Technische Universität Braunschweig

ISF

News Miner

Gruppe 1

Software-Entwicklungspraktikum (SEP)
Sommersemester 2013

Testspezifikation

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, 10. Juli 2013

Versionsübersicht

Version	Datum	Autor	Status	Kommentar
1.2	08.07.2013	Maria Born	abgenommen	TrendExtractor Testfälle bearbeitet
1.3	09.07.2013	Jennifer Sieg	abgenommen	RSSGrabber Testfälle bearbeitet
1.4	09.07.2013	Viviane Werner	abgenommen	TwitterCrawler Testfälle bearbeitet
1.5	09.07.2013	Arne Brüsch	abgenommen	NewsAggregator Testfälle bearbeitet
1.6	09.07.2013	Jana Riquel	abgenommen	Abnahmetest bearbeitet
1.7	09.07.2013	Maria Born	abgenommen	Integrationstest bearbeitet
1.8	09.07.2013	Maria Born, Arne Brüsch, Jana Riquel, Jennifer Sieg, Viviane Werner	abgenommen	Testprotokolle bearbeitet
1.9	10.07.2013	Maria Born, Arne Brüsch, Jana Riquel, Jennifer Sieg, Viviane Werner	abgenommen	Korrektur gelesen
2.0	10.07.2013	Maria Born, Arne Brüsch, Jana Riquel, Jennifer Sieg, Viviane Werner	abgenommen	Testspezifikation fertiggestellt

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Das Testen der Software ist im Entwicklungsprozess unverzichtbar, um die geforderte Qualität zu erreichen, und sollte somit bei keiner Softwareentwicklung fehlen. Die Ziele des Testens bestehen unter anderem darin Fehler zu ermitteln, die Qualität der Software zu erhöhen und die Softwarespezifikation zu verifizieren.

Aus diesem Grund wird die Webanwendung *News Miner* auf verschiedene Qualitätsmerkmale getestet, mit dem Ziel die gewünschten Anforderungen zu erfüllen. *News Miner* ist eine Webanwendung, die die Möglichkeit bietet aktuelle und wichtige Informationen für den Nutzer aus einem individuellen Pool von Nachrichtendiensten (in Form von RSS-Feeds) zu filtern.

Die Applikation muss alle geforderten Funktionen realisieren, welche aus Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz und Änderbarkeit bestehen. Hierbei können Präferenzen angegeben werden, sodass die Angemessenheit erhöht wird. Zudem sollen die Nutzerdaten bei der Speicherung durch Sicherheitsmechanismen geschützt werden. Des Weiteren sollte die Applikation eine normale Fehlertoleranz aufweisen.
Eine weitere Anforderung, die gestellt wird, ist die einfache und intuitive Bedienbarkeit der Anwendung, um die Einarbeitungszeit sehr gering zu halten. Außerdem darf die Reaktionszeit für Nutzereingaben nicht länger als 5 Sekunden betragen. Zudem besteht die Option, die Anwendung durch individuelle Einstellungen zu modifizieren. Eine hohe Modifizierbarkeit wird durch einen gut nachvollziehbaren und analysierbaren Quellcode gewährleistet, da dies Änderungen vereinfacht.

2 Testplan

Im folgenden Abschnitt wird der Testplan für *News Miner* dargelegt. Es wird auf die einzelnen Komponenten und Funktionen eingegangen, aus denen *News Miner* besteht. Außerdem wird die Vorgehensweise der einzelnen Tests beschrieben.

2.1 Zu testende Komponenten

News Miner liegt als Client-Server Architektur vor. Der Client stellt eine Internetseite dar. Die serverseitige Software zerfällt in die folgenden Komponenten:

TwitterCrawler: Der Twittercrawler liest ständig den Twitter-Stream mit und speichert in einer Sliding-Window-Technik Tweets im NewsMinerStorage ab, d.h. neue Tweets werden kontinuierlich gespeichert, während die ältesten kontinuierlich gelöscht werden.

RSSGrabber: Der RSS-Grabber holt regelmäßig RSS-Feeds und lädt neu dazugekommene Artikel nach. Diese werden im NewsMinerStorage abgespeichert.

TrendExtractor: Der TrendExtractor entdeckt Trends in den abgespeicherten Tweets im NewsMinerStorage. Diese werden in geeigneter Form dort wieder abgelegt.

NewsAggregator: Der NewsAggregator filtert mit den Trends des TrendExtractors die Artikel des RSSGrabbers und gibt das Ergebnis an den NewsMinerStorage ab.

NewsMinerGUI: Die NewsMinerGUI stellt die gefilterten Artikel in ansprechender Form dar.

NewsMinerStorage: Der NewsMinerStorage übernimmt die gesamte Speicherverwaltung von *News Miner*.

2.2 Zu testende Funktionen/Merkmale

- $\langle RM1 \rangle$ RSS-Feeds empfangen: RSS-Feeds werden empfangen.
- $\langle RM2 \rangle$ Twitter Stream lesen: Der Twitter-Stream wird empfangen.
- $\langle RM5 \rangle$ Darstellung der Nachrichten: Die empfangegen RSS-Feeds werden auf der Hauptseite dargestellt.

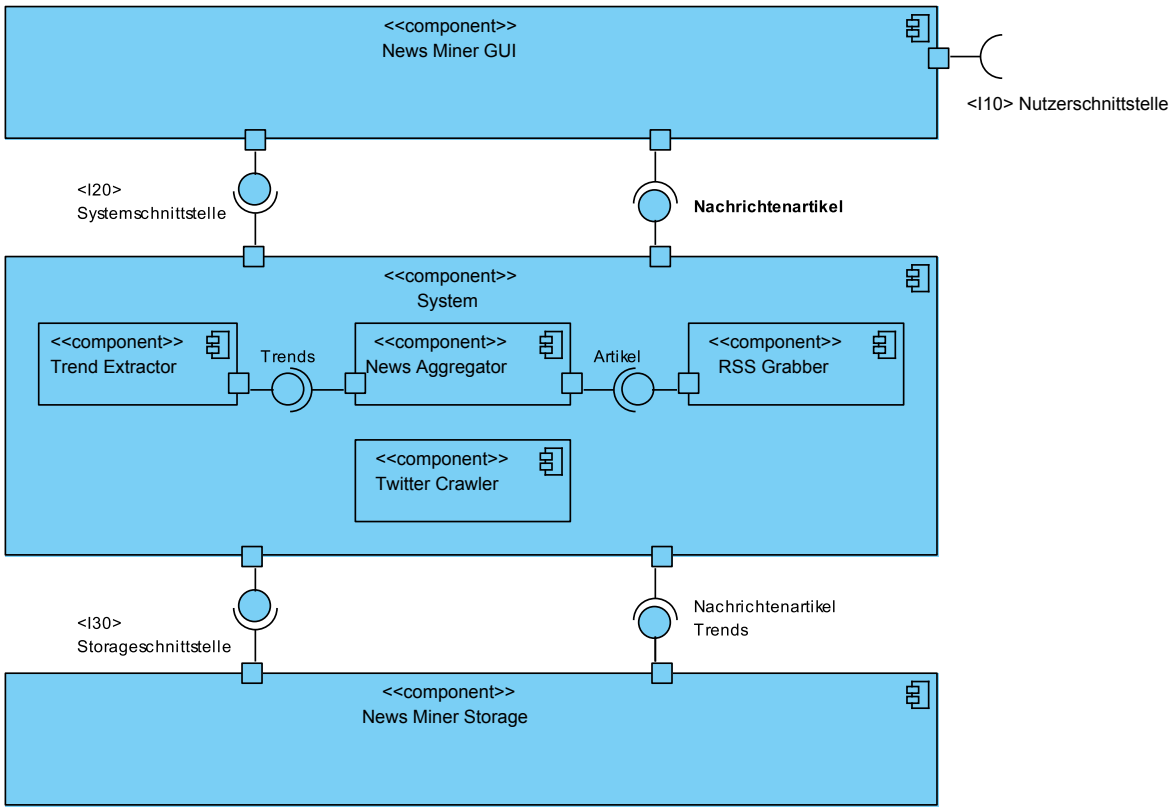

Abbildung 2.1: Aufbau von NewsMiner

- $\langle RM6 \rangle$ Nutzerkonto: Es ist möglich, ein Nutzerkonto anzulegen, sich damit anzumelden und sich damit abzumelden.
- $\langle F10 \rangle$ Registrieren: Es ist möglich, sich zu registrieren.
- $\langle F20 \rangle$ Anmelden: Es ist möglich, sich nach der Registrierung anzumelden.
- $\langle F20 \rangle$ Anmelden, $\langle F30 \rangle$ Abmelden: Wenn der Nutzer angemeldet ist, ist es ihm möglich, sich wieder abzumelden.
- $\langle F20 \rangle$ Anmelden, $\langle F80 \rangle$ Nachrichtenquellen hinzufügen: Wenn ein Nutzer angemeldet ist, ist es ihm möglich, Nachrichtenquellen hinzuzufügen
- $\langle F20 \rangle$, $\langle F80 \rangle$, $\langle F90 \rangle$ Nachrichtenquellen Löschen: Wenn ein Nutzer angemeldet ist und bereits Nachrichtenquellen hinzugefügt hat, ist es möglich, diese wieder zu entfernen.
- $\langle F20 \rangle$, $\langle F80 \rangle$, $\langle F40 \rangle$ Präferierte Themengebiete hinzufügen: Wenn ein Nutzer angemeldet ist und bereits Nachrichtenquellen hinzugefügt hat, ist es ihm möglich, Themengebiete als präferiert zu markieren.
- $\langle F20 \rangle$, $\langle F80 \rangle$,$\langle F50 \rangle$ Präferenzen hinzufügen: Wenn ein Nutzer angemeldet ist und bereits Nachrichtenquellen hinzugefügt hat, ist es ihm möglich, Präferenzen anzugeben und diese werden berücksichtigt.
- $\langle F20 \rangle$, $\langle F80 \rangle$, $\langle F60 \rangle$ Trend folgen: Wenn ein Nutzer angemeldet ist und bereits Nachrichtenquellen hinzugefügt hat, ist es ihm möglich, einem Trend zu folgen.
- $\langle F20 \rangle$, $\langle F80 \rangle$, $\langle F60 \rangle$, $\langle F70 \rangle$ Trend nicht mehr folgen: Wenn ein Nutzer angemeldet ist, bereits Nachrichtenquellen hinzugefügt hat und zuvor einen Trend verfolgt hat ist es ihm möglich, dem Trend nicht mehr zu folgen.
- $\langle Q10 \rangle$ leicht verständliche Fehlermeldungen: die Fehlermeldungen sind leicht Verständlich.
- $\langle Q20 \rangle$ intuitive Bedienbarkeit/Benutzbarkeit: Das Produkt ist intuitive bedienbar und gut benutzbar.
- $\langle Q30 \rangle$ Unabhängigkeit vom Betriebssystem: Das Produkt kann unabhängig vom Betriebssystem genutzt werden.
- $\langle Q40 \rangle$ Fehlerhafte Nutzereingaben beeinträchtigen nicht die Funktionsweise: Fehlerhafte Nutzereingaben werden abgefangen, sodass die Software zuverlässig Funktioniert.
- $\langle Q50 \rangle$ Antwortzeit von höchstens 5 Sekunden: die Zeit, die das System braucht um auf Nutzereingaben zu antworten, darf nicht länger als 5 Sekunden sein.

2.3 Nicht zu testende Funktionen

Die semantische Korrektheit der Funktionen $\langle RM3 \rangle$ - “Trends extrahieren” und $\langle RM4 \rangle$ - “RSS-Nachrichten nach Trends filtern” kann nicht getestet werden. Es ist nicht überprüfbar, ob ein von uns extrahierter Trend tatsächlich ein Trend ist, da dazu Trends manuell extrahiert werden müssten. Da nicht einmal die manuelle Extraktion eindeutig sein kann, sind diese Funktionen nicht testbar.

Bei der Funktion $\langle F50 \rangle$ - “Präferenzen angeben” kann lediglich getestet werden, ob die dafür angezeigten Nachrichtenartikel die Wortkombination der Präferenz enthalten. Das heißt insbesondere, dass die Relevanz der angezeigten Nachrichtenartikel nicht zu testen ist.

2.4 Vorgehen

Abnahme- und Funktionstest:

Die Funktionen $\langle RM1 \rangle$ - “RSS-Feeds empfangen”, $\langle RM2 \rangle$ - “Twitter Stream lesen”, $\langle RM6 \rangle$ - “Nutzerverwaltung” sind von uns im Vorfeld des Abnahmetests zu überprüfen.

Die Funktion $\langle RM1 \rangle$ - “RSS-Feeds empfangen” wird getestet, indem die empfangenen RSS-Feeds ausgegeben werden. Ebenso wird mit dem Twitter Stream bezüglich $\langle RM2 \rangle$ - “Twitter Stream lesen” verfahren.

$\langle RM6 \rangle$ - “Nutzerverwaltung” wird durch das Anlegen von 5 Nutzerkonten überprüft. Weitere Tests dieser Funktion werden im Verlauf der Abnahmetests durch den Kunden durchgeführt.

Die Anwendungsfälle aus dem Pflichtenheft werden über das Web-Interface getestet. Hierbei muss der Kunde jede Funktionalität auf Korrektheit prüfen. Die Funktion $\langle F10 \rangle$ - “Registrieren” wird getestet, indem der Kunde versucht sich zu registrieren. Erst wenn dies geschehen ist, kann der Test der weiteren Funktionen durchgeführt werden. Dazu muss sich der Kunde mit seinem Nutzerkonto anmelden. So wird auch die Funktion $\langle F20 \rangle$ - “Anmelden” getestet. Anschließend muss er versuchen verschiedene Nachrichtenquellen hinzuzufügen um die Funktion $\langle F80 \rangle$ - “Nachrichtenquellen hinzufügen” zu testen. Jetzt können die Funktionen $\langle F40 \rangle$ - $\langle F70 \rangle$ sowie die Funktion $\langle F90 \rangle$ in beliebiger Reihenfolge getestet werden. Die Tests werden alle durch den Kunden durchgeführt, welcher hierzu lediglich ein Skript erhält, das den Ablauf beschreibt. Dies ist eine Möglichkeit für den Kunden, selbst die Qualitätsanforderung $\langle Q20 \rangle$ und $\langle Q40 \rangle$ zu testen.

$\langle Q20 \rangle$und $\langle RM5 \rangle$ werden weiterhin durch eine Umfrage im Vorfeld getestet. Dabei werden 20 Personen befragt. Diese 20 Personen werden gebeten das Produkt zu testen und anschließend seine Bedienbarkeit und die Strukturiertheit der Benutzeroberfläche zu bewerten. Dazu wird ein Fragebogen erstellt.

$\langle Q10 \rangle$ wird ebenfalls durch eine Umfrage getestet. Hierbei werden wir 20 Personen bewusst falsche Eingaben tätigen lassen um dann zu erfragen, was die Fehlermeldungen, die sie bekommen bedeuten. Die Ergebnisse der Umfrage werden dem Kunden ebenfalls übergeben.

$\langle Q30 \rangle$ wird getestet, indem man die Webandwendung von PC's mit gängigen Betriebssystemen aufruft. Hierbei werden Windows 7, Windows 8, Debian Wheezy, sowie Mac OS 10.9 getestet.

Die Qualitätsanforderung $\langle Q50 \rangle$ wird vom Kunden beim Testen der verschiedenen Funktionen durchgeführt. Wird bei der Durchführung der Test keine erhöhte Antwortzeit festgestellt, so ist das Kriterium erfüllt.

2.5 Testumgebung

Während der Entwicklung wird die Software auf einem Testserver des IfIS laufen und möglichst früh über ein Buildsystem getestet werden. Zur Anwendung auf Serverseite kommen Unit-Test mit JUnit und/oder ScalaTest. Die Internetseite wird sehr einfach aufgebaut sein, deswegen wird hierfür eine manuelle Testung mithilfe eines Protokolls ausreichen. "Weiche" Kriterien wie etwa die intuitive Bedienung der Internetseite werden durch Benutzer-Umfragen überprüft. Für den Abnahmetest wird die Software auf einem Server des IfIS laufen.

3 Abnahmetest

Der Abnahmetest für *News Miner* dient dazu, die vom Kunden an den Auftraggeber im Pflichtenheft festgelegten Anforderungen an das Endprodukt auf ihre Vollständigkeit zu testen. Er soll sowohl die durch den Auftraggeber vorgestellten technischen Hintergrundprozesse sowie die eigentliche Bedienung des Produkts bewerten. Dies bedeutet, dass dem Kunden das vollständige Produkt mit all seinen Funktionen vorgestellt wird und er dieses mit seinen Vorstellungen abgleicht. Ziel ist es, dass der Kunde mit dem Endprodukt rund um zufrieden ist und keine weiteren Anmerkungen hat, sodass das Produkt frei gegeben werden kann und die Abnahme erfolgt.

3.1 Zu testende Anforderungen

Der Kunde soll im Verlauf des Abnahmetests alle Funktionen, die der Nutzer später verwendet, ebenfalls ausführen. Dies bedeutet, dass er einen gesamten Anmeldevorgang vornimmt, inklusive der Gesamtheit der Optionen, die auch der zukünftige Nutzer besitzt. Dies soll in der beispielhaften Reihenfolge in einem Durchgang geschehen:

- $\langle F10 \rangle$ Der Kunde registriert sich auf der Webplattform von *News Miner*.
- $\langle F20 \rangle$ Anmeldung auf der Webplattform des Produkts.
- $\langle F80 \rangle$ Nachrichtenquellen zum Nutzerkonto hinzufügen.
- $\langle F40 \rangle$ Der Kunde fügt präferierte Themengebiete zu seinem Nutzerkonto hinzu.
- $\langle F50 \rangle$ Der Kunde fügt Präferenzen zu seinem Nutzerkonto hinzu.
- $\langle F60 \rangle$ Der Kunde folgt einem aktuellen Trend
- $\langle F70 \rangle$ Das Folgen des Trends wird abgebrochen.
- $\langle F90 \rangle$ Nachrichtenquellen vom Nutzerkonto entfernen.
- $\langle F30 \rangle$ Abmeldung von der Webplattform des Produkts.

3.2 Testverfahren

Der Abnahmetest und die zugehörigen Tests sollen im Plenum mit dem Kunden und allen Auftraggebern erfolgen. Zunächst soll eine Vorstellung von *News Miner* durch den Auftragnehmer durchgeführt werden. Diese soll sämtliche Komponenten des Produkts enthalten. Es werden neben der für den Nutzer sichtbaren Webplattform auch die technischen Umsetzungen beschrieben und deren notwendige Wartung. Nachdem dem Kunden das gesamte Endprodukt erklärt wurde, werden weitere Mitarbeiter des Auftraggebers zur Unterstützung dazu gebeten. Denn nun soll der praktische Teil des Abnahmetests realisiert werden. Hierfür werden sämtliche Teilnehmer mit der Webplattform von *News Miner* in Kontakt gebracht.
Nun wird dem Kunden ein Testskript zur Hand gegeben, in welchem ihm kurz und strukturiert, die von ihm auszuführenden Arbeitsschritte erklärt werden. Zunächst soll der Kunde sich bei *News Miner* selbstständig registrieren $\langle F10 \rangle$. Nach der erfolgreichen Registrierung folgt das Anmelden $\langle F20 \rangle$. Nun soll der Kunde in seinem neu erstellten Nutzerkonto Nachrichtenquellen hinzufügen $\langle F80 \rangle$ sowie präferierte Themengebiete $\langle F40 \rangle$ und Themen $\langle F50 \rangle$. Nach erfolgreichem Abschluss kann er einem Trend folgen $\langle F60 \rangle$, diesen aber auch wieder löschen $\langle F70 \rangle$. Am Ende soll er versuchen, seine Nachrichtenquellen zu löschen $\langle F90 \rangle$ und sich im letzten Schritt abmelden $\langle F30 \rangle$.
Der gesamte Vorgang soll von dem Kunden selbstständig und ohne genauere Einführungen durchgeführt werden, um die einfache und strukturierte Bedienung zu testen. Zudem wird der Kunde gebeten, immer wieder fehlerhafte Eingaben zu machen, um auch die korrekte Ausführung der Fehlerbehandlungen zu testen. Nach dem Testende werden im Plenum die Erfahrungen mit dem Produkt besprochen und sämtliche Erkenntnisse protokolliert.

3.2.1 Testskripte

Zu Beginn des praktischen Teils des Abnahmetests wird dem Kunden eine Anleitung übergeben, in der ihm die zu testenden Funktion, deren Bedeutung und auszuführende Reihenfolge übergeben wird. Für den Abnahmetest von *News Miner* werden sonst keine Testskripte benötigt.

3.3 Testfälle

Testfall $\langle T100 \rangle$ - Nachrichten anzeigen

Ziel

Überprüfung, ob nach der Registrierung und Anmeldung die Nachrichten der jeweiligen angegebenen RSS-Feeds korrekt angezeigt werden.

Objekte/Methoden/Funktionen

$\langle F10 \rangle$ Registrieren, $\langle F20 \rangle$ Anmelden, $\langle F80 \rangle$ Nachrichtenquellen hinzufügen, $\langle UI10 \rangle$ Startseite, $\langle UI20 \rangle$ Hauptseite, $\langle UI30 \rangle$ Benutzerprofil

Pass/Fail Kriterien

Festellen, ob bei der angegebenen E-mail-Adresse eine Bestätigungsmail eingegangen ist und ob nach Auswahl des Links in der E-mail eine Weiterleitung zur Bestätigung erfolgt ist, ansonsten ist die Registrierung ungültig.

Verifikation, ob die Testperson nach der Anmeldung von der Startseite auf die Hauptseite gelangt, ansonsten ist die Anmeldung ungültig und es erscheint eine Fehlermeldung.

Überprüfen, ob Nachrichtenquellen korrekt hinzugefügt werden können, ansonsten muss eine Fehlermeldung ausgegeben werden.

Nachweisen, inwiefern die angezeigten Nachrichten aus den jeweils angegebenen RSS-Feeds stammen, ansonsten ist die Nachrichtenausgabe nicht korrekt.

Der Test ist erfolgreich, wenn alle Schritte korrekt waren.

Vorbedingung

Der Nutzer klickt auf der Startseite der Webanwendung den “create an account”-Button an, gibt die erforderten Daten an, erhält die Bestätigungsmail und stimmt der Registrierung durch Anklicken des Links zu. Anschließend meldet er sich an und geht auf das Nutzerprofil.

Einzelschritte

Eingabe:

1. Anklicken des “create an account”-Button auf der Startseite.
2. Feld unterhalb von “e-mail” anklicken.
3. Eingabe der E-mail-Adresse.
4. Feld unterhalb von “username” anklicken.
5. Eingabe des gewünschten Nutzernamen.
6. Feld unterhalb von “password” anklicken.
7. Eingabe des gewünschten Passwortes.
8. Mit Enter oder durch Klicken auf den “sign up”-Button bestätigen.

Ausgabe:

Ausgabe einer Bestätigung zur erfolgreichen Registrierung.

Eingabe:

1. Auf die Startseite der Webanwendung gehen.
2. Feld unterhalb von “e-mail” anklicken.

3. Eingabe der E-mail-Adresse.
4. Feld unterhalb von “password” anklicken.
5. Eingabe des gewünschten Passwortes.
6. Mit Enter oder durch Klicken auf den “sign in”-Button bestätigen.

Ausgabe:
Hauptseite der Webanwendung oder
bei nicht erfolgreichem Anmelden, Ausgabe einer Fehlermeldung.

Eingabe:

1. Auf das Nutzerprofil gehen.
2. Feld in der Kategorie “RSS-Feeds” anklicken.
3. Eingabe der URL von der jeweiligen Internetseite, von welcher der RSS-Feed abonniert werden soll.
4. Mit Enter oder durch Klicken auf den “confirm”-Button bestätigen.
5. Zur Hauptseite der Webanwendung wechseln.

Ausgabe:
Ausgabe von aufgrund von Trends gefilterten Nachrichten der jeweils angegebenen RSS-Feeds oder
bei bereits nicht erfolgreichem Abonnieren der RSS-Feeds, Ausgabe einer Fehlermeldung.

Beobachtungen / Log / Umgebung

Die Testperson erhält eine Bestätigungsmail. Fortan kann die Testperson sich anmelden, ohne sich vorher nochmal registrieren zu müssen.

Testfall $\langle T200 \rangle$ - Themen angeben

Ziel

Überprüfung, ob nach der Anmeldung die Testperson persönliche Präferenzen angeben kann und die angezeigten Nachrichten einen Bezug zu den jeweiligen Angaben haben, ansonsten sollen regulär nur allgemeine Nachrichten angezeigt werden.

Objekte/Methoden/Funktionen

$\langle F50 \rangle$ Präferenzen hinzufügen, $\langle F80 \rangle$ Nachrichtenquellen hinzufügen, $\langle UI20 \rangle$ Hauptseite, $\langle UI30 \rangle$ Benutzerprofil

Pass/Fail Kriterien

Falls die angezeigten Nachrichten keinen inhaltlichen Bezug zu den gewählten Präferenzen haben, so ist der Test nicht erfolgreich.

Vorbedingung

Die Testperson muss registriert sein, sich angemeldet haben und sich auf dem Nutzerprofil der Webanwendung befinden.

Einzelschritte

Eingabe:

1. Feld in der Kategorie "Favourite Subjects" anklicken.
2. Eingabe von persönlichen Präferenzen, also Themen, die die Testperson interessieren.

Ausgabe:

Ausgabe von Nachrichten der RSS-Feeds mit inhaltlichem Bezug zu den angegebenen Präferenzen auf der Hauptseite unter der Kategorie "Favourite Subjects" oder
bei Fehlschlag erfolgt nur die reguläre Ausgabe von allgemeinen Nachrichtenartikeln.

Abhängigkeiten

$\langle T100 \rangle$ Die Testperson muss erstmal mindestens einen RSS-Feed angeben, bevor daraus Nachrichten für die persönlichen Präferenzen ermittelt werden können.

Testfall $\langle T300 \rangle$ - Themengebiete auswählen

Ziel

Überprüfung, ob die Testperson vordefinierte Themengebiete angeben kann und die angezeigten Nachrichten einen Bezug zu den jeweiligen Angaben haben, ansonsten sollen regulär nur allgemeine Nachrichten angezeigt werden.

Objekte/Methoden/Funktionen

$\langle F40 \rangle$ Präferierte Themengebiete hinzufügen, $\langle F80 \rangle$ Nachrichtenquellen hinzufügen, $\langle UI20 \rangle$ Hauptseite, $\langle UI30 \rangle$ Benutzerprofil

Pass/Fail Kriterien

Wenn die angezeigten Nachrichten keinen inhaltlichen Bezug zu den gewählten Themengebieten haben, so ist der Test nicht erfolgreich.

Vorbedingung

Die Testperson muss registriert sein, sich angemeldet haben und sich auf dem Nutzerprofil der Webanwendung befinden.

Einzelschritte

Eingabe:

Felder von Themengebieten, die die Testperson interessieren, in der Kategorie “Favourite Topics” anklicken.

Ausgabe:

Ausgabe von Nachrichten der RSS-Feeds mit inhaltlichem Bezug zu den ausgewählten Themengebieten auf der Hauptseite unter der Kategorie “Favourite Topics” oder
bei Fehlschlag erfolgt nur die reguläre Ausgabe von allgemeinen Nachrichtenartikeln.

Abhängigkeiten

$\langle T100 \rangle$ Die Testperson muss erstmal mindestens einen RSS-Feed angeben, bevor daraus Nachrichten für die Themengebiete ermittelt werden können.

Testfall $\langle T400 \rangle$ - Trend folgen

Ziel

Überprüfung, ob die Testperson den Trends von den angezeigten Nachrichten folgen kann, ansonsten muss eine Fehlermeldung erscheinen.

Objekte/Methoden/Funktionen

$\langle F60 \rangle$ Trend folgen, $\langle F80 \rangle$ Nachrichtenquellen hinzufügen, $\langle UI20 \rangle$ Hauptseite, $\langle UI30 \rangle$ Benutzerprofil

Pass/Fail Kriterien

Falls der Trend im Nutzerprofil unter der Kategorie “Trends I follow” aufgelistet wird oder der “follow trend”-Button beim Nachrichtenartikel verschwunden ist, folgt die Testperson dem Trend und der Test ist erfolgreich.

Vorbedingung

Die Testperson muss registriert sein, sich angemeldet haben und sich auf der Hauptseite befinden sowie einen Nachrichtenartikel ausgewählt haben, sodass der vollständige Artikel erscheint.

Einzelschritte

Eingabe:

1. Auswählen eines Nachrichtenartikels.
2. Anklicken des “follow trend”-Buttons.
3. Zum Nutzerprofil wechseln.

Ausgabe:

Anzeige der Trends, denen die Testperson folgt oder
bei Fehlschlag Ausgabe einer Fehlermeldung.

Abhängigkeiten

⟨$T100$⟩ Die Testperson muss erstmal mindestens einen RSS-Feed angeben, bevor daraus Nachrichten angezeigt werden können, denen man Folgen kann.

Testfall ⟨$T500$⟩ - Trend nicht mehr folgen

Ziel

Überprüfung, ob die Testperson den Trends, denen sie folgt, auch nicht mehr folgen kann, ansonsten muss eine Fehlermeldung erscheinen.

Objekte/Methoden/Funktionen

⟨$F70$⟩ Trend nicht mehr folgen, ⟨$F80$⟩ Nachrichtenquellen hinzufügen, ⟨$UI20$⟩ Hauptseite, ⟨$UI30$⟩ Benutzerprofil

Pass/Fail Kriterien

Wenn der Trend im Nutzerprofil unter der Kategorie "Trends I follow" nicht mehr aufgelistet wird oder der "don't follow"-Button beim Nachrichtenartikel nicht mehr angezeigt wird, folgt die Testperson dem Trend nicht mehr und der Test ist erfolgreich.

Vorbedingung

Die Testperson muss registriert sein, sich angemeldet haben sowie mindestens einem Trend folgen und sich auf dem Nutzerprofil der Webanwendung befinden.

Einzelschritte

Eingabe:

1. Auf das Nutzerprofil gehen.
2. Anklicken des "don't follow"-Buttons von dem Trend, dem die Testperson nicht länger folgen möchte.

Ausgabe:

Anzeige der übrigen Trends, denen die Testperson folgt oder
bei Fehlschlag Ausgabe einer Fehlermeldung.

Abhängigkeiten

⟨$T100$⟩, ⟨$T500$⟩ Die Testperson muss erstmal mindestens einen RSS-Feed angeben, bevor daraus Nachrichten angezeigt werden können, denen man Folgen kann.

Testfall ⟨$T600$⟩ - Nachrichtenquellen löschen

Ziel

Überprüfung, ob die Testperson Nachrichtenquellen löschen kann, ansonsten muss eine Fehlermeldung erscheinen.

Objekte/Methoden/Funktionen

$\langle F90 \rangle$ Nachrichtenquellen löschen, $\langle UI30 \rangle$ Benutzerprofil

Pass/Fail Kriterien

Wenn die Nachrichtenquelle nicht mehr im Nutzerprofil angezeigt wird, so ist der Test erfolgreich.

Vorbedingung

Die Testperson muss registriert sein, sich angemeldet haben sowie mindestens einen RSS-Feed angegeben haben und sich auf dem Nutzerprofil von *News Miner* befinden.

Einzelschritte

Eingabe:

1. Auf das Nutzerprofil gehen.
2. Löschen der Nachrichtenquelle aus dem Feld.
3. Mit Enter oder durch Klicken auf den "confirm"-Button bestätigen.

Ausgabe:

Das Feld, aus welchem die Nachrichtenquelle gelöscht wurde, ist nun leer oder bei Fehlschlag Ausgabe einer Fehlermeldung.

Abhängigkeiten

$\langle T100 \rangle$ Die Testperson muss erstmal RSS-Feeds angeben haben, bevor RSS-Feeds gelöscht werden können.

Testfall $\langle T700 \rangle$ - Nutzer abmelden

Ziel

Überprüfung, ob sich die angemeldete Testperson abmelden kann, ansonsten muss eine Fehlermeldung ausgegeben werden.

Objekte/Methoden/Funktionen

$\langle F30 \rangle$ Abmelden, $\langle UI10 \rangle$ Startseite, $\langle UI20 \rangle$ Hauptseite

Pass/Fail Kriterien

Der Testfall ist erfolgreich, wenn die Testperson nach dem Drücken auf den "log out"-Button wieder auf die Startseite gelangt.

Vorbedingung

Die Testperson muss sich registriert und angemeldet haben.

Einzelschritte

Eingabe:

Anklicken des "log out"-Buttons.

Ausgabe:

Startseite der Webanwendung *News Miner* oder

bei nicht erfolgreichem Abmelden, Ausgabe einer Fehlermeldung.

Abhängigkeiten

⟨$T100$⟩ Die Testperson muss registriert und angemeldet sein, um sich abmelden zu können.

4 Integrationstest

Die Integrationstests sollen sicherstellen, dass die einzelnen Komponenten des *NewsMiner* im Zusammenspiel funktionieren.

4.1 Zu testende Komponenten

- *TwitterCrawler*
- *TrendExtractor*
- *NewsAggregator*
- *RSSGrabber*

4.2 Testverfahren

Die Kompontenten wurden inkrementell integriert, weshalb auch beim Testen nach dieser Strategie vorgegangen wird. Das heißt also, dass immer zunächst das Zusammenspiel von 2 Komponenten getestet wird, um anschließend inkrementell weitere Komponenten hinzuzufügen, um diese zu testen.

Dafür wurde das *System* zunächst integriert und folgt dabei folgender Struktur: der *TwitterCrawler* ruft nachdem er eine Tweetdatei erstellt hat den *TrendExtractor* auf. Dieser berechnet Trends auf der Grundlage dieser Datei und ruft den *NewsAggregator* auf, welcher zunächst den *RSSGrabber* aufruft. Wenn der *RSSGrabber* seine Arbeit beendet hat, arbeitet der *NewsAggregator*. Während dieses Prozesses, werden verschiedene benötigte Daten in die Datenbank geschrieben. Diese Datenbank stellt die Schittstelle zwischen *NewsMinerGUI* und den Systemkomponenten dar.

Daraus ergeben sich die unten beschriebenen Testfälle auf dem Gesamtsystem.

4.2.1 Testskripte

Es wurden keine Testskripte verwendet.

4.3 Testfälle

Testfall $\langle T800 \rangle$ - *TwitterCrawler* + *TrendExtractor*

Ziel

Das Ziel dieses Tests ist es abzusichern, dass die Komponenten *TwitterCrawler* und *TrendExtractor* korrekt zusammenarbeiten

Objekte/Methoden/Funktionen

`findEmergentTopics()`, `extractTags()`

Pass/Fail Kriterien

Pass: Es erfolgt keine Ausgabe, oder eine Ausgabe aufgrund einer abgefangenen Exception
Fail: Es erfolgt die Ausgabe einer Java-Fehlermeldung.

Vorbedingung

keine

Einzelschritte

- Den *TwitterCrawler* starten.
- Den *TwitterCrawler* nach 2 Tagen stoppen.
- Ausgaben überprüfen

Beobachtungen / Log / Umgebung

keine

Besonderheiten

In zwei Tagen erstellt der *TwitterCrawler* 24 Tweetdateien und und ruft genau so oft den *TrendExtractor* auf. Wenn irgendwelche Ausgaben zu sehen sind, ist der Test fehlgeschlagen.

Abhängigkeiten

keine

Testfall $\langle T900 \rangle$ - *TrendExtractor* + *NewsAggregator*

Ziel

Überprüfung der korrekten Integration von *TrendExtractor* und *NewsAggregator*

Objekte/Methoden/Funktionen

`findEmergentTopics()`, `filterArticles()`

Pass/Fail Kriterien

Es treten keine Fehlermeldungen auf

Vorbedingung

Der *TwitterCrawler* und der *TrendExtractor* arbeiten korrekt zusammen.

Einzelschritte

Führe den TwitterCrawler aus. Dieser ruft den *TrendExtractor* auf, welcher den *NewsAggregator* aufruft.

Beobachtungen / Log / Umgebung

keine

Besonderheiten

keine

Abhängigkeiten

TwitterCrawler

Testfall ⟨*T*1000⟩ - *NewsAggregator* + *RSSGrabber*

Ziel

Überprüfe ob der *NewsAggregator* und *RSSGrabber* korrekt zusammen funktionieren

Objekte/Methoden/Funktionen

`filterArticles()`, `getFeeds()`

Pass/Fail Kriterien

Der NewsAggegrator arbeitet korrekt

Vorbedingung

In der Datenbank müssen Testnutzer und zugehörige Feeds vorhanden sein.

Einzelschritte

Führe den *NewsAggregator* aus, dieser ruft die Methode `getFeeds()` des *RSSGrabbers* auf.

Beobachtungen / Log / Umgebung

Der RSSGrabber erstellt Dateien und schreibt deren Dateinamen in die Datenbank. Hierauf greift der NewsAggregator zu.

Besonderheiten

keine

Abhängigkeiten

NewsMinerStorage

5 Unit-Tests

Die Unit-Tests haben das Ziel, jede einzelne Komponente auf Fehleranfälligkeit zu testen. Hierbei wird das Augenmerk insbesondere auf Methoden gerichtet, die eine Eingabe von außen bekommen. Damit ist also gemeint, dass die Eingabe nicht aus unserem System stammt. Dafür wird die Komponente für sich auf fehlerhafte Eingaben, mit denen zu rechnen ist, getestet. Das sind also solche Eingaben, die durch den Nutzer ins System gelangen bzw. im Twitterstream oder einem RSS-Feed ihren Ursprung haben. Deshalb werden die Methoden, die solche Eingaben behandeln besonders ausführlich getestet.

Nicht getestet werden vor allem die Methoden, deren semantische Korrektheit nicht testbar ist. Das sind also insbesondere die Methoden zur Trendextraktion aus dem *TrendExtractor*, aber auch die Methoden aus dem *NewsAggregator*, die Trends mit News vergleichen.

5.1 Zu testende Komponenten

- *TwitterCrawler*
- *TrendExtractor*
- *NewsAggregator*
- *RSSGrabber*

5.2 Testverfahren

Der Test der Komponente *TwitterCrawler* wird mittels eines JUnit Tests durchgeführt. Hier wird nur getestet, ob das Programm den TwitterStream mitliest.

Die Komponente *TrendExtractor* wird durch ein Testskript sowie durch die Ausführung getestet.

5.2.1 Testskripte

`TrendExtractorTest.java`

5.3 Testfälle

5.3.1 TwitterCrawler

Testfall $\langle T1100 \rangle$ **- Klasse run()**

Ziel

Überprüfung auf Funktion der Komponente

Objekte/Methoden/Funktionen

In diesem Test wird die Methode `run()` auf Funktionalität überprüft. Dazu wird ein neues Objekt vom Typ *TwitterCrawler* erstellt.

Pass/Fail Kriterien

Der Test war erfolgreich, wenn keine Fehlermeldung erscheint.

Vorbedingung

Es besteht eine Verbindung zum Internet.

Einzelschritte

Dieser Test kann mithilfe von 'sbt', ein Simple Build Tool für Java und Scala Projekte, gestartet und ausgeführt werden. Dazu muss man in den Ordner der Anwendung /emph-NewsMiner wechseln und 'sbt' starten. Anschließend führt man dort den Test mittels dem Aufruf 'run' auf.

Beobachtungen / Log / Umgebung Es wird keine spezielle Umgebung benötigt. Jedoch muss 'sbt' auf dem System vorhanden sein.

Testfall $\langle T1200 \rangle$ **- Klasse run()**

Ziel

Erneute Überprüfung auf Funktion der Komponente

Objekte/Methoden/Funktionen

In diesem Test wird die Methode `run()` auf Funktionalität überprüft. Dazu wird ein neues Objekt vom Typ *TwitterCrawler* erstellt.

Pass/Fail Kriterien

Test war erfolgreich, wenn keine Fehlermeldung erscheint.

Vorbedingung

Es besteht eine Verbindung zum Internet.

Einzelschritte

Dieser Test kann mithilfe von 'sbt' gestartet und ausgeführt werden. Dazu muss man in den Ordner der Anwendung /emphNewsMiner wechseln und 'sbt' starten. Anschließend führt man dort den Test durch den Aufruf 'run' auf.

Beobachtungen / Log / Umgebung Es wird keine spezielle Umgebung benötigt. Jedoch muss 'sbt' auf dem System vorhanden sein.

5.3.2 TrendExtractor

Testfall $\langle T1300 \rangle$ **-** ***TrendExtractor***

Ziel

Funktionsfähigkeit des *TrendExtractor*s sicherstellen

Objekte/Methoden/Funktionen

`countIntersection()`

Pass/Fail Kriterien

Wenn die erwartete Anzahl der Tweets und die berechnete übereinstimmt, ist der Test erfolgreich verlaufen.

Einzelschritte

- Erstelle Testdaten.
- Berechne die Anzahl der Tweets im Durchschnitt der Testdaten.
- Vergleiche diese mit der erwarteten Anzahl.

Beobachtungen / Log / Umgebung

NetBeans

Testfall $\langle T1400 \rangle$ **-** ***TrendExtractor***

Ziel

Funktionsfähigkeit des *TrendExtractor*s sicherstellen

Objekte/Methoden/Funktionen

`countUnion()`

Pass/Fail Kriterien

Wenn die erwartete Anzahl der Tweets und die berechnete übereinstimmt, ist der Test erfolgreich verlaufen.

Einzelschritte

- Erstelle Testdaten.

- Berechne die Anzahl der Tweets in der Vereinigung der Testdaten.
- Vergleiche diese mit der erwarteten Anzahl.

Beobachtungen / Log / Umgebung

NetBeans

Testfall $\langle T1500 \rangle$ - *TrendExtractor*

Ziel

Funktionsfähigkeit des *TrendExtractor*s sicherstellen

Objekte/Methoden/Funktionen

`getCorrelation()`

Pass/Fail Kriterien

Wenn die erwartete Korrelation und die berechnete übereinstimmt, ist der Test erfolgreich verlaufen.

Vorbedingung

Der *TrendExtractor* muss zuvor Tags aus einer Tweetdatei extrahiert haben, damit die Variable hashTagNumber einen Wert > 0 hat.

Einzelschritte

- Erstelle Testdaten.
- Berechne den Korrelationswert.
- Vergleiche diese mit der erwarteten Anzahl.

Beobachtungen / Log / Umgebung

NetBeans

Besonderheiten

Dieser Wert ist Abhängig von der Anzahl der Tweets im momentanen Zeitfenster und kann deshalb nur im Zusammenhang mit einer Tweetdatei ausgeführt werden

Abhängigkeiten

`extractTags()`

Testfall $\langle T1600 \rangle$ - *TrendExtractor*

Ziel

Fehleranfälligkeit des *TrendExtractor*s minimieren

Objekte/Methoden/Funktionen

`extractTags()`

Pass/Fail Kriterien

Es wird eine Fehlermeldung ausgegeben und die zurückgegeben ArrayList ist leer oder es wird keine Fehlermeldung ausgegeben und zurückgegebene ArrayList enthält mindestens ein Element.

Vorbedingung Die Datei aus der die Tags zu extrahieren sind, muss an der richtigen Stelle liegen

Einzelschritte

- Rufe extractTags(Dateiname) auf.
- Gebe den Rückgabewert aus.
- Überprüfe die Ausgabe.

Beobachtungen / Log / Umgebung

NetBeans

Besonderheiten

Hier wird nur getestet ob im richtigen Fall Fehlermeldungen ausgegeben werden.

Abhängigkeiten

`extractTags()`, *TwitterCrawler*

Testfall ⟨*T*1700⟩ - *TrendExtractor*

Ziel

Fehleranfälligkeit des *TrendExtractor*s minimieren

Objekte/Methoden/Funktionen

`findEmergentTopics()`, `readHistory()`, `saveHistory()`, `saveTrend()`, `correlatedTags()`, `getCorrelation()`, `countUnion()`, `countIntersection()`, `findSeedTags()`, `seedTag()`, `sortList()`, `ignoreTags()`, `extractTags()`,`scorePairs()`, `predict()`

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn keine Fehlermeldung auftritt.

Vorbedingung

Die Dateien aus der die Tags zu extrahieren sind, muss an der richtigen Stelle liegen

Einzelschritte

- Führe den *TrendExtractor* mit 50 verschiedenen Tweetdateien aus.

Beobachtungen / Log / Umgebung

NetBeans

Abhängigkeiten

TwitterCrawler

Testfall $\langle T1800 \rangle$ - *TrendExtractor*

Ziel

Funktionsfähigkeit des *TrendExtractor*s sicherstellen

Objekte/Methoden/Funktionen

`ignoreTags()`

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn die Liste hinterher nur Elemente enthält, deren Popularität größer ist als der eingegeben Parameter

Einzelschritte

- Erstelle Testdaten.
- Führe Methode auf Testdaten aus.
- Überprüfe das Ergebnis auf fehlerhafte Werte.

Beobachtungen / Log / Umgebung

NetBeans

Testfall $\langle T1900 \rangle$ - *TrendExtractor*

Ziel

Funktionsfähigkeit des *TrendExtractor*s sicherstellen

Objekte/Methoden/Funktionen

`seedTag()`

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn die Liste hinterher nur so viele Elemente enthält, wie es der eingegebene Parameter angibt.

Einzelschritte

- Erstelle Testdaten.
- Führe Methode auf Testdaten aus.
- Überprüfe die Anzahl der Elemente im Ergebnis.

Beobachtungen / Log / Umgebung

NetBeans

5.3.3 NewsAggregator

Testfall $\langle T2000 \rangle$ **-** ***NewsAggregator***

Ziel

Korrekte Implementierung und Funktion des NewsAggregators prüfen

Objekte/Methoden/Funktionen

`filterArticles()`

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn keine Exceptions geworfen werden, bzw. diese abgefangen werden und nach Ausführung Daten in der Tabelle 'ArticleBelongsToTrend' eingetragen sind.

Einzelschritte

- Erstelle Testdaten bestehend aus Trends und Nachrichtenartikeln.
- Führe Methode auf Testdaten aus, speichere dabei die Ausgabe.
- Überprüfe die Ausgabe und das Vorhandensein der Daten in der genannten Tabelle.

Beobachtungen / Log / Umgebung

keine

Besonderheiten

Die semantische Korrektheit ist ein weiches Kriterium, weswegen innerhalb dieses Tests lediglich richtig Implementierung getestet werden kann.

5.3.4 RSSGrabber

Testfall $\langle T2100 \rangle$ **-** ***RSSGrabber***

Ziel

Überprüfen, ob fehlerfreie RSS-Feeds in der Datenbank abgespeichert werden können.

Objekte/Methoden/Funktionen

`testCreateDocument()`

Pass/Fail Kriterien

Es wird keine Fehlermeldung ausgegeben.

Vorbedingung

Keiner der RSS-Feeds ist bereits in der Datenbank gespeichert.

Einzelschritte

- Ausführen der Methode.

- Vergewisserung, dass keine Fehlermeldungen geworfen wurden.

Beobachtungen / Log / Umgebung

Eclipse SDK

Abhängigkeiten

`nameFinder()`, `saveArticle()`

Testfall $\langle T2200 \rangle$ - *RSSGrabber*

Ziel

Sicherstellen, dass keine RSS-Feeds doppelt in der Datenbank abgespeichert werden.

Objekte/Methoden/Funktionen

`test2CreateDocument()`

Pass/Fail Kriterien

Es wird eine SQLException geworfen.

Einzelschritte

- Ausführen der Methode.
- Überprüfung, dass die Fehlermeldung geworfen wurde.

Beobachtungen / Log / Umgebung

Eclipse SDK

Abhängigkeiten

`nameFinder()`, `saveArticle()`

Testfall $\langle T2300 \rangle$ - *RSSGrabber*

Ziel

Vergewissern, dass keine fehlerhaften RSS-Feeds in der Datenbank abgespeichert werden.

Objekte/Methoden/Funktionen

`test3CreateDocument()`

Pass/Fail Kriterien

Es wird die Fehlermeldung "Test: A wrong URL has been detected." und eine NullPointerException mit einem Fehlertext, der eine falsche URL enthält, ausgegeben.

Einzelschritte

- Ausführen der Methode.
- Sicherstellung, dass die Fehlermeldungen geworfen wurden.

Beobachtungen / Log / Umgebung

Eclipse SDK

Abhängigkeiten

`nameFinder()`, `saveArticle()`

Testfall $\langle T2400 \rangle$ - *RSSGrabber*

Ziel

Überprüfung, ob die Sonderzeichen aus den RSS-Feeds entfernt wurden.

Objekte/Methoden/Funktionen

`testNameFinder()`

Pass/Fail Kriterien

Es wird keine Fehlermeldung ausgegeben.

Einzelschritte

- Ausführen der Methode.
- Vergewisserung, dass keine Fehlermeldungen geworfen wurden.

Beobachtungen / Log / Umgebung

Eclipse SDK

Testfall $\langle T2500 \rangle$ - *RSSGrabber*

Ziel

Feststellen, ob die Nachrichtenartikel in der Datenbank abgespeichert werden.

Objekte/Methoden/Funktionen

`testSaveArticle()`

Pass/Fail Kriterien

Es wird keine Fehlermeldung geworfen.

Einzelschritte

- Ausführen der Methode.
- Sicherstellen, dass keine Fehlermeldungen geworfen wurden.

Beobachtungen / Log / Umgebung

Eclipse SDK

Testfall $\langle T2600 \rangle$ - *RSSGrabber*

Ziel

Vergewissern, ob die URLs der RSS-Feeds aus der Datenbank abgerufen werden können.

Objekte/Methoden/Funktionen

`testGetFeeds()`

Pass/Fail Kriterien

Es wird keine Fehlermeldung geworfen.

Vorbedingung

Es befindet sich mindestens eine URL in der Datenbank.

Einzelschritte

- Ausführen der Methode.
- Feststellen, dass keine Fehlermeldungen geworfen wurden.

Beobachtungen / Log / Umgebung

Eclipse SDK

Abhängigkeiten

`createDocument()`