



UNIVERSITÉ DE FRIBOURG SUISSE
UNIVERSITÄT FREIBURG SCHWEIZ

Wirtschafts- und Sozialwissenschaftliche Fakultät Universität Fribourg
Information Systems Research Group

Eingereicht bei : Prof. Dr. Andreas Meier

Betreuer: Darius Zumstein

Analyse von Online-Kundenbewertungen
—
Prototyp eines Rating-Parsers

Eggel Elias

Fribourg, 30. Oktober 2010

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1. Einleitung	4
1.1 Vorwort	4
1.2 Der Weg und Trend zum eTourismus	5
1.3 Bedeutung von Kundenbewertungen	5
1.4 Hotelpartner GmbH	7
2. Vorgehensweise	8
2.1 Prototyping	8
2.2 Kurze Beschreibung des Prototyps	9
3. Verwendete Sprachen / Hilfsmittel	10
3.1 Java	10
3.2 PHP	10
3.3 Cron	11
3.4 SQL	11
4. Implementierung in Netbeans	13
4.1 Zugriff auf Bewertungsportale	13
4.2 Herauslesen der Bewertungen	14
4.3 Beispiel am Portal Booking.com	14
4.4 Beziehungen zwischen den Java-Klassen	16
4.5 Übersicht der Datenbank Tabellen	16
5. Schlusswort	19
6. Literaturverzeichnis	20
7. Anhang (CD mit Source-Code)	-

Abbildungen

Abb. 1	Relevante Informationsquellen für die Kaufentscheidung
Abb. 2	Internetauftritt von hotelpartner.ch
Abb. 3	Skizze Yield-Management
Abb. 4	Rating-Parser Prinzip
Abb. 5	Bewertungen auf dem online Portal booking.com
Abb. 6	Beziehungen zwischen den Java-Files
Abb. 7	Tabelle RAT_Hotel
Abb. 8	Tabelle RAT_Portal
Abb. 9	Tabelle RAT_Links
Abb. 10	Tabelle RAT_Ratings
Abb. 11	Tabelle tmp
Abb. 12	Datenbankschema

Listings

Listing 1	parse.sh
Listing 2	SQL-Abfrage
Listing 3	Webseite parsen
Listing 4	Zeilenumbruch löschen
Listing 5	Zerteilung in Blöcke
Listing 6	Herausfiltern der Bewertungen

Tabellen

Tabelle 1	cronjob
-----------	---------

Abkürzungen

PHP	Hypertext Preprocessor
-----	------------------------

1. Einleitung

1.1 Vorwort

Bei steigender Konkurrenzzahl wird heutzutage bei den meisten Unternehmen stark auf die Kundenzufriedenheit geachtet. Diese kann auch die verschiedensten Arten gemessen werden.

Eine davon ist die direkte Rückmeldung der Kunden über das Internet, sei es positiv oder negativ.

Besonders für Hotels ist eine hohe Kundenzufriedenheit unerlässlich und es ist wichtig, Feedbacks von Kunden zu erhalten und so gewisse Probleme zu beseitigen.

Es gibt online viele Portale, auf denen ehemalige Kunden von Hotels ihre Meinung kundgeben können. Mit einem Rating Parser werden diese Portale zusammengefasst und übersichtlich auf einer einzigen Seite wiedergegeben.

Der Parser wird später, wenn auch nicht genau in dieser Form, von der Firma Hotelpartner als Angebot für die Hotels genutzt und weiter ausgearbeitet.

Dabei gibt es noch sehr viel Spielraum, wie das Angebot vergrößert und noch interessanter gestaltet werden kann.

Die drei implementierten Portale des Rating-Parsers dieser Arbeit sind:

- Booking.com [Booking 2010]
- Hotel.de [Hotel 2010]
- Expedia.de [Expedia 2010]

Mit dieser Arbeit soll erkenntlich werden, wie bedeutsam solche Kundenbewertungen auf Online-Portalen überhaupt sind. Ausserdem soll gezeigt werden, wie man sich diese Bewertungen zu Nutze machen kann, welche Gefahren diese mit sich bringen und an einem Prototyp gezeigt werden, wie man einen Rating-Parser selber umsetzen kann.

1.2 Der Weg und Trend zum eTourismus

In den 70er Jahren gab es noch keine solche Portale und es war auch nicht möglich, Buchungen direkt über das Internet zu tätigen. In den Reisebüros wurde mit Computer-Reservierungssystemen (CRS) gearbeitet. Anhand diesen System konnten Verfügbarkeiten abgefragt und Buchungen gemacht werden. Später, um 1990, verloren solche Systeme mit dem Aufschwung des Internets an Bedeutung. Konsumenten mussten Buchungen nicht mehr über Intermediäre wie Reisebüros tätigen, sondern konnten von nun an direkt mit den Tourismusbetrieben agieren.

Da die Zahl der Internetnutzer und damit die Anzahl potentieller Kunden rapide ansteigt, ist es in der Tourismus Branche extrem wichtig, Online Dienste anzubieten und zu verbessern. Schon 2009 informierten sich 47% der Deutschen im Internet über Reiseangebote und 24% buchen ihre Reisen mittlerweile online. Die Tourismus Branche erkannte diesen Trend und war so die Erste, die sich umfassend mit Informations- und Kommunikationstechnologien beschäftigte und auch einsetzte. Gut erkennen kann man diesen Trend anhand der Umsätze im europäischen Online-Reisemarkt. Im Jahr 2000 betrug dieser 2.5 Milliarden Euro pro Jahr und im Jahr 2009 65.2 Milliarden pro Jahr [Lassnig & Reich 2009].

1.3 Bedeutung von Kundenbewertungen

Durch das Internet wurde es aber nicht nur einfacher, selber Reisen zu buchen, sondern man wurde als Kunde auch immer interaktiver. Eine wichtige Rolle spielen dabei Portale, auf denen es den Kunden möglich ist, Hotels zu bewerten und die eigene Meinung zu äussern.

Die Haltung eines Kunden gegenüber einer Firma spielt in jedem Unternehmen eine sehr grosse Rolle. So auch in Hotelbetrieben, welche eine grosse Konkurrenzzahl aufweisen. Die Meinung der Kunden wurde früher über Mund-zu-Mund-Propaganda kundgetan, was nur einer beschränkten Anzahl potentieller neuer Kunden zugute kam. Heute sind fast alle Hotels online buchbar und auf diversen Portalen registriert.

Wenn sich heute jemand ein Hotel buchen will, ist die Wahrscheinlichkeit, dass sich dieser im Internet zuerst Kundenbewertungen anschaut, sehr gross. Verschiedene Studien bestätigen, dass sich Reisende oft mehr Reiseberichten und Meinungen anderer Reisenden vertrauen als Informationen der Hotelbetrieben selbst. Aus diesem Grund ist es sehr wichtig für die Hotels, diese Kundenbewertungen zu lesen und auch ernst zu nehmen. Denn bei einem Hotel mit überwiegend schlechten Bewertungen wird sich kaum jemand ein Zimmer mieten.

So können Bewertungssysteme als Chance genutzt werden, positiv Werbung zu betreiben oder aber als Bedrohung angesehen werden [Lassnig & Reich 2009].

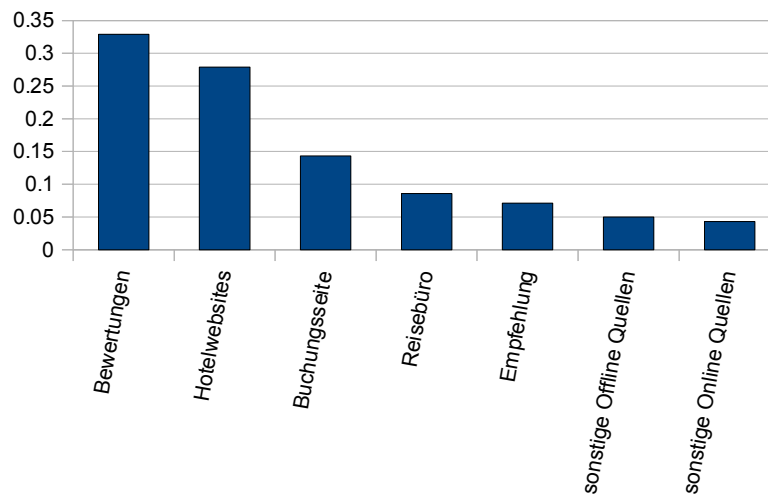


Abb. 1 Relevante Informationsquellen für die Kaufentscheidung
[Lassnig & Reich 2009]

Das Reisebüro als Informationsquelle existiert zwar noch, aber ist klar irrelevanter als die 3 Online Quellen Kundenbewertungen, Hotelwebsites und Buchungsseiten. Dies wird auch in der Abb.1 klar: Mit weniger als 10% Anteil an der Kaufentscheidung ist die Wichtigkeit des Reisebüros deutlich kleiner als die der Hotelbewertungen mit über 30%. In den nächsten Jahren wird das Reisebüro als Informationsquelle mit dem Anstieg des Internets wohl immer weniger wichtig werden.

Dass Kundenbewertungen kritisch hinterfragt werden sollten, zeigt ein Beispiel aus der Elektronik Branche. Ein Geschäftsführer von der Firma Wetab stellte unter falschen Namen Kommentare auf den Online-Shop Amazon und lobte das Gerät hoch. In diesem Fall konnte der 'richtige' User auffindig gemacht werden, da er in seinem Account vergass, Einstellungen zu machen, welche seine Persönlichkeit verschleiern können [Zeit 2010].

Allerdings ist es in anderen Fällen meistens sehr schwierig oder unmöglich, solche gefälschten Kommentare zu erkennen und sind daher mit Vorsicht zu genießen.

Zu Unterscheiden sind die positiven von den negativen Bewertungen, da die negativen viel stärker den Kunden bei seiner Kaufentscheidung beeinflussen als positive. Im Fall von negativen Aussagen über ein Produkt kauft 43% der Kunden das Produkt selten oder nie. Auf Grund von positiven Aussagen kaufen aber nur 28% das angepriesene Produkt [Heckmann 2008].

1.4 Hotelpartner GmbH

Hotelpartner GmbH wurde zum Zweck gegründet, Hotelbetriebe und Hoteliers, so wie professionelle Anbieter von Dienstleistungen und IT-Systemen im modernen Vertrieb und in technologischen Angelegenheiten zu unterstützen und zu beraten [Hotelpartner 2010].

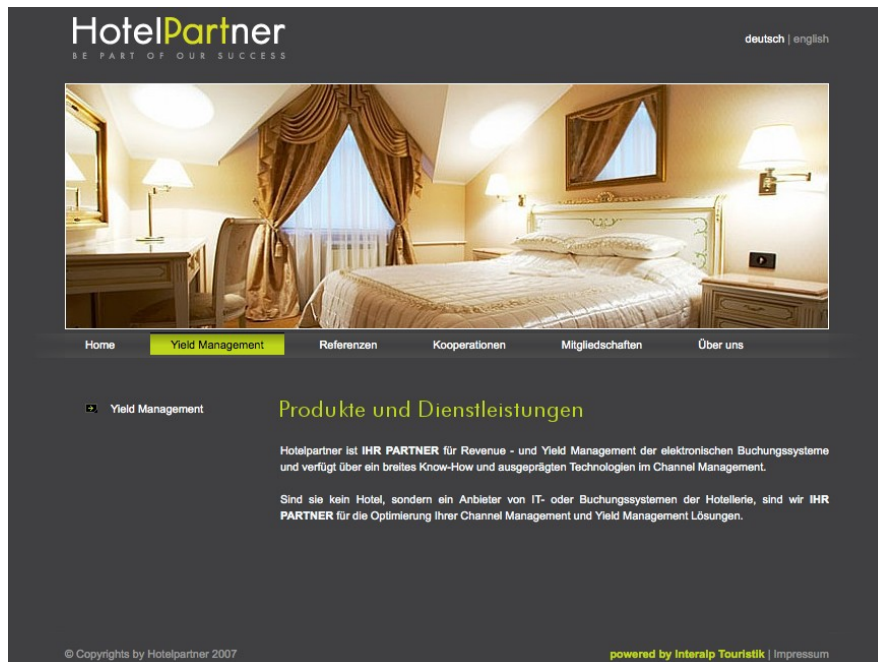


Abb. 2. Internetauftritt von Hotelpartner.ch

Die Firma betreibt auf Online Portalen für verschiedene Hotels Yield Management (Ertragsmanagement) und berechnet dynamisch Preise für die Hotelzimmer.

In Perioden mit grösserer Nachfrage steigen die Preise und bei niedrigerer Nachfrage sinken sie.

Allerdings variieren die Preise nicht nur auf einer Produktebene, sondern direkt für einzelne Produkte. Es wird also nicht nur nach bestimmten Zeitperioden, wie beispielsweise Ferienzeit, unterschieden, sondern, je nach Verfügbarkeit, nach jedem einzelnen Tag.

Das Konzept wird an Abb.3 klar. Ein Hotel sendet dabei seine Verfügbarkeiten an Hotelpartner [1], welches anhand dieser Informationen die Preise und Verfügbarkeiten dynamisch festsetzt und an die Buchungsportale weiterleitet [2]. Auf diesen Portalen können Kunden die gewünschten Zimmer buchen und die Buchungsinformationen werden zurück zum Hotel geschickt [3], welches anhand dieser Informationen die tatsächliche Reservierung durchführt.



Hotelpartner GmbH | Laupenstrasse 6 | Postfach | CH-3001 Bern | t +41 31 302 70 90 | f +41 31 302 70 91 | info@hotelpartner.ch | www.hotelpartner.ch

Abb. 3. Skizze Yield-Management [Hotelpartner 2010]

2. Vorgehensweise

2.1 Prototyping

Prototyping ist eine Methode in der Softwareentwicklung. Anhand dieser Methode kommt man schnell zu ersten Ergebnissen und Änderungswünsche können ohne grosse Probleme in das Projekt einfließen. Ausserdem werden Fehler schon früh erkannt und können so schnell behoben werden können [Von Utz 2009].

Es gibt verschiedene Arten von Prototyping. In dieser Arbeit wurde das evolutionäre Prototyping verwendet, dass sich gut für kleinere Projekte eignet. Bei diesem Ansatz wird der Kern des Programmes programmiert. Das Ziel dabei ist es, schon früh eine erste Version zu liefern, dass man sich einen ersten Überblick über das Programm machen kann [Schröder 2008].

Im Gegensatz zum explorativen und experimentellen Prototyping wird der Prototyp zum Schluss nicht weggeworfen. Daher ist es wichtig, bei der Implementierung sauber zu programmieren, um spätere Probleme zu vermeiden.

Das Projekt wird fortlaufend erweitert und die Ziele sind noch nicht klar gesetzt [Mayr 2005].

Es ist daher dem Wachstumsmodell (evolutionäres Modell) zuzuordnen. Die Aufgaben können in kleinere Teilaufgaben zerteilt werden und können autonom entwickelt und später zusammengesetzt werden.

2.2 Kurze Beschreibung des Prototyps

Auf den verschiedenen Portalen sucht der Rating Parser nach allen Bewertungen, welche ehemalige Hotelgäste zu bestimmten Hotels abgegeben haben. Dabei wird je nach Portal zusätzlich unterschieden, ob der Kommentar negativ oder positiv ist.

Diese Bewertungen werden dann in eine Datenbank zusammengefasst und später über ein Interface, realisiert mit Hypertext Preprocessor, kurz PHP, angezeigt.

Das PHP Interface greift auf eine mySQL Datenbank zu, in der die Bewertungen abgespeichert werden. Dabei wird das Aussuchen der Bewertungen an sich wird mit Hilfe von Java bewerkstelligt. Um auch immer die aktuellen Kundenbewertungen angezeigt zu haben, werden die Portale mittels Cron einmal am Tag nach neuen Bewertungen von Kunden durchsucht.

3. Verwendete Sprachen / Hilfsmittel

3.1 Java

Der grösste Teil des Rating-Parsers ist in der Programmiersprache Java geschrieben. Java ist objektorientiert und wurde um 1990 von der Firma Sun entwickelt. Sie war zuerst nicht für Internetanwendungen geplant, sondern um Systeme im Elektronikbereich wie z.B. Waschmaschinen zu steuern.

Ohne grossen Erfolg in diesem Bereich hatte Java diesen dann wenige Jahre später, während dem Boom des Internets. Da es noch keinen Webbrowser gab, der Java unterstützte (da es Java bis dahin auch noch gar nicht gab), veröffentlichte Sun den Browser HotJava, der auf Java basierte. Ab 1995, als der erste Java-kompatible Browser (Netscape Navigator) auf den Markt kam, gelang der Durchbruch von Java [Middendorf/Singer/Heid 2002].

Mit Java war es nun möglich, im Internet Applikationen zu nutzen, die klein und sicher waren. Das erste Mal waren die Webseiten im Internet nicht mehr nur statisch, sondern es konnten online interaktive Elemente ausgeführt werden [Samaschke 2004].

Java kann von jedem kostenlos genutzt werden, ist aber keine hundertprozent freie Software, da bestimmte Klassen von Sun zwar verwendet werden können, man in diese aber keine Einsicht hat.

3.2 PHP

Die Geschichte von PHP begann etwas später als die von Java um 1995. Es wurde entwickelt, um die Programmierung von Webseiten einfacher zu gestalten (die Abkürzung PHP bedeutete 'Personal Homepage'). Davor wurde vor allem in C oder Pearl geschrieben, was sehr viel umständlicher war. Heute wird PHP vielseitiger gebraucht als nur für die Programmierung von Webseiten und die Abkürzung steht mittlerweile für 'Hypertext Preprocessor'.

Der Durchbruch von PHP gelangte mit der Version 4.0, welche im Jahr 2000 veröffentlicht wurde und wurde zur wichtigsten Sprache für die Webprogrammierung.

Im Gegensatz zu Java ist PHP ein Open-Source-Projekt. Aus diesem Grund wird PHP bei Banken und grossen Unternehmen oft nicht eingesetzt, da das Vertrauen in kommerzielle Anbieter immer noch grösser ist [Hudson 2006].

3.3 Cron

Cron entstand aus dem Wunsch heraus, Programme automatisch immer wieder durchlaufen zu lassen, ohne selber etwas zu machen. Der Befehl dazu wird in der crontab, eine sechs spaltige Tabelle, gespeichert. Dabei dienen die ersten fünf Spalten der Zeitangabe, wann der Befehl ausgeführt werden soll. Die sechste Spalte ist für den Benutzernamen reserviert, unter welchem der Befehl ausgeführt werden soll und in der siebten Spalte steht der eigentliche Befehl.

Der implementierte Cron-job sieht folgendermassen aus:

minute	hour	mday	month	wday	who	command
*	17	*	*	*	*	/.../parse.sh

Es wird also jeden Tag im Jahr um 17.00 Uhr die Datei parse.sh ausgeführt. Ein Stern steht dabei für jede/n Minute/Tag/Monat/Wochentag. Der Stern unter 'who' sagt aus, dass es unabhängig von Benutzer ausgeführt werden soll. In der Datei parse.sh stehen folgende 3 Zeilen:

<pre>#!/bin/bash cd /Users/senf/Documents/ratings/ java -jar ratings.jar</pre>
--

Dabei wird in der ersten Zeile festgelegt, dass es sich um ein Bash-Skript handelt. Im nächsten Schritt wird der Ort angegeben, wo sich die auszuführende Datei befindet und zum Schluss wird die Datei ratings.jar gestartet.

3.4 SQL

SQL steht für *Structured Query Language* und wird verwendet, um bestimmte Datensätze aus Datenbanken herauszulesen oder zu ändern. Im Gegensatz zu prozeduralen Sprachen, die abstrakter und für Anfänger schwierig zu verstehen sind, ist SQL eine deskriptive Sprache.

Diese ist unserer natürlichen (englischen) Sprache ähnlich und erscheint auch gelegentlichen Nutzern logisch.

Ein Beispiel einer SQL-Abfrage: „Liste **alle Namen** aus der Tabelle **Mitarbeiter** auf, deren Vorname **Hans** ist.“ In SQL sieht dies folgendermassen aus:

Listing 2: SQL-Abfrage

```
SELECT * (alle Namen)  
FROM Mitarbeiter (aus der Tabelle Mitarbeiter)  
WHERE Vorname = 'Hans' (mit dem Vornamen Hans)
```

Es braucht also keine allzu grossen Kenntnisse von SQL, um damit arbeiten zu können. Es muss nicht genau bestimmt werden, wie die Abläufe zur Bereitstellung der Information aussehen müssen.

Neben den Operatoren SELECT, FROM und WHERE gibt es noch andere, nützliche Befehle. So kann man beispielsweise mit DISTINCT nur die Einträge auslesen, welche sich von allen anderen unterscheiden und mit ORDER BY die Ausgabe nach verschiedenen Kriterien ordnen.

4. Implementierung in Netbeans

4.1 Zugriff auf Bewertungsportale

Um auf die Bewertungen der Kunden Zugriff zu bekommen, wird zuerst eine Verbindung zu den Portalen aufgebaut. Diese ist für alle gleich.

Es werden in diesem Fall zwei Java-Files gebraucht: WebFile.java und HttpGet.java.

Der Code von WebFile.java wurde aus dem Internet entnommen und unverändert gebraucht. Mit der Java-Klasse InputStream wird der gesamte Inhalt der Webseite eingelesen und an HttpGet.java weitergegeben.

Die Klasse HttpGet kann für die einzelnen *Portal-Parser* verwendet werden, um den Inhalt der Webseiten zu erhalten. Dabei wird eine neue Instanz von HttpGet erzeugt:

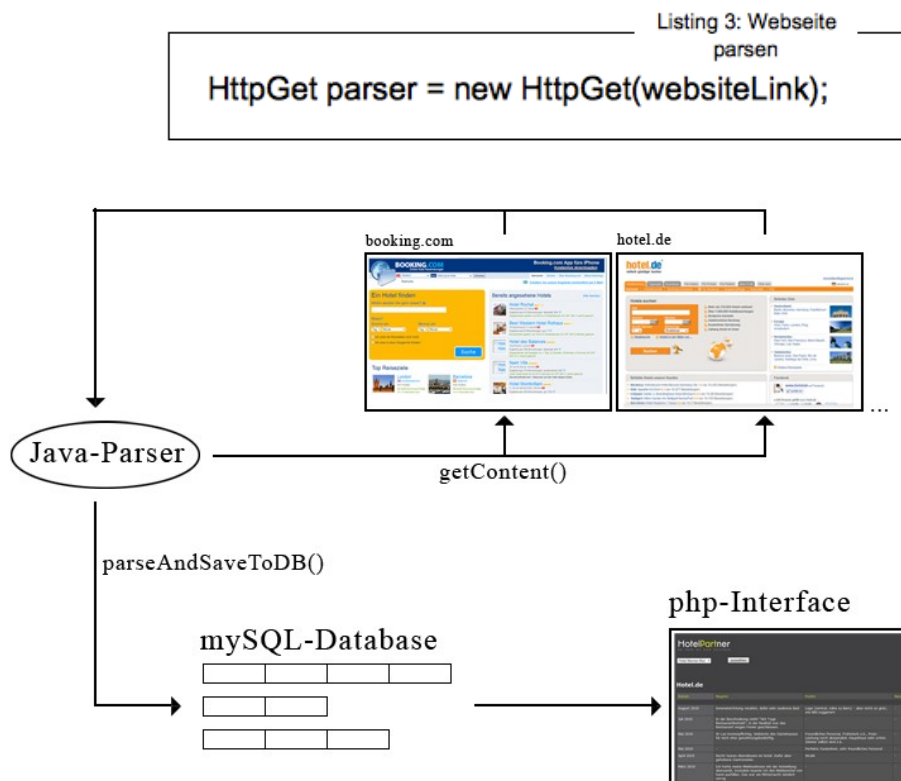


Abb. 4. Rating-Parser Prinzip

In der Abb. 4 ist der Ablauf des Rating-Parsers dargestellt. Mittels einer Methode `getContent()` wird der Quellcode des Portals übergeben. Dieser Code wird dann zurechtgeschnitten und an eine Datenbank weitergegeben. Diese wird zuletzt über ein php-Interface benutzerfreundlich dargestellt.

4.2 Herauslesen der Bewertungen

In den Klassen für die jeweiligen, konkreten Portalen, wird dann von `HttpGet.java` dieser Inhalt übertragen und muss dann Portal-spezifisch zerlegt werden.

In jedem Falle müssen im Text alle Zeilenumbrüche entfernt werden, damit der Text auf einer einzigen Linie angezeigt wird. Ansonsten gibt es Probleme bei der Weiterverarbeitung des Textes.

Dies wird mit einem `replace`-Befehl gemacht. Anstatt einem Zeilenumbruch wird ein Leerzeichen gesetzt. Der gesamte ausgegebene Text ist in der Variable `response` gespeichert und wird wie folgt zerlegt.

Listing 4: Zeilen-
umbruch löschen

```
response = response.replace("\n", " ");
```

Soweit ist der Vorgang für alle 3 Portale derselbe. Ab diesem Zeitpunkt fängt dann das eigentliche 'parsen', die Verarbeitung des rohen Textes an.

4.3 Beispiel am Portal Booking.com

Für jede Bewertung, die zu einem einzelnen User gehört, muss ein Block gebildet werden, der dann in weitere Teile zerschnitten wird.

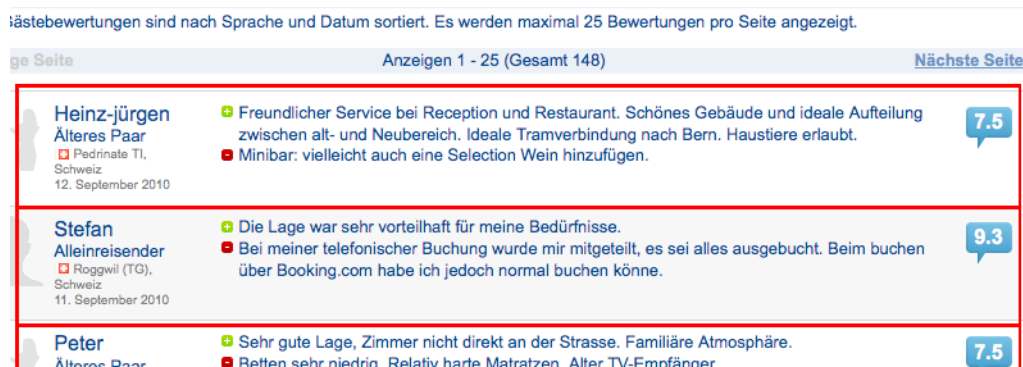


Abb. 5. Bewertungen auf dem online Portal Booking.com

Dies geschieht mit den Klassen *Pattern* und *Matcher*:

Listing 5: Zerteilung in
Blöcke

```
Pattern patt = Pattern.compile("<span class=\"cell_user_date dtreviewed\">.*</p>");  
Matcher m = patt.matcher(string);
```

Es wird nun nach Blöcken im Quelltext gesucht, die mit `` anfangen und mit `</p>` enden. Dies schliesst die positive und negative Bewertung des Users ein.

Nun muss aber noch zwischen positiver und negativer Bewertung unterschieden werden, da diese später in der Datenbank verschiedene Plätze belegen werden.

Mit folgendem Code werden (hier die positiven) die Bewertungen herausgefiltert:

Listing 6: Herausfiltern
der Bewertungen

```
String line = string.substring( m.start(), m.end() );
String[] Tmp_pos = line.split("<p class=\"comments_good\" >");

if (Tmp_pos.length != 1) {
    String[] Tmp2_pos = Tmp_pos[1].split("</p>");
    pos = Tmp2_pos[0];
} else {
    pos = "-";
}
```

Die Variable *line* enthält nun jeweils einen Block, wie man ihn oben gesehen hat.

Dieser Textabschnitt wird nun in weitere Unterblöcke zugeschnitten (Block positive Bewertung + Block negative Bewertung).

Dies geschieht nach einem ähnlichen Prinzip wie beim ersten Zuschneiden. Der Text wird nun in Teile untergeteilt (*gesplittet*), welche mit `<p class=comments_good">` anfangen und mit `</p>` enden.

4.4 Beziehungen zwischen den Java-Klassen

Der Rating-Parser besteht aus zehn Java-Klassen. Die drei Klassen *Booking.java*, *HotelDe.java* und *Expedia.java* machen dabei, wie weiter oben schon beschrieben, die eigentliche Arbeit. Sie haben alle die gleiche Struktur und implementieren das Interface *portalInterface.java*, welches die 3 Klassen zwingt, die Methoden *getResponse()* und *parseAndSaveToDB(String response)* zu implementieren. *MainTest.java* ist die zu ausführende Datei. Sie macht aus den Klassen *Booking*, *HotelDe*, *Expedia*, *CreateTmp* und *DeleteTmp* Instanzen und bringt so das Programm zum laufen. Die Datei *DBConnection.java* stellt eine Verbindung zur Datenbank zur Verfügung und enthält den dazugehörigen Usernamen und Passwort. *HttpGet.java* wird gebraucht, um überhaupt auf die Portale zugreifen zu können und braucht dafür die Klasse *Webfile.java*, welche den entsprechenden Service bereitstellt.

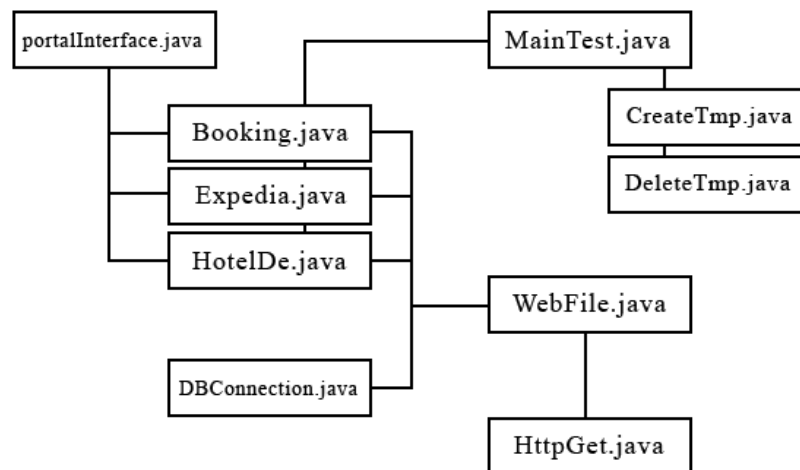


Abb. 6. Beziehungen zwischen den Java-Files

4.5 Übersicht der Datenbank Tabellen

Die Datenbank besteht aus fünf Tabellen, die miteinander verknüpft sind.

In der ersten Tabelle *RAT_Hotel* sind diejenigen Hotels gespeichert, von denen man die Bewertungen ansehen kann. In der zweiten Tabelle *RAT_Portal* sind die 3 implementierten Portale aufgelistet. Hotels und Portale sind dabei jeweils einem Primärschlüssel zugeordnet.

In der Tabelle *RAT_Links* werden die Hotels jeweils einem Portal zugewiesen und in der Spalte *link* die Links zu den Portalen mit den Bewertungen der jeweiligen Hotels gespeichert.

Die Tabelle *RAT_Ratings* enthält dann die eigentlichen Bewertungen. Zuletzt gibt es noch eine temporäre Tabelle namens *tmp*, deren Bedeutung weiter unter genauer ausgeführt ist.

Es braucht also 4 beständige und eine temporäre Tabelle in der Datenbank.

Abb. 7. Tabelle *RAT_Hotel*

#	id	name
1	1	Hotel Sternen Muri
2	2	Hotel Bern
3	3	Hotel Glärnischhof
4	4	Schweizerhof Zürich

Abb. 8. Tabelle *RAT_Portal*

#	id	name
1	1	booking
2	2	hotel.de
3	3	expedia

Abb. 9. Tabelle *RAT_Links*

#	id	linkPortal	linkHotel	link
1	1	1	1	http://www.booking.com/hotel/ch/landg...
2	2	2	1	http://www.booking.com/hotel/ch/swiss...
3	3	3	1	http://www.booking.com/hotel/ch/glaer...
4	4	4	1	http://www.booking.com/hotel/ch/schw...

Abb. 10. Tabelle *RAT_Ratings*

#	linkPortal	linkHotel	date	neg	pos	neu
1	1	1	2 September 26, 2010	the location is a touch too f...	the room was clean and all necesar...	-
1	1	1	2 September 26, 2010	the location is a touch too f...	the room was clean and all necesar...	-
3	1	1	2 September 13, 2010	The room was small; few el...	The location and the food at the res...	-
4	1	1	2 August 17, 2010	-	Central location. Nearby clock towe...	-
5	1	1	2 August 4, 2010	-	Well situated, comfortable mid rang...	-

Abb. 11. Tabelle *tmp*

#	linkPortal	linkHotel	date	neg	pos	neu
1	1	1	2 September 26, 2010	the location is a touch too f...	the room was clean and all necesar...	-
2	1	1	2 September 21, 2010	-	We were made to feel very welcom...	-
3	1	1	2 September 13, 2010	The room was small; few el...	The location and the food at the res...	-
4	1	1	2 August 17, 2010	-	Central location. Nearby clock towe...	-
5	1	1	2 August 4, 2010	-	Well situated, comfortable mid rang...	-

Um die Tabellen in einem Datenbankschema darzustellen wurden ihre Namen ein wenig verändert.

Die Tabellen *Hotel* und *Portal* sind Entitätsmengen mit dem Primärschlüssel H# bzw. P#.

Die Tabellen *verlinkt* und *bewertet* sind Beziehungsmengen mit jeweils denselben Fremdschlüssel.

Genau so wie die Tabelle *tmp*, welche dieselbe Struktur wie *bewertet* hat.

Hotel

H#	Name
----	------

Portal

P#	Name
----	------

verlinkt

P#	H#	Link
----	----	------

bewertet

P#	H#	Datum	Bewertung pos.	Bewertung neg.	Bewertung neutral
----	----	-------	----------------	----------------	-------------------

tmp

P#	H#	Datum	Bewertung pos.	Bewertung neg.	Bewertung neutral
----	----	-------	----------------	----------------	-------------------

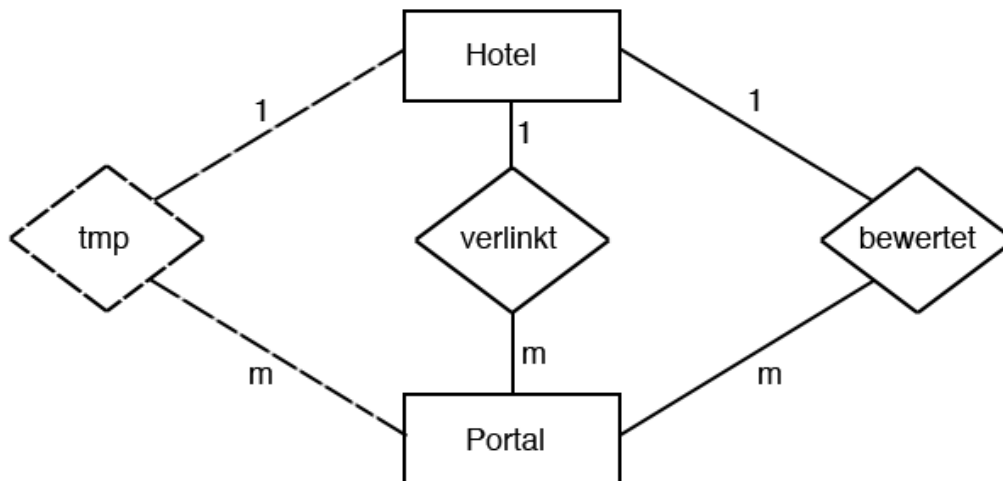


Abb. 12. Datenbankschema

Ein Hotel ist also auf mehreren Portalen verlinkt und wird auch auf mehreren Portalen bewertet.

Die Tabelle *tmp* wird bei jedem Durchlauf des Programmes anfangs gelöscht und dann jedes mal auf das Neue wiedererstellt.

Diese Tabelle ist nötig, da in der Tabelle *RAT_Ratings* nach mehrmaligem Durchlauf des Programms Redundanzen auftauchen. Aus der Tabelle *RAT_Ratings* werden die Tabellen eins zu eins übernommen und in die Tabelle *tmp* gespeichert. Mit dem Unterschied, dass durch eine SQL-Query mit *'AS SELECT'* gleiche Zeilen nur einmal vorkommen.

5. Schlusswort

Das Ziel dieser Arbeit war es, ein kleines Programm zu schreiben, welches der Hotellerie zur Unterstützung dienen sollte. Dabei wurden die Online Bewertungen von Hotelgästen untersucht, welche aus verschiedenen Portalen entnommen und zusammengefasst wurden. Ähnliche Tools gibt es zwar schon, sind aber immer noch rar. So macht es durchaus Sinn, selber ein solches zu schreiben. Vor allem da das Programm von der Firma *Hotelpartner* weiter ausgearbeitet wird und diese sicherlich auch Kunden (Hotels) haben werden, welche bereits Partner sind.

Das bis zu diesem Zeitpunkt implementierte Programm wird in diesem Rahmen noch nicht einen sehr grossen Nutzen haben, aber es ist ein Anfang eines Produkts, dass später bestimmt nutzvoll genutzt werden kann. Schon jetzt wird es, allerdings nicht von mir, von *Hotelpartner* weiter ausgearbeitet und kann dann hoffentlich einmal effektiv von Hotelbetrieben genutzt werden.

Auf der Webseite www.reviewpro.com ist die Idee der Zusammenführung der Bewertungen verschiedener Portale in einem grösseren Umfang umgesetzt worden.

So kann man unter anderem

- den Zeitraum der Kommentare selber bestimmen
- nur Kommentare bestimmter Sprachen anzeigen lassen
- anhand von Graphen sehen, wann und wie oft Bewertungen über das Hotel eingegangen sind
- den prozentualen Anteil von positiven, negativen und neutralen Bewertungen anzeigen

Weiter sollte in der Arbeit untersucht werden, wie bedeutsam Kundenbewertungen sind und wie sich diese auswirken. Dabei wurde klar, dass diese eine wichtige Stellung einnehmen und in Zukunft auch immer bedeutender werden, da immer mehr Menschen das Medium Internet in Anspruch nehmen. Da diese Informationsquellen so entscheidend für die Kaufentscheidung vieler Leute sind, müssen diese in Zukunft von Tourismusbetrieben auch genau beobachtet werden. Die Gefahr von Missbrauchsmöglichkeiten, sei es seitens der Kunden oder deren der Hotels selber, besteht nämlich und kann gravierende Folgen haben.

6. Literaturverzeichnis

- [Middendorf/Singer/Heid 2002] Middendorf, Stefan; Singer, Reiner, Heid, Jörn; *Java, Programmierhandbuch und Referenz für die Java-2-Plattform*, Standart Edition; 3. Auflage, dpunkt.verlag, Heidelberg, 2002
- [Hotelpartner 2010] Hotelpartner GmbH. available: <http://www.hotelpartner.ch>, zugegriffen am 28. Oktober 2010
- [Samaschke 2005] Samaschke, Karsten; *Java Master Class, Einstieg für Anspruchsvolle*, Pearson Studium, München, 2005
- [Hudson 2006] Hudson, Paul; *PHP in a nutshell*; Deutsche Übersetzung von Sigrid & Ulrich Speidel; O'Reilly Media, Inc., 1005 Gravenstein, 2006
- [Münz 2005] Münz, Stefan; *Professionelle Websites*, Programmierung, Design Administration von Webseiten, Addison-Wesley, München, 2005
- [Meier 2006] Meier, Andreas; *Relationale und postrelationale Datenbanken*, 6. Auflage, Springer, Berlin, Heidelberg, 2006
- [Lassnig & Reich 2009] Lassnig, Markus; Reich, Siegfried; *eTourismus*, dpunkt.verlag, Heidelberg, 2009
- [Zeit 2010] Zeit Online News: weTab-Chef zieht sich zurück. available: <http://www.zeit.de/wirtschaft/unternehmen/2010-10/wetab-chef-rueckzug>, zugegriffen am 30. Oktober 2010
- [Heckmann 2008] Heckmann, Benjamin; *Der Einfluss des Internets auf die Entscheidungsfindung des Kunden im Bankensektor*, Eine empirische Analyse; 1. Auflage, GRIN, 2008

- [Von Utz 2009] Von Utz, Daniel Motus; *Referenzmodell für die Montageplanung in der Automobilindustrie*, Herbert Utz Verlag, 2009
- [Schröder 2008] Schröder, Ulf; *Prototyping in der Softwareentwicklung*, Studienarbeit, GRIN, 2008
- [Mayr 2005] Mayr, Herwig; *Projekt Engineering: Ingenieurmäßige Softwareentwicklung in Projektgruppen*, Hanser Fachbuchverlag, 2005