



**Hochschule**  
**Augsburg** University of  
Applied Sciences

# Projektdokumentation Rodisare

## SS 2013

### Agile Webanwendungen mit Python

Fakultät Informatik

vorgelegt von: Roland Reitböck (924762)

Dietmar Sach (924738)

Dozent: Dipl.-Inf. (FH), Dipl.-Designer (FH) Erich Seifert

© 2013

Dieses Werk einschließlich seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Wahlpflichtfach: Agile Webanwendungen mit Python . . . . .	1
1.2	Grundlagen des World Wide Webs . . . . .	1
<b>2</b>	<b>Projekt Rodisare</b>	<b>3</b>
2.1	Projektplanung . . . . .	3
2.1.1	Web-Framework . . . . .	3
2.1.2	Versionsverwaltung . . . . .	3
2.1.3	Integrierte Entwicklungsumgebung . . . . .	3
2.1.4	Modulplanung . . . . .	4
2.1.5	Projektdokumentation . . . . .	4
2.2	Entwurf . . . . .	5
2.2.1	Projektstruktur . . . . .	5
2.2.2	Rapid Prototyping . . . . .	5
2.3	Implementierung . . . . .	6
2.3.1	Serverkonfiguration . . . . .	6
2.3.1.1	Installed Apps . . . . .	6
2.3.1.2	Static Files . . . . .	6
2.3.1.3	Datenbank . . . . .	7
2.3.2	Bootstrap Framework . . . . .	8
2.3.3	Benutzerverwaltung . . . . .	8
2.3.4	Session Management . . . . .	8
2.3.5	Feedparser Bibliothek . . . . .	9
2.3.5.1	RSS . . . . .	9
2.3.5.2	Atom . . . . .	9
2.3.6	Feed-Verwaltung . . . . .	9
2.3.7	Benutzer-Verwaltung . . . . .	10
2.3.8	Feed-Hauptansicht . . . . .	10
2.3.9	Feed-Eintrag . . . . .	12
2.3.10	Feed-Informationen . . . . .	12
2.3.11	Feed Suche . . . . .	13
<b>3</b>	<b>Fazit</b>	<b>15</b>
	<b>Literaturverzeichnis</b>	<b>16</b>

# 1 Einführung

## 1.1 Wahlpflichtfach: Agile Webanwendungen mit Python

Im Rahmen des Wahlpflichtfaches Agile Webanwendungen mit Python bei Herrn Erich Seifert wurden verschiedene Projekte, in kleinen Gruppen, im Sommersemester 2013 umgesetzt.

In der Vorlesung, dem theoretischen Teil des Wahlfaches, wurde uns ein guter Einblick in die verschiedenen Frameworks wie web2py<sup>1</sup>, Django<sup>2</sup> und Flask<sup>3</sup> gegeben. Zudem behandelten wir zahlreiche Grundlagen der Programmiersprache Python sowie Besonderheiten der Web-Entwicklung.

**Sämtliche Planungsphasen und Programmierarbeiten wurden mittels Pair Programming in Zusammenarbeit umgesetzt.**

## 1.2 Grundlagen des World Wide Webs

Für die Kommunikation zwischen dem Browser des Clients und dem Webserver ist das Hypertext Transfer Protocol (HTTP) zuständig. Der Nachrichtenaustausch erfolgt textbasiert über den von der IANA<sup>4</sup> festgelegten Port 80.

Der Austausch der Daten erfolgt nach dem Request-Response-Modell. Dabei stellt der Client eine Anfrage an den Server (Request) und erhält anschließend eine Antwort (Response) auf seine Anfrage.

Mit der Anfrage können zahlreiche Operationen durchgeführt werden. Folgende Übersicht stellt die wichtigsten da:

- GET: Operation für den Abruf von Ressource vom Server
- POST: Operation zur Übertragung von Informationen wie z. B. Formulardaten an den Server
- HEAD: HTTP-Header wird ohne dem eigentlichen Dokumentinhalt (Body) abgefragt

---

<sup>1</sup>**web2py** bezeichnen ein Open-Source-Framework für die agile Entwicklung von datenbankbasierten Webanwendungen. Dabei versucht es einen großen Teil der Webentwicklung abzudecken.

<sup>2</sup>**Django** bezeichnen ein in Python geschriebenes quelloffenes Web-Framework, welches dem Model-View-Controller-Schema folgt und in vielerlei Hinsicht das Python-Gegenstück zu Ruby on Rails ist.

<sup>3</sup>**Flask** bezeichnen ein in Python geschriebenes Micro-Framework zur Entwicklung von Webanwendungen. Hierbei liegt der Fokus in der Erweiterbarkeit und einer sehr guten Dokumentation.

<sup>4</sup>**IANA** (Internet Assigned Numbers Authority) bezeichnen eine Abteilung der ICANN, die für die Zuordnung von Nummern und Namen im Internet zuständig ist.

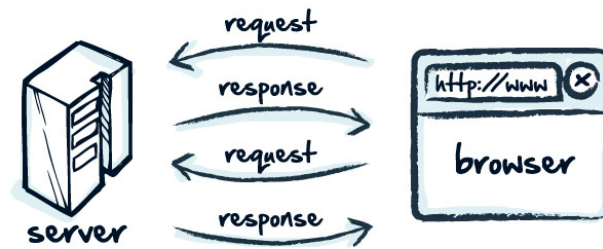


Abbildung 1.1: Request-Response-Modell (Quelle: <http://www.realtime.co>)

- PUT: Ressource wird unter Angabe der Ziel-URI<sup>5</sup> auf den Webserver hochgeladen
- DELETE: Ressource wird vom Server gelöscht (heute kaum implementiert)

Jede Anfrage durch den Client wird durch einen HTTP-Statuscode beantwortet. Zum Beispiel lassen sich daraus Informationen zum Erfolg oder Misserfolg eines Requests ermitteln. Die dreistelligen Werte lassen sich in folgende Kategorien einteilen: [Wong 2000]

- 1xx - Informationen
- 2xx - Erfolgreiche Operation
- 3xx - Umleitung
- 4xx - Client-Fehler
- 5xx - Server-Fehler

---

<sup>5</sup>URI (Uniform Resource Identifier) bezeichnet einen Identifikator, der zur Identifizierung einer abstrakten oder physischen Ressource dient. Eine URL ist eine Sonderform einer URI, die nach einem gewissen Schema folgt.

## 2 Projekt Rodisare

In unserem Projekt Rodisare haben wir uns für einen FeedReader zum Einlesen und Anzeigen von News Feeds entschieden. Mittels einer Authentifizierung kann sich der Benutzer über den Browser einloggen und anschließend die Kurznachrichten von verschiedenen Betreibern in den Formaten RSS<sup>6</sup> und Atom<sup>7</sup> abonnieren. Zudem soll eine Suchfunktionalität die Filterung der Feeds ermöglichen. Mittels dieser Anwendung lassen sich Programmier Techniken und Funktionen verschiedener Programmbibliotheken im Umfeld des Webs optimal nutzen.

### 2.1 Projektplanung

#### 2.1.1 Web-Framework

Bei der Entscheidung über das Entwicklungs-Framework fiel die Wahl auf Django. Dieses baut auf das MVT-Prinzip auf (Model View Template), eine abgewandelte Form des MVC-Prinzipes (Model View Controller). Die Views sind bei MVT ähnlich den Controllern und Templates können als Views ansehen werden. Django bietet eine umfangreiche Template-Sprache, einen Object-relational mapper, ein automatisch erzeugtes Admin-Interface und ein mit regulären Ausdrücken anpassbares URL-Design. [Jeff Forcier 2008]

#### 2.1.2 Versionsverwaltung

Um den Programmcode verteilt entwickeln zu können, haben wir uns für freie Versionsverwaltung Apache Subversion (SVN) entschieden. Nach der Installation auf einem eigenen virtuellen Server lässt sich das Repository mit dem Programm TortoiseSVN, welches sich in den Windows Explorer integriert, verwenden.

#### 2.1.3 Integrierte Entwicklungsumgebung

Wir entschieden uns für die integrierte Entwicklungsumgebung PyCharm der Firma JetBrains, welche Django unterstützt. Schlüssel-Features dieser IDE sind Refactoring sowie vielfältige Möglichkeiten zum automatischen Erstellen von Code. [Pycharm 2013]

---

<sup>6</sup>**RSS** bezeichnen ein Format für die strukturierte Veröffentlichung von Änderungen auf Websites im standardisierten XML-Format.

<sup>7</sup>**Atom** bezeichnen ein Format für den plattformunabhängigen Austausch von Informationen im HTML- und XML-Format und ermöglicht Webinhalte zu bearbeiten und zu verbreiten.

### 2.1.4 Modulplanung

In unsrer Planungsphase haben wir uns überlegt, welche Funktionen wir dem Benutzer mit unserem FeedReader bieten möchten und welche sich in der Zeit des Semesters realisieren lassen. In folgende Punkte haben wir das Projekt aufgeteilt, die anschließend Schrittweise umgesetzt wurden:

- Konfiguration des Web-Frameworks
- Erstellung der Datenbankstruktur
- Benutzerverwaltung (Registrierung und Anmeldung)
- Feeds von Webseiten parsen
- Front-End gestalten
- Verwaltungsseite für die Benutzer
- Session Management
- Suche in Feeds

### 2.1.5 Projektdokumentation

Als System für die Projektdokumentation bot sich uns LaTeX an. LaTeX ist ein Softwarepaket, das die Benutzung des Textsatzsystems TeX mit Hilfe von Makros vereinfacht. Anders als herkömmliche Textverarbeitungsprogramme basiert LaTeX nicht auf dem WYSIWYG-Prinzip<sup>8</sup>. Es wird mit Textdateien gearbeitet, in denen beispielsweise zu formatierende Passagen, Überschriften oder auch komplexe Formeln durch Befehle textuell definiert werden. Besonders für wissenschaftliche Dokumentationen wird bevorzugt LaTeX verwendet.

Mit LaTeX kann zudem sehr schnell ein weitaus eleganteres Erscheinungsbild erzeugt werden, als dies mit konventionellen Textverarbeitungsprogrammen der Fall ist.

---

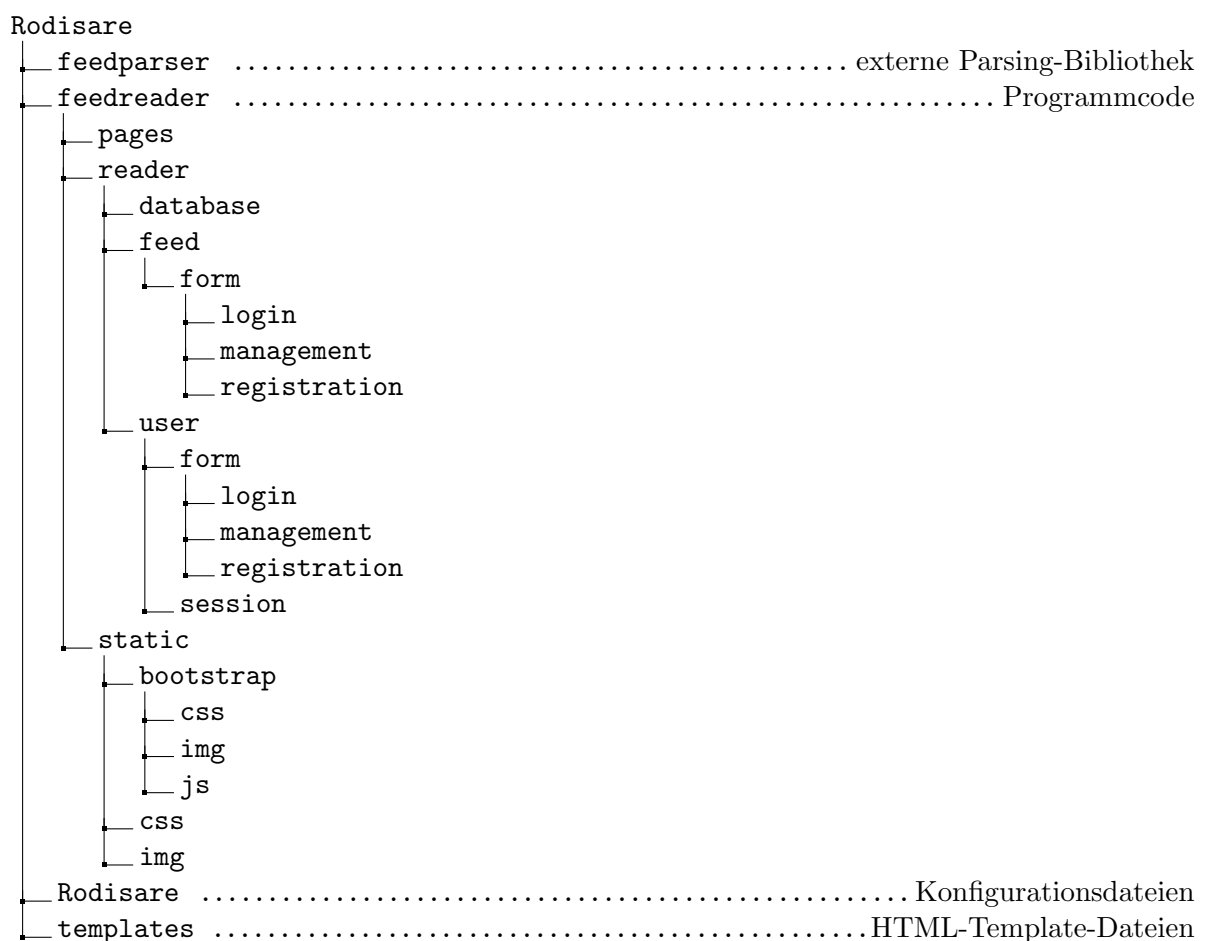
<sup>8</sup>**WYSIWYG** (What You See Is What You Get) bezeichnet die Anzeige eines Dokumentes während dem Bearbeiten, die wie die Ausgabe über ein anderes Gerät, z. B. einem Drucker, aussieht.

## 2.2 Entwurf

### 2.2.1 Projektstruktur

Nachdem das Projekt mittels PyCharm angelegt wurde, existierte bereits die grundlegende Verzeichnisstruktur. Erweitert wurde diese mit der externen Parsing-Bibliothek „feedparser“, den HTML-Template-Dateien, applikationsspezifische Konfigurationen, dem eigentlichen Programmcode, sowie Dateien für die grafische Gestaltung.

Ordnerstruktur:



### 2.2.2 Rapid Prototyping

Für die Softwareentwicklung im kleinen Team eignete sich für uns Rapid Prototyping, da diese Methode schnell zu ersten Ergebnissen führt und frühzeitiges Feedback bezüglich der Eignung eines Moduls als Lösungsansatz liefert. [[Gustav Pomberger 1993](#)]

Hierbei wurden als erstes zwei grundlegende Seiten erstellt. Eine für die schnelle Verwendung der Feedparser Bibliothek und die andere für die grafische Gestaltung mittels des Bootstrap-Frameworks [2.3.2](#).

## 2.3 Implementierung

### 2.3.1 Serverkonfiguration

Die Django Anwendung lässt sich sehr einfach in der Datei settings.py konfigurieren. Hierbei wurden folgende Einstellung von uns vorgenommen:

#### 2.3.1.1 Installed Apps

Damit sich diverse Module in unserer Anwendung verwenden lassen, müssen diese in den INSTALLED\_APPS angegeben werden. Hierzu gehört auch der FeedReader.

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    # Admin App
    'django.contrib.admin',
    # FeedReader App
    'feedReader',
)
```

Abbildung 2.1: Installierte Module im FeedReader

#### 2.3.1.2 Static Files

Damit wir in unserer Anwendung weitere Dateien wie Bilder, JavaScript oder CSS verwenden können, wird das django.contrib.staticfiles zu den INSTALLED\_APPS hinzugefügt. Anschließend können wir die STATIC\_URL definieren, welche das entsprechende Verzeichnis referenziert. Zuletzt muss im Template in der ersten Zeile {% load staticfiles %} angegeben werden, damit sich auf dieser die Dateien einbinden lassen.

```
<link href="{% static 'bootstrap/css/bootstrap-responsive.css' %}" rel="stylesheet">
<link href="{% static 'css/styleshet.css' %}" rel="stylesheet">
```

Abbildung 2.2: Verwendung von Dateien aus dem Static Verzeichnis



### 2.3.1.3 Datenbank

Die Grundlegenden Einstellungen der verwendeten Datenbank werden in der Serverkonfiguration [2.3.1](#) abgelegt. Da wir uns aufgrund der Einfachheit für eine einfache SQLite Datenbank [[Jeff Forcier 2008](#)] entschieden haben, ist keine aufwändige Anpassung nötig.

```
1 DATABASES = {  
2     'default': {  
3         'ENGINE': 'django.db.backends.sqlite3',  
4         'NAME': 'rodissare.db',  
5         # The following settings are not used with sqlite3:  
6         'USER': '',  
7         'PASSWORD': '',  
8         'HOST': '',  
9         'PORT': '',  
10    }  
11 }
```

Abbildung 2.3: Konfiguration der verwendeten SQLite Datenbank

Zu den bereits vorhandenen Tabellen der Administrationsseite wurde zudem noch eine einfache Tabelle für die Ablage der Feeds des jeweiligen Benutzers angelegt.

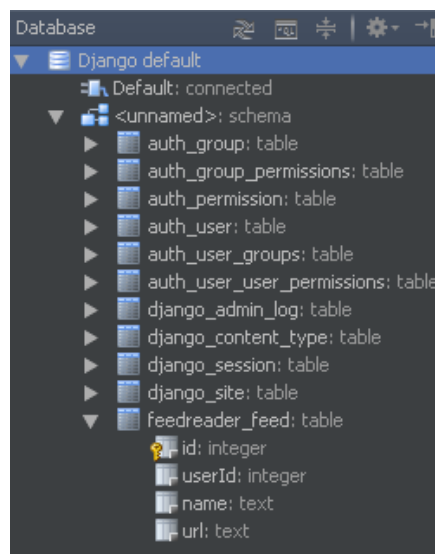


Abbildung 2.4: Konfiguration der verwendeten SQLite Datenbank

Das schöne bei der Verwendung von Datenbank in Django ist der einfache Umgang mit Object-relational mapper, die einem die Datensätze in Python Objekte zurück geben. Somit lassen sich diese einfach im Programmcode verarbeiten.

### 2.3.2 Bootstrap Framework

Das Bootstrap Framework ist eine freie Sammlung von Hilfsmitteln für die Gestaltung von Websites und Webanwendungen. Es enthält verschiedene HTML und CSS<sup>9</sup> basierende Gestaltungsvorlagen, die uns eine Gestaltung im Bereich des Front-Ends<sup>10</sup> erheblich erleichtern. [amdo 2013]

Dem Benutzer stehen hierbei verschiedenste Komponenten wie DropDowns, Buttons, Navigationslisten oder auch Thumbnails zur Verfügung, die sich einfach einbinden lassen. Für deren Verwendung müssen lediglich das Stylesheet sowie die Scriptdatei in der HTML-Seite eingebunden werden. Zudem muss für DOM<sup>11</sup>-Manipulationen und -Navigation die JavaScript-Bibliothek jQuery<sup>12</sup> verwendet werden.

### 2.3.3 Benutzerverwaltung

Das Django Framework bietet ein leistungsstarkes Admin Modul, welches grundlegende Funktionalitäten wie die Benutzerverwaltung oder die Verwendung von Sessions bereitstellt. Mit der Einrichtung des Projektes in Pycharm wurden uns bereits alle Abhängigkeiten wie Datenbanktabellen und URLs angelegt.

### 2.3.4 Session Management

Wie bereits in 2.3.3 erwähnt wurde, ermöglicht das Admin Modul eine einfache und effiziente Verwendung von Sessions. In unserer Anwendung lässt sich somit optimal die aktuelle Sitzung eines angemeldeten Benutzers speichern.

Sessions werden zum einen in der Datenbank gespeichert und zum anderen als Cookie auf dem Client. Die Dauer einer Session bzw. die Gültigkeit des Cookies wurde von uns auf eine Woche festgelegt.

Bei aktiver Nutzung der Webseite wird diese vor Ablauf der Session verlängert. Somit muss der Benutzer sich nicht jede Woche neu authentifizieren.

---

<sup>9</sup> **CSS** (Cascading Style Sheets) bezeichnet eine deklarative Sprache für Stilvorlagen von strukturierten Dokumenten. Mit ihr können Farben, Layout, Schrifteigenschaften usw. auf einer Webseite gestaltet werden.

<sup>10</sup> **Front-End** bezeichnen in der Informationstechnik die Schicht, welche sich näher am Benutzer befindet wie z. B. die grafischen Benutzeroberfläche.

<sup>11</sup> **DOM** (Document Object Model) bezeichnet eine Spezifikation einer Schnittstelle für den Zugriff auf HTML- oder XML-Dokumente.

<sup>12</sup> **jQuery** ist eine freie, umfangreiche JavaScript-Bibliothek, welche komfortable Funktionen zur DOM-Manipulation und -Navigation zur Verfügung stellt.

### 2.3.5 Feedparser Bibliothek

Um beliebige RSS- und Atom-Feeds in objektbezogene Daten umzuwandeln, entschieden wir uns für eine externe Bibliothek, die diese Funktionalitäten bereit stellt. Wir wählten die Bibliothek „feedparser“<sup>13</sup>, da sich diese einfach und intuitiv verwenden lässt. [Feedparser 2012]

#### 2.3.5.1 RSS

RSS<sup>14</sup> ist eine seit dem Anfang des Jahres 2000 kontinuierlich weiterentwickelte Familie von Formaten für die einfache und strukturierte Veröffentlichung von Neuigkeiten auf Websites in einem standardisierten Format (XML). RSS-Dienste werden in der Regel in Form spezieller Service-Websites (sogenannter RSS-Channels) angeboten. Ein RSS-Channel versorgt den Adressaten oft, ähnlich einem Nachrichtenticker, mit kurzen Informationsblöcken, die aus einer Schlagzeile mit kurzem Textanriss und einem Link zur Originalseite bestehen. Zunehmend werden aber auch komplette Inhalte klassischer Webangebote ergänzend als Volltext-RSS bereitgestellt. Die Bereitstellung von Daten im RSS-Format bezeichnet man auch als RSS-Feed<sup>15</sup>. Wenn ein Benutzer einen RSS-Channel abonniert hat, so sucht der Client in regelmäßigen Abständen beim Server nach Aktualisierungen im RSS-Feed.

#### 2.3.5.2 Atom

Das Atom Syndication Format (ASF) ist ein XML-Format für den plattformunabhängigen Austausch von Informationen. Es ist die am häufigsten anzutreffende Umsetzung der Atom-Standards. ASF wird entwickelt, um die Nachfolge von RSS antreten und ist damit die moderne Form eines Newsletters.

Atom entstand aus dem Bedürfnis heraus, die Vorteile der unterschiedlichen RSS-Formate in einem neuen Format zusammenzufassen und um neue Elemente zu ergänzen. Dabei haben die Entwickler, in überwiegender Mehrzahl Blogger, ASF auch so gestaltet, um den speziellen Bedürfnissen von Weblogs und Nachrichtenseiten gerecht zu werden. Die maßgeblichen Unterstützer von Atom sind in der Industrievereinigung AtomEnabled Alliance organisiert.





### 2.3.6 Feed-Verwaltung

In der Rodisare-Oberfläche ist es für den Benutzer über die Feed-Verwaltung möglich, seine abonnierten Feeds als Auflistung zu sehen oder Feeds per Eingabe der URL hinzuzufügen und zu entfernen. Jeder Benutzer hat dabei lediglich Zugriff auf seine persönlichen Feeds.

<sup>13</sup>feedparser ist eine Python-Bibliothek, die alle gängigen Feed-Formate parsen kann.  
<http://code.google.com/p/feedparser/>

<sup>14</sup>engl. ursprünglich Rich Site Summary, später Really Simple Syndication

<sup>15</sup>englisch to feed: im Sinne von füttern, einspeisen, zuführen

Name	URL	
Spiegel.de	http://www.spiegel.de/schlagzeilen/tops/index.rss	
xda-developers.com	http://www.xda-developers.com/tag/all-android/feed/	
gizmodo	http://www.gizmodo.de/feed	
Golem Atom1.0	http://rss.golem.de/rss.php?feed=ATOM1.0	

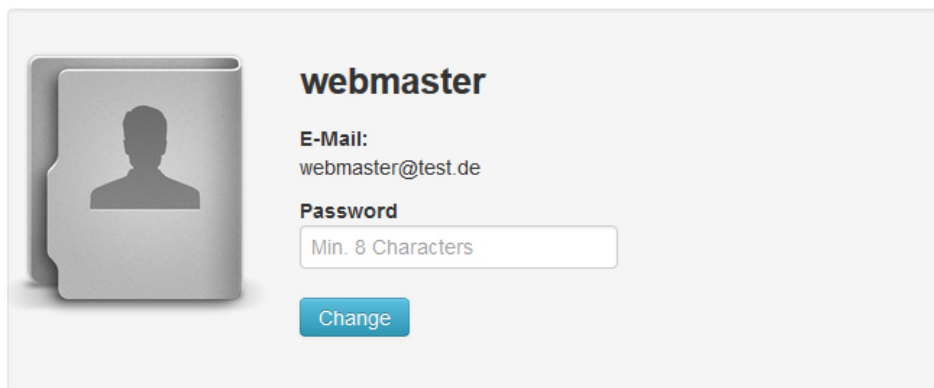
<input type="text" value="Feed Name"/>	<input type="text" value="Feed URL"/>	<input type="button" value="Add"/>
--	---------------------------------------	------------------------------------

Abbildung 2.5: Feed-Verwaltung

### 2.3.7 Benutzer-Verwaltung

In der Verwaltung ist es für einen Benutzer ebenfalls möglich, sein Passwort zu ändern.

#### User



**webmaster**

**E-Mail:**  
webmaster@test.de

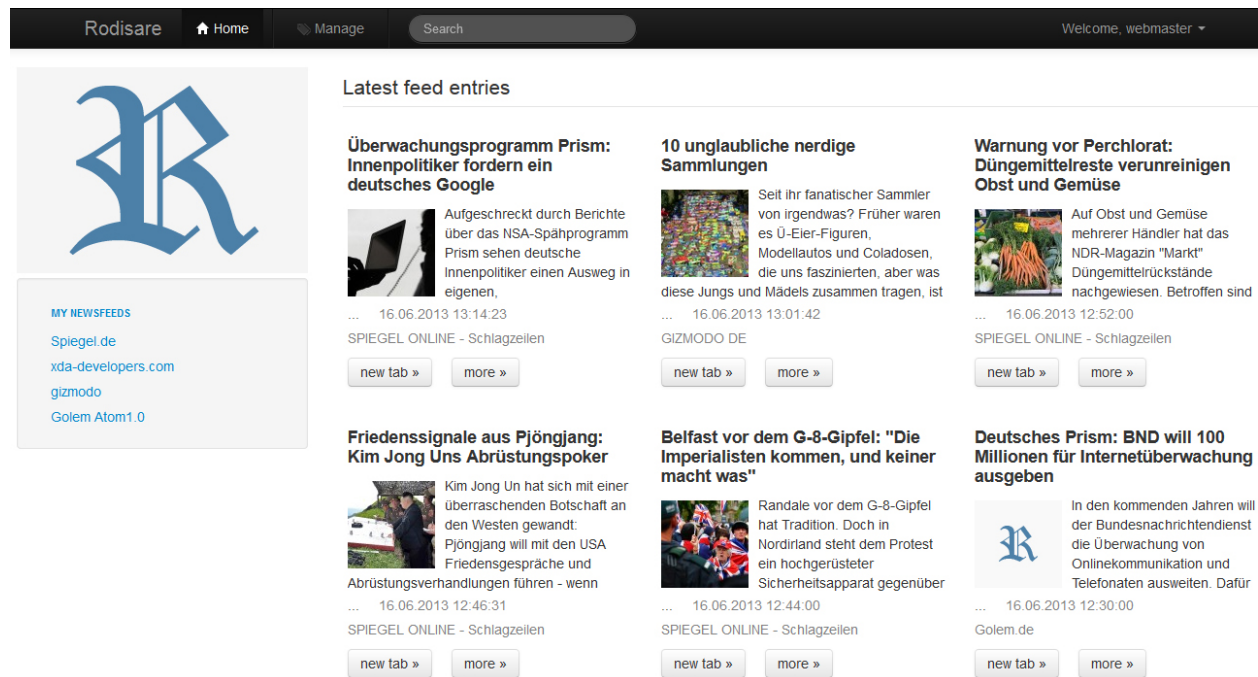
**Password**

Abbildung 2.6: Änderung des Benutzerpasswortes

### 2.3.8 Feed-Hauptansicht

Sofern ein Benutzer mindestens eine Feed-URL angegeben hat, werden auf der Rodisare-Startseite Einträge aus allen Feeds angezeigt.

In diesem Beispiel werden Einträge aus vier verschiedenen Feeds angezeigt, absteigend sortiert nach Datum. In der Standardeinstellung werden nur die 18 neuesten Einträge aus allen Feeds zusammen angezeigt.



The screenshot shows the Rodisare website interface. At the top is a navigation bar with 'Rodisare', 'Home', 'Manage', and a search bar. Below this is a large blue 'R' logo. To the left of the main feed is a sidebar titled 'MY NEWSFEEDS' containing links to 'Spiegel.de', 'xda-developers.com', 'gizmodo', and 'Golem Atom1.0'. The main area is titled 'Latest feed entries' and displays a grid of news items. Each item includes a title, a thumbnail image, a brief description, a timestamp, the source, and buttons for 'new tab' and 'more'.

**Latest feed entries**

**Überwachungsprogramm Prism: Innenpolitiker fordern ein deutsches Google**  
Aufgeschreckt durch Berichte über das NSA-Spähprogramm Prism sehen deutsche Innenpolitiker einen Ausweg in eigenen,  
... 16.06.2013 13:14:23  
SPIEGEL ONLINE - Schlagzeilen  
new tab » more »

**10 unglaubliche nerdige Sammlungen**  
Seit ihr fanatischer Sammler von irgendwas? Früher waren es Ü-Eier-Figuren, Modellautos und Coladosen, die uns faszinierten, aber was diese Jungs und Mädels zusammen tragen, ist  
... 16.06.2013 13:01:42  
GIZMODO DE  
new tab » more »

**Warnung vor Perchlorat: Düngemittelreste verunreinigen Obst und Gemüse**  
Auf Obst und Gemüse mehrerer Händler hat das NDR-Magazin "Markt" Düngemittelrückstände nachgewiesen. Betroffen sind  
... 16.06.2013 12:52:00  
SPIEGEL ONLINE - Schlagzeilen  
new tab » more »

**Friedenssignale aus Pjöngjang: Kim Jong Uns Abrüstungspoker**  
Kim Jong Un hat sich mit einer überraschenden Botschaft an den Westen gewandt: Pjöngjang will mit den USA Friedensgespräche und Abrüstungsverhandlungen führen - wenn  
... 16.06.2013 12:46:31  
SPIEGEL ONLINE - Schlagzeilen  
new tab » more »

**Belfast vor dem G-8-Gipfel: "Die Imperialisten kommen, und keiner macht was"**  
Randalie vor dem G-8-Gipfel hat Tradition. Doch in Nordirland steht dem Protest ein hochgerüsteter Sicherheitsapparat gegenüber  
... 16.06.2013 12:44:00  
SPIEGEL ONLINE - Schlagzeilen  
new tab » more »

**Deutsches Prism: BND will 100 Millionen für Internetüberwachung ausgeben**  
In den kommenden Jahren will der Bundesnachrichtendienst die Überwachung von Onlinekommunikation und Telefonaten ausweiten. Dafür  
... 16.06.2013 12:30:00  
Golem.de  
new tab » more »

Abbildung 2.7: Startseiten-Ansicht

Zu jedem Feed-Eintrag wird der Name des Feeds sowie das Veröffentlichungs-Datum angezeigt. Mit je zwei Schaltflächen hat man die Möglichkeit, den abgeschnittenen Text weiter zu lesen oder den Link zu der Webseite dieses Eintrages in einem neuen Tab zu öffnen. Des Weiteren wird zu jedem Feed-Eintrag das übermittelte Miniaturbild angezeigt, sofern vorhanden. Andernfalls wird eine Standardgrafik als Platzhalter eingeblendet.


### 2.3.9 Feed-Eintrag

In der Detailansicht zu einem Feed-Eintrag werden weitere Details angezeigt, beispielsweise der Name des Autors oder ein Link zu Kommentaren zu dem Artikel. Zudem hat man hier die Möglichkeit, detailliertere Informationen zu dem beinhaltenden Feed oder dessen gesamten Einträge anzeigen zu lassen.

#### Google will weltweites Internet mit Ballons bereitstellen [Video]

Author	Johannes Gräßler
Comments	<a href="http://www.gizmodo.de/2013/06/15/google-will-weltweites-internet-mit-ballons-bereitstellen-video.html#comments">http://www.gizmodo.de/2013/06/15/google-will-weltweites-internet-mit-ballons-bereitstellen-video.html#comments</a>



Wenn man durch die weite Wüste wandert oder einen Berg erklimmt, sich selbst am Südpol, auf einer abgelegenen Farm oder sonst irgendwie im Nirgendwo wiederfindet, was ist dann? Genau, ihr habt kein Internet! Allerdings berichtet Wired, dass Google das ändern möchte, indem dort Ballons in den Himmel geschickt und so der ganzen Welt drahtlosen Internetzugriff gegeben werden soll. Was zum...?! mehr...

GIZMODO DE (15.06.2013 10:42:11)

[Read on in a new tab »](#)
[View feed information »](#)
[View all entries of this feed »](#)

Abbildung 2.8: Detailansicht eines Feed-Eintrages

### 2.3.10 Feed-Informationen

In dieser Ansicht werden die wichtigsten Informationen über einen Feed detailliert aufgelistet. Dies sind der Titel, Untertitel und die URL zum Feed. Des Weiteren wird die Anzahl der Einträge und die verwendete RSS / Atom-Version angezeigt. In dieser Übersicht werden zudem die Titel aller vorhandenen Feed-Einträge aufgelistet.

## GIZMODO DE

[View complete feed »](#)

Das Gadget-Weblog

## Detailed feed information

Title	GIZMODO DE
Subtitle	Das Gadget-Weblog
URL	<a href="http://www.gizmodo.de/feed">http://www.gizmodo.de/feed</a>
Numer of entries	10
Version	rss20
Current entries	<p>10 unglaubliche nerdige Sammlungen</p> <p>Nur echt mit 2.989 Steinen: Lego bringt Sydney Opera House als Creator-Expert-Model</p> <p>Laser-Technologie sei Dank: Archäologen finden in Kambodscha verborgene Tempel-Stadt</p> <p>Air-Hockey-Roboter aus Japan zockt euch ab. Immer.</p> <p>We See In Every Direction: Chaotischer Browser, der all seinen Nutzern die Kontrolle gibt – gleichzeitig</p>

Abbildung 2.9: Detaillierte Informationen zu einem Feed

### 2.3.11 Feed Suche

Angemeldete Benutzer haben zudem die Möglichkeit sich nur Nachrichten, die gewissen Schlagworte beinhalten, anzeigen zu lassen. Hierfür wird die Suche in der oberen Menüleiste verwendet.



Abbildung 2.10: Eingabefeld der Suche

Nach der Eingabe der Suchbegriffe (UND-Verknüpft) findet eine Weiterleitung auf die für die Suche angepasste Anzeige statt. Die Feeds, in denen alle Suchbegriffe in der Überschrift oder im Text vorkommen werden dort aufgelistet.

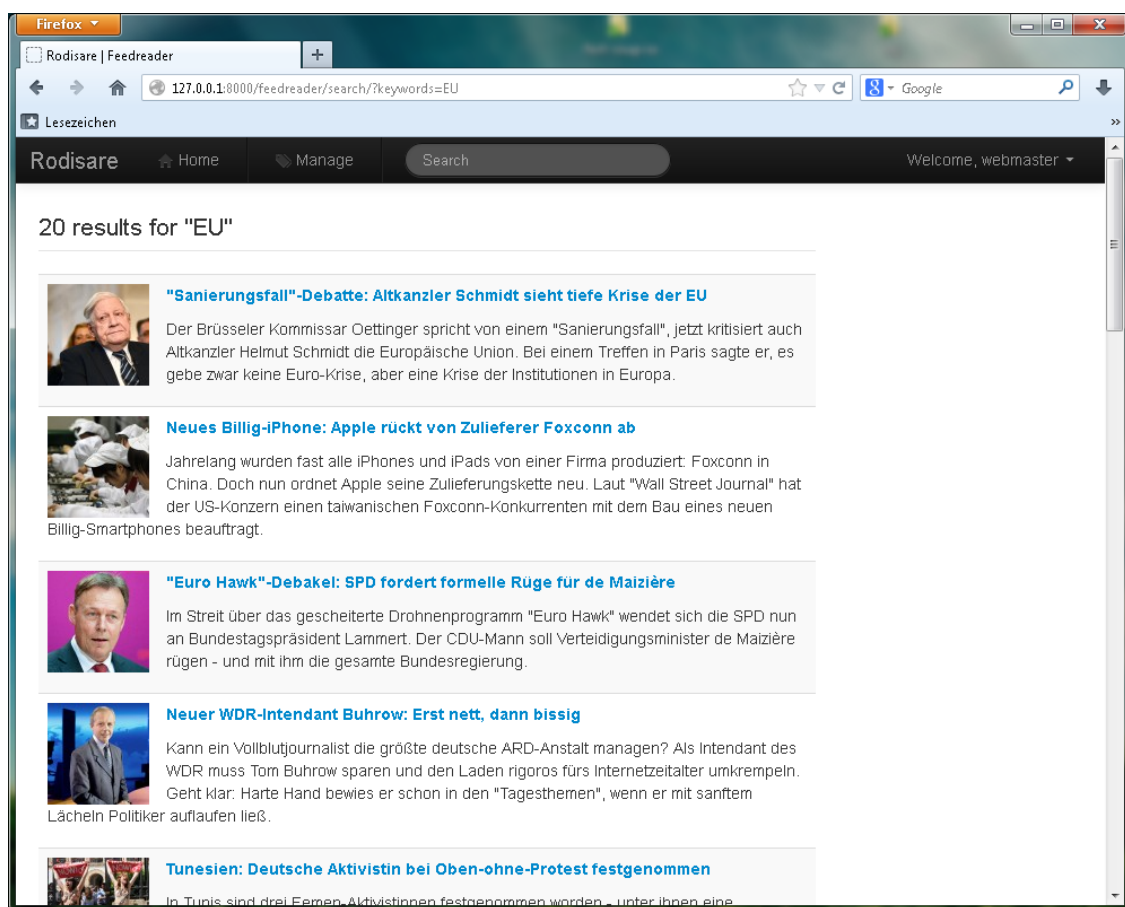


Abbildung 2.11: Suchergebnis



## 3 Fazit

Im Rahmen unserer Projektarbeit des Wahlpflichtfaches Agile Webanwendungen mit Python konnten wir unser erlerntes theoretisches Wissen aus der Vorlesung sowie die Grundlagen aus Programmieren 3 anwenden, vertiefen und an einem Projekt in der Praxis umsetzen. Neben bereits bekannten Techniken und Vorgehensweisen wurden uns weitere näher gebracht und uns ein guter Einblick in die Thematik Agile Webanwendungen mit Python gegeben. In unserem Projekt verwendeten wir zudem einige Programmierparadigmen, die in der Theorie nicht immer ganz selbstverständlich waren, sich jedoch in der Praxis bewährten. Das Resultat unseres Projektes bietet dem Benutzer grundlegende Funktionalitäten und könnte bereits produktiv eingesetzt werden.

## Literaturverzeichnis

- [Feedparser 2012] FEEDPARSER: *Python Feedparser Library*. <http://code.google.com/p/feedparser>, December 2012. – [Version 5.1.3; Stand: 9.04.2013] 2.3.5
- [Gustav Pomberger 1993] GUSTAV POMBERGER, Guenther B.: *Grundlagen des Software-Engineering (Prototyping und objektorientierte Software-Entwicklung)*. 1993. – ISBN 3-446-16262-3 2.2.2
- [Jeff Forcier 2008] JEFF FORCIER, Wesley C. Paul Bissex B. Paul Bissex: *Python Web Development with Django*. 2008. – ISBN 978-0132356138 2.1.1, 2.3.1.3
- [@mdo 2013] @MDO, @fat: *Sleek, intuitive, and powerful front-end framework for faster and easier web development*. <http://twitter.github.io/bootstrap>, 2013. – [Version 2.3.2; Stand: 02.05.2013] 2.3.2
- [Pycharm 2013] PYCHARM: *Quick Start Guide: PyCharm as a Django development tool*. [http://www.jetbrains.com/pycharm/quickstart/django\\_guide.html](http://www.jetbrains.com/pycharm/quickstart/django_guide.html), 2013. – [Version 2.7.3; Stand: 22.06.2013] 2.1.3
- [Wong 2000] WONG, Clinton: *HTTP, kurz & gut*. 2000 1.2