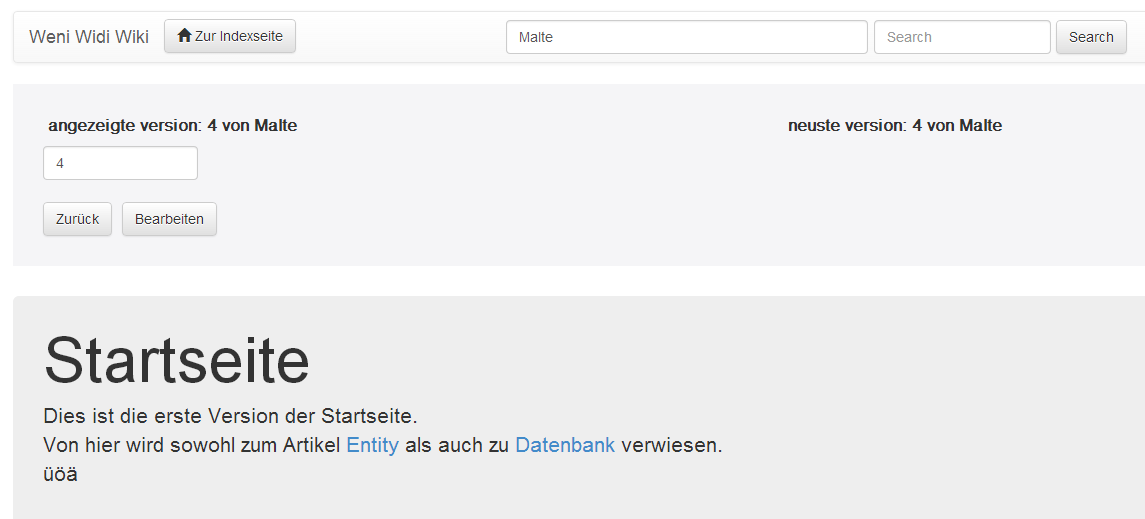
# Lizenz

Das Projekt wurde unter der Apache Lizenz 2.0 lizensiert.  
Diese lässt sich im Projekt auch unter *./lizenz/lizenz.html* abrufen:

Copyright 2013 Michael Bertram, Malte Weiß  
  
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at  
  
http://www.apache.org/licenses/LICENSE-2.0  
  
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

## Aufbau Die show.xhtml

Die *show.xhtml* zeigt Wiki-Themen an, und bietet die Möglichkeit, ältere Versionen anzuzeigen.



3

2

1

Grafik 1: show.xhtml

Über eine Textbox (1) kann ein Nutzername eingetragen werden.  
Dieser wird in einer Instanz von *ThemenView* gespeichert. *ThemenView* hat einen *SessionScope* was bedeutet, dass nur eine Instanz pro Nutzer-Session existiert. Der Nutzer-Name wird also dauerhaft in *ThemenView* gespeichert und kann von anderen Seiten aus wieder aufgegriffen werden.

Wenn ein Name eingetragen wird, wird der Bearbeiten-Button (2) angezeigt. Dies geschieht dynamisch über Ajax, sodass nur ein kleiner Teil der Seite neugeladen werden muss.

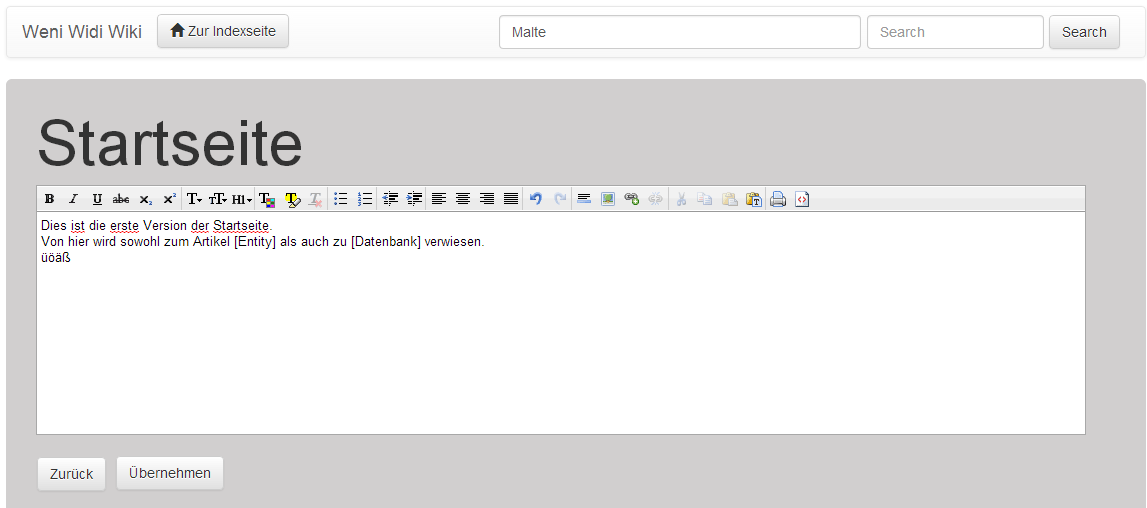
Über den Bearbeiten-Button erfolgt eine Weiterleitung zur *edit.xhtml*, wo das ausgewählte Thema bearbeitet werden kann.

Im selben Bereich befindet sich auch ein Eingabefeld, über das die angezeigte Version eines Themas bestimmt werden kann. Die Eingaben sind durch einen Validator abgesichert, der zusichert, dass nur Zahlen im Bereich von 0 bis zur aktuellsten Versionsnummer des Themas weiterverarbeitet werden.  
Bei einer Eingabe wird das Thema (3) über Ajax neugeladen, sodass ein Bestätigen-Button an dieser Stelle nicht erforderlich ist.

Die Anzeige der Verlinkungen in einem Thema (3) wurde durch einen Converter gelöst. In diesem werden, mittels eines Regulären-Audrucks , alle Schlagwörter in eckigen Klammern durch HTML-Links ersetzt, die zur *show.xhtml* weiterleiten und den Namen des Themas als Parameter übergeben. Eine Verlinkung könnte beispielsweise so aussehen: *./show.xhtml?thema=Startseite*  
Die Verwaltung der Daten geschieht auf der *show.xhtml* über die Klasse *ThemaShow*. Diese ist *ViewScoped*, wird also nur beim Neu laden oder Aufruf der Seite neu erstellt.  
Die Klasse verwaltet ein Entity-Objekt des angezeigten Themas sowie die Versionsnummern und Autoren auf der Seite.

### Die edit.xhtml

Auf der edit.xhtml können Themen bearbeitet werden. Änderungen werden als neue Versionen abgespeichert.



2

1

Grafik 2: edit.xhtml

Als Eingabefeld dient der Editor (1) von *Primefaces*. Dieser bietet zusätzliche Möglichkeiten zum Formatieren des Textes, erhöht allerdings auch die Größe des Texts in der Datenbank, da dieser als HTML abgespeichert wird.  
Eine neue Version kann über den Übernehmen-Button erstellt werden. Arbeiten mehrere Nutzer gleichzeitig an einer neuen Version, werden diese beim Abspeichern in der richtigen Reihenfolge übernommen, sodass keine Beiträge verloren gehen.  
Desweiteren werden die Eingaben des Nutzers vor dem Übernehmen durch einen Validator überprüft.  
Zum einen überprüft dieser, ob ein Nutzername eingegeben wurde, da eine Version immer einen Autor haben muss. Zum anderen werden die Klammerungsregeln überprüft.  
Diese wurden zuerst durch einen Regulären Ausdruck validiert. Bei längeren Themen kam es allerdings zu einer StackOverflowException, sodass wir eine eigene Lösung ausprogrammiert haben.

Die Daten-Verwaltung der *edit.xhtml* Seite geschieht über die Klasse *ThemaEdit* die einen *RequestScope* hat. Bei Instanziierung liest diese den Namen des Themas sowie die ausgewählte Versionsnummer aus dem ThemenView aus und kann damit das zu bearbeitende Thema aus der Datenbank laden.

# Erweiterungen

Neben kleinen Erweiterungen wie die Verwendung von Ajax und Primefaces haben wir zwei weitere Funktionen eingebaut und Bootstrap für die Darstellung der Benutzeroberfläche verwendet.

## Die index.xhtml



Grafik 3: index.xhtml

Auf der index.xhtml befindet sich eine Übersicht der n letzten Änderungen in Themen. Die Anzahl der Zeilen der Tabelle lassen sich über das Eingabefeld bestimmen.  
Bei einer Eingabe wird die Tabelle über Ajax neugeladen. Über den Aktualisieren-Button lässt sich die Seite Neu laden, sodass die Datenbank erneut ausgelesen wird.  
Über den Button „Zur Startseite“ kommt der Nutzer auf die *show.xhtml* mit dem Thema „Startseite“.

Die Daten werden über die Klasse *ThemaIndex* verwaltet, welche einen *RequestScope* hat.  
Sie speichert die Anzahl der anzuzeigenden Themen sowie eine Liste der aktuellsten Themen.

## Die search.xhtml

Im Header befinden sich ein Eingabefeld sowie ein Suchen-Button, über die Such-Anfragen gestellt werden können.  
  


Grafik 4: search.xhtml

Damit von jeder Seite aus Suchanfragen gestellt werden können, muss die *ThemaSearch* Instanz mindestens einen *SessionScope* haben.  
An dieser Stelle wäre auch ein *ApplicationScope* denkbar. Allerdings würden die Nutzer dann die Suchanfragen der jeweils anderen Nutzer sehen, weswegen wir uns dagegen entschieden haben.

Bei einer Suche können mehrere Fälle auftreten, die verschieden behandelt werden:  
Ist der Suchbegriff in genau einem Namen eines Themas enthalten, erfolgt eine direkte Weiterleitung zu dem Thema.  
Wurden keine oder mehr als eine Übereinstimmung gefunden, wird zur *search.xhtml* weitergeleitet.  
Dort wird dann entweder eine Liste der gefundenen Themen angezeigt, oder ein Hinweis, dass die Suche keine Treffer erzielt hat.