



NEWS MINER

Gruppe 1

Software-Entwicklungspraktikum (SEP)
Sommersemester 2013

Testprotokolle

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

Inhaltsverzeichnis

| | |
|--|-----------|
| 1 Testdurchführung (2013-06-02) | 4 |
| 1.1 Testumgebung | 4 |
| 1.2 Testprotokoll | 4 |
| 1.3 Zusammenfassung | 5 |
| 2 Testdurchführung (2013-06-04) | 6 |
| 2.1 Testumgebung | 6 |
| 2.2 Testprotokoll | 6 |
| 2.3 Zusammenfassung | 6 |
| 3 Testdurchführung (2013-06-14-001) | 7 |
| 3.1 Testumgebung | 7 |
| 3.2 Testprotokoll | 7 |
| 3.3 Zusammenfassung | 8 |
| 4 Testdurchführung (2013-06-14-002) | 9 |
| 4.1 Testprotokoll | 9 |
| 4.2 Zusammenfassung | 10 |
| 5 Testdurchführung (2013-06-15-001) | 11 |
| 5.1 Testumgebung | 11 |
| 5.2 Testprotokoll | 11 |
| 5.3 Zusammenfassung | 11 |
| 6 Testdurchführung (2013-06-15-002) | 12 |
| 6.1 Testprotokoll | 12 |
| 6.2 Zusammenfassung | 12 |
| 7 Testdurchführung (2013-06-17) | 14 |
| 7.1 Testumgebung | 14 |
| 7.2 Testprotokoll | 14 |
| 7.3 Zusammenfassung | 14 |

| | |
|---|-----------|
| 8 Testdurchführung (2013-06-22) | 16 |
| 8.1 Testumgebung | 16 |
| 8.2 Testprotokoll | 16 |
| 8.3 Zusammenfassung | 16 |
| 9 Testdurchführung (2013-06-23-001) | 17 |
| 9.1 Testumgebung | 17 |
| 9.2 Testprotokoll | 17 |
| 10 Testdurchführung (2013-06-23-002) | 18 |
| 10.1 Testprotokoll | 18 |
| 10.2 Zusammenfassung | 18 |
| 11 Testdurchführung (2013-07-01) | 19 |
| 11.1 Testumgebung | 19 |
| 11.2 Testprotokoll | 19 |
| 11.3 Zusammenfassung | 19 |
| 12 Testdurchführung (2013-07-05) | 20 |
| 12.1 Testumgebung | 20 |
| 12.2 Testprotokoll | 20 |
| 12.3 Zusammenfassung | 20 |
| 13 Testdurchführung (2013-07-09) | 21 |
| 13.1 Testumgebung | 21 |
| 13.2 Testprotokoll | 21 |
| 13.3 Zusammenfassung | 23 |

1 Testdurchführung (2013-06-02)

Im folgenden wird der Test der Komponente *TwitterCrawler* protokolliert.

Art des Tests: UnitTest

Ausgeführte Testfälle: $\langle T1100 \rangle$ $\langle T1200 \rangle$

Beteiligte Tester: Viviane Werner, Jana Riquel

Abgedeckte Funktionen: Twitterstream lesen $\langle RM2 \rangle$

1.1 Testumgebung

Die Testfälle wurden unter Windows 7 mit sbt durchgeführt. Dabei wurde eine Testdatei verwendet.

1.2 Testprotokoll

| | |
|------------------------------------|--|
| Testfall | $\langle T1100 \rangle$ |
| Tester | <i>Viviane Werner, Jana Riquel</i> |
| Eingaben | <i>Keine</i> |
| Soll - Reaktion | <i>Der Test wird ohne Fehler ausgeführt.</i> |
| Ist – Reaktion | <i>Der Test wird ohne Fehler ausgeführt.</i> |
| Ergebnis | <i>Der Test ist erfolgreich.</i> |
| Unvorhergesehene Ereignisse | <i>Keine</i> |
| Nacharbeiten | <i>Keine</i> |

| | |
|------------------------|--|
| Testfall | $\langle T1200 \rangle$ |
| Tester | <i>Viviane Werner, Jana Riquel</i> |
| Eingaben | <i>Keine</i> |
| Soll - Reaktion | <i>Der Test wird ohne Fehler ausgeführt.</i> |
| Ist – Reaktion | <i>Der Test wird ohne Fehler ausgeführt.</i> |

| | |
|------------------------------------|----------------------------------|
| Ergebnis | <i>Der Test ist erfolgreich.</i> |
| Unvorhergesehene Ereignisse | <i>Keine</i> |
| Nacharbeiten | <i>Keine</i> |

1.3 Zusammenfassung

Die beiden Testläufe haben gezeigt, dass die getesteten Methoden korrekt funktionieren. Möglichkeiten Fehler abzufangen, die entstehen könnten, falls die Struktur des Twitterstreams geändert wird, werden von diesem Test jedoch nicht abgedeckt.

2 Testdurchführung (2013-06-04)

Art des Tests: UnitTest

Ausgeführte Testfälle: $\langle T2100 \rangle$, $\langle T2200 \rangle$, $\langle T2300 \rangle$, $\langle T2400 \rangle$, $\langle T2500 \rangle$

Beteiligte Tester: Jennifer Sieg

Getestete Funktionalitäten: RSS-Feeds abspeichern

2.1 Testumgebung

Die Testfälle wurden unter Windows XP mit Eclipse SDK (Version: 4.2.1) durchgeführt.

2.2 Testprotokoll

| | |
|------------------------|--|
| Testfall | $\langle T2100 \rangle$, $\langle T2200 \rangle$, $\langle T2300 \rangle$, $\langle T2400 \rangle$, $\langle T2500 \rangle$ |
| Tester | <i>Jennifer Sieg</i> |
| Eingaben | <i>Keine</i> |
| Soll - Reaktion | <i>Es wird die Fehlermeldung "Test: A wrong URL has been detected." und ggf. eine NullPointerException mit einem Fehlertext, der eine falsche URL enthält, ausgegeben.</i> |
| Ist – Reaktion | <i>Es werden nur die oben beschriebenen Fehlermeldungen ausgegeben, ansonsten keine weiteren mehr.</i> |
| Ergebnis | <i>Der Test ist erfolgreich.</i> |
| Nacharbeiten | <i>Keine</i> |

2.3 Zusammenfassung

Dieser Test hat aufgezeigt, dass die Methode `nameFinder()` problemlos die Sonderzeichen rausfiltern kann und die Methode `createDocument()` ein Document erstellt, in welchem von einem Nachrichtenartikel jeweils der Titel, eine Kurzbeschreibung, der Link sowie der Link des Ursprungsfeeds enthalten ist. Zudem wurde getestet, dass der Artikel in der Datenbank erfolgreich abgespeichert wird.

3 Testdurchführung (2013-06-14-001)

Art des Tests: UnitTest

Ausgeführte Testfälle: $\langle T1300 \rangle$, $\langle T1400 \rangle$, $\langle T1500 \rangle$, $\langle T1600 \rangle$

Beteiligte Tester: Maria Joanna Born

Abgedeckte Funktionen: Trends extrahieren $\langle RM4 \rangle$

3.1 Testumgebung

Die Testfälle wurden unter Windows 7 mit NetBeans durchgeführt. Es wurde ein Testskript, sowie einige Testdateien verwendet.

3.2 Testprotokoll

| | |
|------------------------|--|
| Testfall | $\langle T1100 \rangle$, $\langle T1200 \rangle$, $\langle T1300 \rangle$, $\langle T1400 \rangle$ |
| Tester | <i>Maria Joanna Born</i> |
| Eingaben | <i>Zwei Dateien : TestTweet und RightTweet</i> |
| Soll - Reaktion | <i>Internal Error: Tweetfile not found. extractTags erfolgreich Internal Error: Tweetfile not found. extractTags erfolgreich extractTagsTest erfolgreich countIntersectionTest erfolgreich countUnionTest erfolgreich getCorrelationTest erfolgreich countIntersectionTest erfolgreich countUnionTest erfolgreich getCorrelationTest erfolgreich</i> |

| | |
|------------------------------------|---|
| Ist – Reaktion | <i>Internal Error: Tweetfile not found. extractTags erfolgreich Internal Error: Tweetfile not found. extractTags erfolgreich Internal Error: Tweetfile not found. extractTagsTest Fehlgeschlagen countIntersectionTest erfolgreich countUnionTest Fehlgeschlagen: 2 != 1 getCorrelationTest Fehlgeschlagen: 0.0 != NaN countIntersectionTest erfolgreich countUnionTest erfolgreich getCorrelationTest Fehlgeschlagen: 1.0 != Infinity</i> |
| Ergebnis | <i>Der Test ist missglückt. Die Gründe sind zum einen, dass der Tester vergessen hat, die beiden benötigten Dateien in den korrekten Ordner zu legen (Das betrifft die Tweetfiles). Außerdem scheint es einen Fehler in den Methoden countUnion und getCorrelation zu geben, der durch diesen Test aufgedeckt wurde.</i> |
| Unvorhergesehene Ereignisse | <i>Fehl Schlag der Tests für countUnion() und getCorrelation()</i> |
| Nacharbeiten | <i>Ein Ausrufezeichen in Zeile 276 des TrendExtractors, hat dazu geführt, dass die Anzahl der Elemente in der Vereinigung zweier Tag-Tupel falsch berechnet wurde. Das Ausrufezeichen wurde entfernt. Außerdem wurde der Vergleich in der selben Zeile versehentlich auf den TweetsAndHashTags-Objekten durchgeführt. Da es aber nur um die Anzahl der Tweets geht, war das auch falsch. Auch der Fehler wurde korrigiert indem tweetA.equals(tweetB) durch tweetA.getTweet().equals(tweetB.getTweet()) ersetzt wurde. Das hat dann auch den ersten Fehler in der getCorrelation()-Methode korrigiert. Der zweite Fehler lag nicht im TrendExtractor, sondern im Aufbau des Tests. Die Korrelation von zwei TagTupeln bezieht außer der Vereinigung und dem Durchschnitt aller Tweets auch noch die Anzahl aller HashTags mit ein. Diese ist 0, wenn man vorher keine HashTags extrahiert hat. Da bei der Berechnung durch diesen Wert geteilt wird, macht Java daraus ein Infinity. Außerdem ist das Ergebnis dadurch falsch eingeschätzt worden. Das erwartete Ergebnis wurde von 1.0 zu 0.007751937984496124 geändert.</i> |

3.3 Zusammenfassung

Bei diesem Testlauf ist ein Fehler im Test aufgefallen und wurde behoben. Außerdem wurden durch diesen Testlauf Fehler in der Methode `countUnion` aufgedeckt, welche ebenfalls behoben wurden. Der Testlauf testet den vollen Umfang der Methoden `countUnion` und `countIntersection`

4 Testdurchführung (2013-06-14-002)

4.1 Testprotokoll

| | |
|------------------------------------|--|
| Testfall | $\langle T1200 \rangle, \langle T1400 \rangle, \langle T1500 \rangle, \langle T1600 \rangle$ <i>countUnionTest</i> |
| Tester | <i>Maria Joanna Born</i> |
| Eingaben | <i>Zwei Dateien : TestTweet und RightTweet</i> |
| Soll - Reaktion | <i>Internal Error: Tweetfile not found. extractTags erfolgreich Internal Error: Tweetfile not found. extractTags erfolgreich extractTagsTest erfolgreich countIntersectionTest erfolgreich countUnionTest erfolgreich getCorrelationTest erfolgreich countIntersectionTest erfolgreich countUnionTest erfolgreich getCorrelationTest erfolgreich</i> |
| Ist – Reaktion | <i>Internal Error: Tweetfile not found. extractTags erfolgreich Internal Error: Tweetfile not found. extractTags erfolgreich extractTagsTest erfolgreich countIntersectionTest erfolgreich countUnionTest erfolgreich getCorrelationTest erfolgreich countIntersectionTest erfolgreich countUnionTest erfolgreich getCorrelationTest erfolgreich</i> |
| Ergebnis | <i>Der Test war erfolgreich</i> |
| Unvorhergesehene Ereignisse | <i>Keine</i> |
| Nacharbeiten | <i>Es wurden keine Nacharbeiten durchgeführt</i> |

4.2 Zusammenfassung

Der Testlauf hat gezeigt, dass die getesteten Methoden korrekt funktionieren.

5 Testdurchführung (2013-06-15-001)

Art des Tests: UnitTest

Ausgeführte Testfälle: $\langle T1700 \rangle$

Beteiligte Tester: Maria Joanna Born

Abgedeckte Funktionen: Trends extrahieren $\langle RM3 \rangle$

5.1 Testumgebung

Die Testfälle wurden unter Windows 7 mit NetBeans durchgeführt. Es wurden außerdem 50 Testdateien verwendet.

5.2 Testprotokoll

| | |
|------------------------------------|--|
| Testfall | $\langle T1700 \rangle$ |
| Tester | <i>Maria Joanna Born</i> |
| Eingaben | <i>50 Dateien: tweets124 - tweets174</i> |
| Soll - Reaktion | <i>Es gibt keine Ausgabe und keine Fehlermeldungen</i> |
| Ist – Reaktion | <i>Es tritt eine <code>StringIndexOutOfBoundsException</code> in der Methode <code>readHistory</code> auf.</i> |
| Ergebnis | <i>Der Test ist fehlgeschlagen</i> |
| Unvorhergesehene Ereignisse | <i>Es ist eine Fehlermeldung aufgetreten und in der Historydatei selbst waren exorbitant viele Kommata</i> |
| Nacharbeiten | <i>Die Fehlermeldung trat aus folgendem Grund auf: Es ist möglich, dass Tweets mit nur einem Hashtag in dieser Datei aufgelistet sind. Dieser Fall wurde von den Entwicklern nicht bedacht. Es wurde also eine weitere Fallunterscheidung eingefügt.</i> |

5.3 Zusammenfassung

Es wurden zwei Fehler aufgedeckt und repariert.

6 Testdurchführung (2013-06-15-002)

6.1 Testprotokoll

| | |
|------------------------------------|---|
| Testfall | <i>⟨T1700⟩</i> |
| Tester | <i>Maria Joanna Born</i> |
| Eingaben | <i>50 Dateien tweets124 - tweets174</i> |
| Soll - Reaktion | <i>Es gibt keine Ausgabe und keine Fehlermeldungen</i> |
| Ist – Reaktion | <i>Es tritt keine Fehlermeldung auf und es gibt keine Ausgabe</i> |
| Ergebnis | <i>Der Test ist erfolgreich</i> |
| Unvorhergesehene Ereignisse | <i>In der Historydatei selbst waren exorbitant viele Kommata</i> |
| Nacharbeiten | <i>Der Fehler lag in der Methode readHistory(). Diese liest immer wieder die Historydatei aus und hinterlegt deren Inhalt in CorrelatedTags-Objekten. Dabei wurde allerdings das Komma, das durch die toString() Methode von ArrayList in die History geschrieben wurde nicht wieder entfernt, sodass jedesmal beim Lesen, der Tweets ein Komma vorne angehängt wurde. Dieser Fehler wurde behoben.</i> |

6.2 Zusammenfassung

Der TrendExtractor nimmt bezüglich des Testens eine besondere Position ein. Er ist zum einen das Herzstück des Projektes NewsMiner, da hier die tatsächlichen Trends berechnet werden. Zum anderen lässt sich das was er tut schlecht testen. Vor allem Semantisch ist nicht überprüfbar, ob die Komponente tatsächlich ihre Aufgabe erfüllt. Der TrendExtractor arbeitet mit Dateien in denen sehr viele Datensätze stehen - diese von Hand nachzustellen würde den Rahmen dieses Projektes mehr als sprengen. Dies betrifft vor allem die Methode extractTags(), aber als Resultat auch correlatedTags(),saveHistory(),readHistory(), predict(), scorePairs(),findEmergentTopics(). Deshalb wurden diese Methoden nur durch einen Test abgedeckt, der nichts über ihre semantische Korrektheit aussagt, sondern nur sicherstellt, dass im laufenden Betrieb keine Exceptions auftreten, sofern keine unerwarteten Ereignisse auftreten. Außerdem

wurden diese Methoden besonders sorgfältig von allen Auftragnehmern einem Review unterzogen.

7 Testdurchführung (2013-06-17)

Art des Tests: UnitTest

Ausgeführte Testfälle: $\langle T1700 \rangle$

Beteiligte Tester: Maria Joanna Born

Abgedeckte Funktionen: Trends extrahieren $\langle RM3 \rangle$

7.1 Testumgebung

Die Testfälle wurden unter Windows 7 mit NetBeans durchgeführt. Es wurde ein Testskript, sowie einige Testdateien verwendet.

7.2 Testprotokoll

| | |
|------------------------------------|--|
| Testfall | $\langle T1700 \rangle$ |
| Tester | <i>Maria Joanna Born</i> |
| Eingaben | <i>50 Dateien: tweets124 - tweets174</i> |
| Soll - Reaktion | <i>Es gibt keine Ausgabe und keine Fehlermeldungen und die History enthält die Kommata nur einfach an der richtigen Stelle</i> |
| Ist – Reaktion | <i>Es gibt keine Ausgabe und keine Fehlermeldungen und die History enthält die Kommata nur einfach an der richtigen Stelle</i> |
| Ergebnis | <i>Der Test ist erfolgreich</i> |
| Unvorhergesehene Ereignisse | <i>Keine</i> |
| Nacharbeiten | <i>Fehler in History-Datei wurde behoben.</i> |

7.3 Zusammenfassung

Der Test wurde in veränderter Form nochmal ausgeführt, da nach der zweiten Ausführung des Tests klar wurde, dass eine weitere Ausgabe überprüft werden muss. Der TrendExtractor hatte zwar bei der Ausführung keine Fehlermeldung mehr ausgegeben, hatte jedoch einen Fehler in der

History-Datei. Dieser Fehler hätte in der Komponenten *NewsAggregator* zu falschen Ergebnissen geführt und wurde behoben.

8 Testdurchführung (2013-06-22)

Art des Tests: Integrationstest

Ausgeführte Testfälle: $\langle T1500 \rangle$

Beteiligte Tester: Maria Joanna Born

Abgedeckte Funktionen: Twitterstream lesen $\langle RM2 \rangle$ und Trends extrahieren $\langle RM3 \rangle$

8.1 Testumgebung

Die Testfälle wurden unter Windows 7 mit NetBeans durchgeführt.

8.2 Testprotokoll

| | |
|------------------------------------|---------------------------------|
| Testfall | $\langle T800 \rangle$ |
| Tester | <i>Maria Joanna Born</i> |
| Eingaben | <i>Keine</i> |
| Soll - Reaktion | <i>Es gibt keine Ausgabe</i> |
| Ist – Reaktion | <i>Es gibt keine Ausgabe</i> |
| Ergebnis | <i>Der Test ist erfolgreich</i> |
| Unvorhergesehene Ereignisse | <i>Keine</i> |
| Nacharbeiten | <i>Keine</i> |

8.3 Zusammenfassung

Der Test hat ergeben, dass der *TrendExtractor* und der *TwitterCrawler* zusammen funktionieren. Das ist deshalb der Fall, weil die Struktur des Twitterstreams immer gleich ist. Das heißt, dass der *TrendExtractor* fest auf diese Struktur eingehen kann. Möglichkeiten Fehler abzufangen, die entstehen könnten, falls die Struktur des Twitterstreams geändert wird, werden von diesem Test nicht abgedeckt.

9 Testdurchführung (2013-06-23-001)

Art des Tests: Integrationstest

Ausgeführte Testfälle: $\langle T900 \rangle$

Beteiligte Tester: Maria Joanna Born

Abgedeckte Funktionen: Twitterstream lesen $\langle RM2 \rangle$ und Trends extrahieren $\langle RM3 \rangle$

9.1 Testumgebung

Die Testfälle wurden unter Windows 7 mit NetBeans durchgeführt.

9.2 Testprotokoll

| | |
|------------------------------------|--|
| Testfall | $\langle T900 \rangle$ |
| Tester | <i>Maria Joanna Born</i> |
| Eingaben | <i>Keine</i> |
| Soll - Reaktion | <i>Es erfolgt eine Ausgabe von gestemmtten Trendwörtern</i> |
| Ist – Reaktion | <i>Es erfolgt die Ausgabe von Tweettexten</i> |
| Ergebnis | <i>Der Test ist fehlgeschlagen</i> |
| Unvorhergesehene Ereignisse | <i>Keine</i> |
| Nacharbeiten | <i>Der Test hat die Auswirkungen eines Kommunikationsproblems aufgedeckt. Der TrendExtractor übergibt dem NewsAggregator eine Liste von TagTupeln. Bei der Schnittstellenabsprache hatte sich der Entwickler des TrendExtractors nicht klar ausgedrückt, so dass der Entwickler des NewsAggregators gedacht hatte, es würde eine Liste von Tweetwörtern übergeben werden. Der Fehler wurde behoben</i> |

10 Testdurchführung (2013-06-23-002)

10.1 Testprotokoll

| | |
|------------------------------------|---|
| Testfall | <i>⟨T900⟩</i> |
| Tester | <i>Maria Joanna Born</i> |
| Eingaben | <i>Keine</i> |
| Soll - Reaktion | <i>Es erfolgt eine Ausgabe von gestemmtten Trendwörtern</i> |
| Ist – Reaktion | <i>Es erfolgt die Ausgabe von gestemmtten Trendwörtern</i> |
| Ergebnis | <i>Der Test ist erfolgreich</i> |
| Unvorhergesehene Ereignisse | <i>Keine</i> |
| Nacharbeiten | <i>Keine</i> |

10.2 Zusammenfassung

Dieser Test hat ergeben, dass der *TrendExtractor* und der *NewsAggregator* insoweit korrekt zusammenarbeiten, dass der *TrendExtractor* dem *NewsAggregator* die korrekten Daten liefert.

11 Testdurchführung (2013-07-01)

Art des Tests: Integrationstest

Ausgeführte Testfälle: $\langle T1000 \rangle$

Beteiligte Tester: Maria Joanna Born

Getestete Funktionalitäten: RSSFeeds speichern und Newsfeeds nach relevanz Filtern

11.1 Testumgebung

Die Testfälle wurden unter Windows 7 mit NetBeans durchgeführt.

11.2 Testprotokoll

| | |
|------------------------------------|---|
| Testfall | $\langle T1000 \rangle$ |
| Tester | <i>Maria Joanna Born</i> |
| Eingaben | <i>Keine</i> |
| Soll - Reaktion | <i>Es erfolgt eine Ausgabe von gestemmtten Artikelwörtern</i> |
| Ist – Reaktion | <i>Es erfolgt die Ausgabe von gestemmttem Artikelwörtern</i> |
| Ergebnis | <i>Der Test ist erfolgreich</i> |
| Unvorhergesehene Ereignisse | <i>Keine</i> |
| Nacharbeiten | <i>Keine</i> |

11.3 Zusammenfassung

Bei diesem Test wurde festgestellt, dass die getesteten Funktionalitäten korrekt ausgeführt wurden.

12 Testdurchführung (2013-07-05)

Art des Tests: Funktionstest

Ausgeführte Testfälle: $\langle T2000 \rangle$

Beteiligte Tester: Arne Brüsch

Abgedeckte Funktionen: Erstellen eines Nutzerkontos $\langle F10 \rangle$, Anmeldung eines Nutzers $\langle F20 \rangle$

12.1 Testumgebung

Als Betriebssystem wurde Debian Wheezy eingesetzt. Die Tests wurden mit sbt ausgeführt.

12.2 Testprotokoll

| | |
|------------------------------------|---|
| Testfall | $\langle T2000 \rangle$ |
| Tester | Arne Brüsch |
| Eingaben | Trends, die aus einem einstündigen Twitter-Mitschnitt extrahiert wurden (konkret enthaltener Trend z.B. 'truth NSA') sowie Newsartikel, die im selben Zeitraum geholt wurden. |
| Soll - Reaktion | Newsartikel sollen gefiltert werden, Ergebnisse müssen nach dem Test in der Datenbank 'NewsMinerDB' abgelegt sein. |
| Ist - Reaktion | NewsAggregator läuft erfolgreich durch und endet ohne Fehler. Daten sind in der Datenbank abgelegt. |
| Ergebnis | Test erfolgreich. |
| Unvorhergesehene Ereignisse | Keine |
| Nacharbeiten | Keine |

12.3 Zusammenfassung

Die Tests sind erfolgreich abgeschlossen. Während der Testläufe stellte sich die falsche Verwendung einer Bibliothek (sspace) heraus, aufgrund der NewsAggregator zunächst nicht funktionsfähig war.

13 Testdurchführung (2013-07-09)

Art des Tests: Evaluation der Benutzeroberfläche durch Testpersonen (Abnahmetest)

Ausgeführte Testfälle: $\langle T100 \rangle$, $\langle T600 \rangle$, $\langle T700 \rangle$

Beteiligte Tester: Jana Sarah Riquel

Abgedeckte Funktionen: Erstellung eines Nutzerkontos $\langle F10 \rangle$, Anmeldung eines Nutzers $\langle F20 \rangle$, Abmelden eines Nutzer $\langle F30 \rangle$, Hinzufügen einer Nachrichtenquelle $\langle F80 \rangle$, Löschen einer Nachrichtenquelle $\langle F90 \rangle$

13.1 Testumgebung

Um eine Evaluation der Benutzeroberfläche von NewsMiner durchzuführen, wurden 25 unabhängige Testpersonen gebeten sich anhand eines Testskripts durch die Webplattform von NewsMiner zu navigieren und anschließend Bewertungen in den Bereichen allgemeiner Eindruck, Design, Bedienung, Nachrichtenartikel und Fehlermeldungen abzugeben. Die Testfälle wurden unter Windows 7 auf einem lokalen Webserver durchgeführt. Es wurde eine englische Systemumgebung verwendet. Die Seiten wurden mit Firefox 19.0.2 geöffnet.

13.2 Testprotokoll

| | |
|------------------|---|
| Testfälle | $\langle T100 \rangle$, $\langle T600 \rangle$, $\langle T700 \rangle$ |
| Tester | <i>Jana Sarah Riquel</i> |
| Eingaben | <i>Jedem Testteilnehmer wurde zu Beginn der Evaluation der Testbogen "Testbogen_NewsMiner.pdf" ausgehändigt mit der Bitte diesen sorgfältig und selbstständig zu bearbeiten. Hierfür sollte er zunächst den Anweisungen des Testbogens folgen und nach dem erfolgreichen Absolvieren die entsprechenden Fragen beantworten. Die Webplattform NewsMiner wurde in diesem Rahmen von den Testpersonen mit dem Nutzerprofil "Tester" genutzt. Hierfür wurden die Logindaten "Loginname: tester; Passwort: newsminer" genutzt.</i> |

| | |
|------------------------|--|
| Soll - Reaktion | <i>Das Ziel der Evaluation bestand darin an unabhängigen Testpersonen die Außenwirkung des Produkts NewsMiner zu testen. Um so garantieren zu können, dass eine selbsterklärende und einfache Nutzung der Webplattform möglich ist, das Produkt einen positiven Eindruck auf den Nutzer macht und eine intuitive Bedienung durch eine ansprechende grafische Oberfläche möglich ist.</i> |
| Ist – Reaktion | <i>Bei der Evaluation von NewsMiner schnitt das Design der Webanwendung mit der Note 1,3 am Besten ab. Besonders positiv fiel den Testpersonen die farbliche Gestaltung der Internetseite auf. Hierbei stellte sich somit heraus, dass die Benutzeroberfläche von NewsMiner optisch besonders ansprechend gestaltet wurde. Auch die Umsetzung und Darstellung der Nachrichtenartikel war mit einer Durchschnittsnote von 1,9 zufriedenstellend. Besonders von der Übersicht der Artikel waren die Tester überzeugt. Die gesuchten Themen waren so schnell und leicht zu finden, was auch dafür sprach, dass die Anzahl der angezeigten Artikel passend war. Die Quantität ermöglichte eine gute Auswahlmöglichkeit, verhinderte aber gleichzeitig durch die vorherige Filterung die Überflutung durch unzählige Artikel. Auch die Qualität der Artikel deckte sich mit den Erwartungen der Testpersonen. Ein Weiterer zu bewertender Aspekt war die Bedienbarkeit der Webplattform. Auch diese schnitt mit einer Durchschnittsnote von 2,1 gut ab. Die Testpersonen gaben an, dass ihnen die Navigation durch das Produkt nach kurzer Eingewöhnungsphase leicht von der Hand ging. Die Menüpunkte der Anwendung sind übersichtlich gestaltet und führen den Nutzer ohne unnötige Umwege zur gewünschten Funktion. Die Letzte zu testende Kategorie war der Umgang mit auftretenden Fehlern und deren Fehlermeldungen. Dieser Aspekt wurde von den Testern mit einer Durchschnittsnote von 2,4 bewertet. Bei fehlerhaften Eingaben war die Information über den aufgetretenen Fehler durch das Produkt zwar zufriedenstellend, jedoch wurde das fehlende Erläutern des Weiteren Vorgehens durch den Nutzer bemängelt.</i> |

| | |
|------------------------------------|---|
| Ergebnis | <i>Die Evaluation von NewsMiner ist wie erwartet sehr gut verlaufen. Die selbstständige Bearbeitung des Testbogens durch die Testpersonen war nahezu komplikationsfrei. Aufrufen der Webplattform, Anmelden, Abmelden und Registrieren auf der Anwendung von NewsMiner klappten in allen Fällen problemlos, lediglich das Einfügen von RSS-Feeds im Nutzerprofil erforderte gelegentlich Hilfestellung durch den anwesenden Experten des Entwicklerteams. Die Ergebnisse der Befragung des Testbogens fielen in allen Kategorien zufriedenstellend aus. Besonders gut wurde hier das Design und die Umsetzung bzw. der Aufbau der Nachrichtenartikel bewertet. Ebenfalls gut abgeschnitten haben die Kategorien "allgemeiner Eindruck" und "Bedienung" der Webplattform. Lediglich die Kategorie "Fehlermeldungen" ist nicht optimal bewertet worden. Die Evaluation von NewsMiner ist somit erfolgreich verlaufen. Und hat gezeigt, dass die zu erreichende Bedienungs-freundlichkeit des Produkts für den Nutzer erzielt wurde.</i> |
| Unvorhergesehene Ereignisse | <i>Keine</i> |
| Nacharbeiten | <i>Nach der Evaluation von NewsMiner konnte festgestellt werden, dass im Bereich des Nutzerprofils eine Verbesserung des Produkts möglich ist. Die Funktion "RSS-Feeds einfügen" könnte durch Standard-Eingaben vereinfacht werden, indem dem Nutzer ein bestimmter Pool an RSS-Feeds zur Verfügung gestellt wird. So ist er nicht in der Pflicht selbstständig RSS-Feeds zu suchen und auszuwählen. Des Weiteren ist bei der Evaluation das mittelmäßige Abschneiden der Fehlermeldungen aufgefallen. Hier wurde durch die Testpersonen der Wunsch geäußert die Fehlermeldungen klarer zu formulieren und vor Allem das weitere Vorgehen näher zu beschreiben. Um den selben Fehler in Zukunft verhindern zu können. Dies kann durch eine einfache Idealisierung des Textes mit genaueren Anweisungen bei Fehlermeldungen realisiert werden.</i> |

13.3 Zusammenfassung

Abbildung 13.1: Ergebnisse der Evaluation

- bei der Evaluation wurden 25 unabhängige Testpersonen ohne jegliche Vorkenntnisse dar-

um gebeten die Webplattform mit Hilfe eines Testbogens zu bewerten

- Design und Umsetzung der Nachrichtenartikel waren hierbei sehr gut
- Bedienung und Fehlermeldungen schnitten ebenfalls gut ab
- generell wurde die allgemeine Gestaltung der Anwendung mit einer Durchschnittsnote von 1,7 bewertet, somit kann die Evaluation als erfolgreich angesehen werden, die optisch ansprechende Benutzeroberfläche von NewsMiner ist zur Zufriedenstellung der zukünftigen Kunden garantiert worden
- alle im Testplan festgelegten zu testenden Musskriterien wurden durch diesen Abnahmetest abgedeckt