeben

⟨RC3⟩ Themen angeben

All diese Informationen sind nutzerbezogen und müssen entsprechend in der Nutzerverwaltung nach ⟨RM4⟩ abgespeichert werden. Ebenso muss der *NewsAggregator* ⟨C50⟩ diese Nutzerbezogenheit widerspiegeln und individuelle Nachrichten aggregieren.

4.4 Abgrenzungskriterien

⟨RW1⟩ Twitter

Durch die konsequente Trennung von Front- und Backend in *NewsMiner* (Frontend: *NewsMinerGUI*, Backend: alle anderen Komponenten) wird größtmögliche Kapselung erreicht, wodurch die Berechnungen vollständig vor dem Nutzer verborgen sind.

⟨RW2⟩ Verlässlichkeit

Aufgrund des modularen Aufbaus und der Nutzung von bewährten Server-Betriebssystemen sowie teilweise clientseitiger Berechnung wird automatisch eine hohe Verlässlichkeit erreicht.

⟨RW3⟩ Trends ohne Feed

Der *NewsAggregator* filtert zunächst die vorhandenen Nachrichtenartikel mit allen Trends. Gibt es Trends ohne zugeordneten Artikel, werden diese ignoriert.

5 Glossar

3-Schichten-Architektur Die 3-Schichten-Referenz-Architektur ist ein weit verbreitetes Strukturierungsprinzip für Software. Hierbei werden die Bestandteile des Systems in Schichten aufgeteilt. Zwischen diesen Schichten kann der Nachrichtenaustausch nur über Schnittstellen stattfinden. Die Anwendung solcher Architekturen führt zu einer geringeren Kopplung verschiedener Systembestandteile untereinander.

API (Application Programming Interface) bzw. Programmierschnittstelle, die eine Anbindung von Anwendungsprogrammen auf Funktionen eines Systems oder eines Programms ermöglicht.

Applikation bzw. Anwendungssoftware, womit der Benutzer nicht systemtechnische Funktionalitäten unterstützen oder bearbeiten kann.

Backend ist der Teil der Anwendung, der näher am Betriebssystem ist. Im Gegensatz dazu steht das Frontend.

Framework ist eine Rahmenstruktur, die Regeln definiert, wie ein Programmierer eine Anwendung realisieren soll. Zudem stellt es Bibliotheken und Werkzeuge bereit, um die Programmierung zu erleichtern.

Frontend ist der Teil der Anwendung, der näher am Nutzer ist. Im Gegensatz dazu steht das Backend.

Hauptseite ist die Seite, auf die der Nutzer kommt, wenn er sich erfolgreich am System angemeldet hat.

indexieren etwas zu einem Index hinzufügen.

Index eine Art Register oder Nachschlagewerk für Daten, welches die Suchzeit verringert.

internetfähiges Endgerät ist z.B. ein PC, Tablet PC, Smartphone oder Fernseher.

Kapselung schützt Informationen vor dem Zugriff von außen. Es werden klare Schnittstellen für den Zugriff vorgegeben, wodurch unkontrollierter Zugriff vermieden wird.

Persistenzschicht ist in einer 3-Schichten-Architektur für die Speicherverwaltung zuständig.

Puffer, puffern ein Puffer ist ein Speicher für die Zwischenlagerung von Daten. Daten zu puffern bedeutet also, sie in einem dafür vorgesehenen Zwischenspeicher abzulegen, um nach einem kurzen Zeitraum darauf zuzugreifen.

RSS-Feed eine simple und strukturierte Veröffentlichung in einem XML-Format, die bei Veränderung einer Website automatisch über die jeweiligen Änderungen informiert.

Server ist ein zentraler und leistungsstarker Netzwerkrechner, der Daten oder Ressourcen bereit stellt.

Startseite ist die Seite, auf der sich der Nutzer anmelden bzw. registrieren kann, und die angezeigt wird, wenn ein beliebiger Internetnutzer auf den Internetdienst gelangt.

Trend ist ein steigender oder fallender Wandlungsprozess.

Tweet ist eine kurze Nachricht mit maximal 140 Zeichen, die über den Internetdienst Twitter veröffentlicht wird.

Workflow (dt. Arbeitsablauf) ist eine Beschreibung für eine Folge von Arbeitsschritte.