



PROJEKTBERICHT

Web Engineering - KI Newsboard

Modulares webbasiertes System zur Anzeige der Suchergebnisse



FH Bielefeld
University of
Applied Sciences

Cem Basoglu, Kevin Schima, Azad Sino, Veronika Sliesarenko

Inhaltsverzeichnis

Inhaltsverzeichnis	1
Projektbeschreibung	3
Vorstellung der Gruppenmitglieder und ihre Rollen im Team	4
Cem Basoglu	4
Kevin Schima.....	4
Azad Sino	5
Veronika Sliesarenko.....	5
Theoretische Grundlagen.....	5
Sentimentanalyse:	5
Crawler	6
Indexierung	6
Volltextsuche	6
Tokenizer	7
Authentifizierungstoken	7
Konzeptionelle Arbeiten	8
Use-Case-Diagramme	8
Sequenzdiagramme	10
Software-Architektur	12
ER-Modell	13
Projektverlauf	16
Treffen 1, 22.10.2016.....	16
Treffen 2, 29.10.2016.....	17
Treffen 3, 03.11.2016.....	18
Treffen 4, 10.11.2016	19
Treffen 5, 17.11.2016.....	20
Treffen 6, 24.11.2016.....	21
Treffen 7, 01.12.2016	22

Treffen 8, 08.12.2016.....	24
Implementierung	25
Middle-tier	25
Client-tier	29
Ausblick	30
Fazit	31
Installationshinweise und Benutzerhandbuch	32
Handbuch.....	33
Administration	38
Anlagen	44
Teamarbeitsplan	44
Klassendiagramm.....	48

Projektbeschreibung

In dem Projekt geht es darum, ein modulares webbasiertes System zur Anzeige von FH-bezogenen Neuigkeiten zu entwickeln, welche auf verschiedenen Oberflächen angezeigt werden sollen. Dazu zählen ein öffentlicher Monitor, welcher im Neubau des Campus Minden aufgestellt wird, Web- und mobile Applikation. Im Gegensatz zu der Web- und mobilen Applikation, soll der öffentliche Monitor über keine Benutzerinteraktionen verfügen. Mit Benutzerinteraktionen sind ein Feedback zu einem einzelnen Beitrag und die Suche nach einzelnen Beiträgen über (fest/vorgegebene) Stichwörter gemeint.

Mit Hilfe von Web-Crawler und Algorithmen der künstlichen Intelligenz, im Folgenden nur noch KI-Algorithmen genannt, soll die Möglichkeit bestehen, Beiträge aus dem Internet zu aggregieren. Die so gesammelten Dokumente werden im Anschluss in einer Datenbank gespeichert und durch verschiedene Analysemodule, z.B. der Sentimentanalyse auf Dokument- oder Satzebene, bewertet werden.

Außerdem sollen diverse Detail-Stufen der Anzeige von Dokumenten erstellt werden. Es soll also möglich sein, sich einen Überblick über ein komplettes Dokument mit Eigenschaften und über einzelne Sätze mit Eigenschaften darstellen zu lassen.

Die Software soll einen modularen Aufbau besitzen. So ist es später einfach über die Schnittstelle möglich sein, eigene Crawler und KI-Algorithmen zu integrieren.

Vorstellung der Gruppenmitglieder und ihre Rollen im Team

Cem Basoglu

Rollen	Beschreibung
Project Lead	Zu Cems Aufgaben gehörten in der Rolle als Project Lead die Zuteilung von Zuständigkeiten anhand der Projektplanung und die Kontrolle der Zielerreichung für die jeweiligen Aufgabenteile.
Middle Tier	Im Middle Tier Bereich war Cem zum einen für die Implementierung der WebServices zuständig und zum anderen für die JAX-RS Implementierung zuständig um der Webservice REST Schnittstelle Java Ressourcen zuzuordnen. Außerdem für die Implementierung der Bean Validation API für die einfache Validierung korrekter Typen. Zusätzlich zeigte er sich für das ExceptionMapping verantwortlich für die korrekte Abwicklung von Laufzeitfehlern.
Backend Tier	Im Backend Tier war Cem auch für das Design und die Konzeptionierung der Datenbank zuständig Kommunikation zu Middleware Schicht

Kevin Schima

Rollen	Beschreibung
Frontend Tier	Zu Kevins Aufgabenbereich im Frontend Tier gehörte das Design der verschiedenen Ansichten für den öffentlichen Monitor zum einen und die responsive Web- / mobile Ansicht.
Middle Tier	

Azad Sino

Rollen	Beschreibung
Middle Tier	Zu Azads Tätigkeitsbereich gehörte überwiegend die Abdeckung aller Entitäten in der JPA Datenzugriffsschicht und Umsetzung aller Use-Cases in der Geschäftslogikschicht

Veronika Sliesarenko

Rollen	Beschreibung
Frontend Tier	Zu Veronikas Aufgaben gehörte unter anderem das Anlegen von Views innerhalb der Administratoransicht für die Verwaltung von Crawlern und Analyzern

Theoretische Grundlagen

In diesem Kapitel werden Begriffe geklärt, die für das Verständnis der Software und des Projektes notwendig sind.

Sentimentanalyse:

In der Künstlichen Intelligenz ist die Sentimentanalyse ein Teilgebiet des Text Minings, welche sich damit beschäftigt, Texte automatisch auszuwerten. Dabei steht das Ziel im Fokus, die mit dem Text geäußerte Haltung als positiv oder negativ zu erkennen.

Bei der Sentimentanalyse werden zwei verschiedene Vorgehen unterschieden.

Zum einen existiert ein linguistisches Verfahren, welches mit Hilfe vordefinierter Listen über positive und negative Wörter und der Häufigkeit der Vorkommnisse dieser Wörter in einem Text eine positive oder negative Tendenz mit dem Text verbindet.

Die zweite Technik ist ein Verfahren, welches auf das Kapitel „Maschinelles Lernen“ der Künstlichen Intelligenz basiert. Im Gegensatz zum linguistischen Verfahren, werden beim zweiten Vorgehen vorgefertigte Listen, auch Trainingsdaten genannt, dazu genutzt, um selbstständig die Tendenz, denen Begriffe zugeordnet sind, zu erlernen.

Analyzer

Ein Analyzer ist eine Implementierung einer Sentimentanalyse. Durchgeführt wird dies im Rahmen des Masterstudiengangs KI. Die Software der Teilnehmer des Masterstudiengangs soll die durch die Schnittstellen bereitgestellten Beiträge analysieren und die Analyseergebnisse veröffentlichen.

In diesem Projekt werden lediglich Beispielanalyzer angelegt, um die Funktionalität der Software zu prüfen.

Crawler

Ein Crawler, oft auch als Webcrawler bezeichnet, ist ein Computerprogramm, das automatisch und kontinuierlich das World Wide Web nach Dokumenten durchsucht. Dabei arbeitet ein Crawler mit Hilfe eines Indexes, um so bestimmte Suchbegriffe unter den Ergebnisseiten aufzulisten. Diese Methode wird Indexierung genannt.

In dem Projekt werden Beispielcrawler dazu genutzt, automatisiert, d.h. in bestimmten Zeitabständen, das Internet nach FH-bezogenen Beiträgen zu durchsuchen und letztlich auf den verschiedenen Weboberflächen anzuzeigen. In Zukunft sollen diese Beispielcrawler durch Crawler ersetzt werden, die im Rahmen des Masterstudiengangs KI entwickelt werden.

Indexierung

Indexierung, häufig auch Verschlagwortung genannt, ist ein Verfahren, welches dazu genutzt wird, Schlagwörter einem Dokument zuzuordnen. Dadurch sind Dokumente einfacher und schneller wieder zu finden.

In dem Projekt wird Apache Lucene zur Indexierung genutzt.

Volltextsuche

Bei der Volltextsuche wird das Internet, der Computer oder ein Server nach einem bestimmten Wort oder einer bestimmten Zeichenfolge durchsucht. Dabei werden die Suchbegriffe mit allen Datensätzen aus dem Dokument abgeglichen.

Für das Projekt wird Apache Lucene als Modul eingebunden, um so die Volltextsuche durchzuführen. Da ein modularer Aufbau vorliegt, kann Apache Lucene in Zukunft durch eigene Implementierungen der Volltextsuche ersetzt werden.

Tokenizer

Ein Tokenizer ist ein Computerprogramm, welches die Aufgabe hat, einen Text oder ein Dokument in kleine Einheiten, z.B. Sätze, Wörter oder Zeichen, zu zerlegen.

In dem Projekt wird diese Aufgabe von Apache Lucene übernommen.

Im Folgenden sind einige Beispiele zur Implementierung von Tokenizern aufgelistet:

Standard Tokenizer

Der *Standard Tokenizer* trennt die Wörter eines Textes voneinander. Dabei werden auch Satzzeichen entfernt.

Whitespace Tokenizer

Diese Art von Tokenizer trennt die einzelnen Zeichenfolgen voneinander nachdem ein Leerzeichen erreicht wird. Die so getrennten Zeichenfolgen werden dann in einer Liste gespeichert.

Authentifizierungstoken

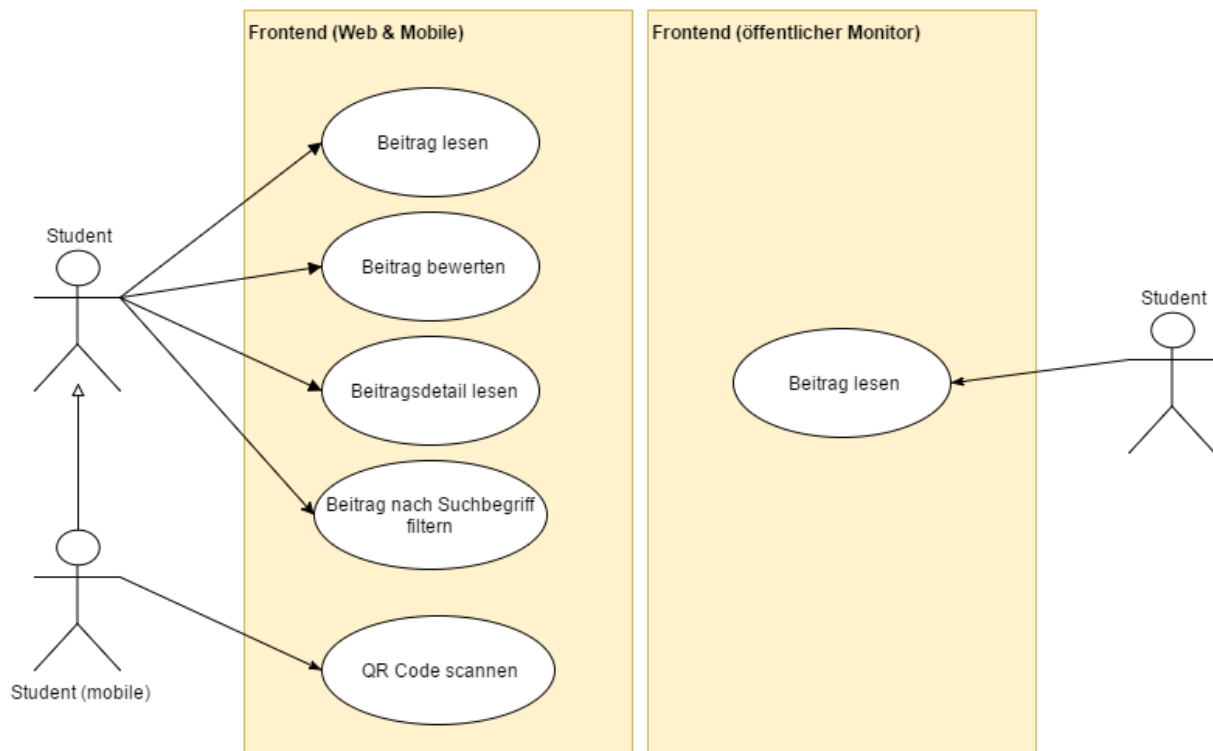
Ein Authentifizierungstoken ist ein Erkennungsmerkmal, welches dazu dient, sich dem System gegenüber zu autorisieren. So wird nichtberechtigter Zugriff auf bestimmte Daten einer Software vermieden.

In dem Projekt werden Tokens für Analyzer- und Crawlerzugänge verwendet, um nur autorisierten Analyzern und Crawlern den Zugriff auf die gespeicherten Dokumenten und das System zu erlauben.

Konzeptionelle Arbeiten

Use-Case-Diagramme

Frontend

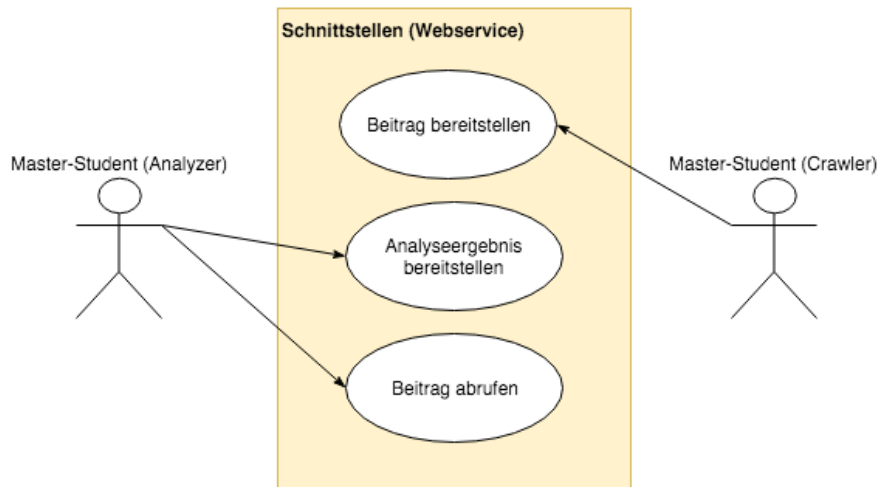


Die zu erfüllenden Use-Cases unterscheiden sich je nach Plattform: Auf dem öffentlichen Monitor ist keine Interaktion möglich, der User kann nur Beiträge lesen und das User Interface muss selbstständig zwischen verschiedenen Newsbeiträgen wechseln.

Auf der Webansicht und auch der mobilen Applikation kann dagegen eine Interaktion erfolgen: Neben dem Lesen von Newsbeiträgen muss es möglich sein, Newseinträge zu bewerten und in eine Detailansicht zu dem jeweiligen Newsbeitrag zu wechseln. In dieser Detailansicht, wird der komplette Text angezeigt, falls dieser aus Platzgründen vorher nur verkürzt angezeigt wurde, außerdem sollen die Sätze nach den Ergebnissen der Sentimentanalyse entsprechend entweder grün für einen positiven oder rot für einen negativen Satz, eingefärbt werden.

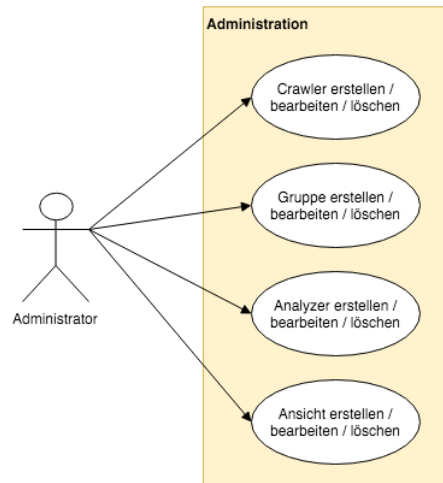
In der Web bzw. Mobilen Ansicht soll auch eine Suche ausgeführt werden können, dann werden nur Newsbeiträge angezeigt die den jeweiligen Suchbegriff, z.B. das Wort „Polizei“ enthalten.

Schnittstellen



Um eine zukünftige Erweiterbarkeit sicherzustellen, soll eine Schnittstelle existieren um später Crawler und Analyzer über die Webservices anzubinden.

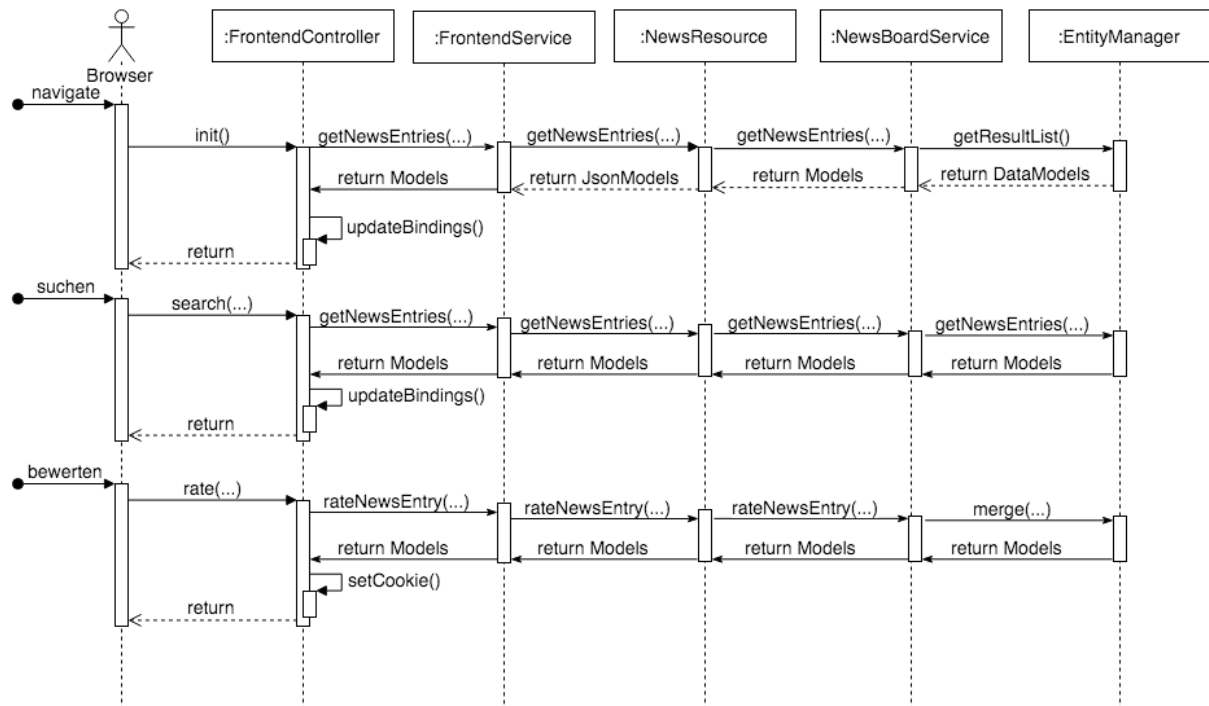
Administration



Des Weiteren soll eine Administrationsoberfläche existieren, dort soll ein befugter Administrator die Möglichkeit haben, Crawler, Analyzer, Gruppen und Ansichten (Views) zu erstellen, zu bearbeiten und zu löschen. Zur Definition dieser Begriffe siehe Kapitel „Theoretische Grundlagen“.

Sequenzdiagramme

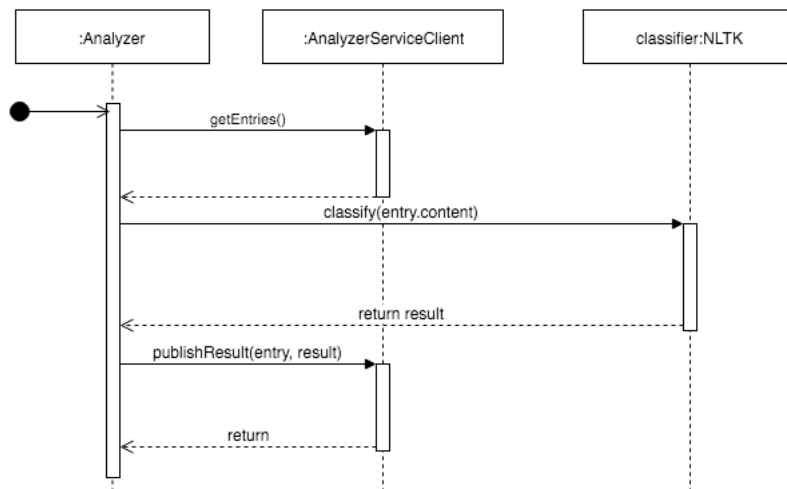
Typische Webinteraktion



Dieses Sequenzdiagramm zeigt eine typische Webinteraktion. Sobald der User die Website aufruft wird der Angular FrontendController initialisiert, dieser ruft die `getNewsEntries()` Methode des Webservice FrontendService auf. Dieser ruft die `getNewsEntries()` Methode des Klasse NewsRessource auf, wobei zwischen FrontendService und NewsRessource ein Übergang von „client tier“ zu „middle tier“ innerhalb des Schichtenmodells stattfindet. Dann wird die `getNewsRessource()` Methode des NewsBoardService aufgerufen der wiederum die `getResultList()` Methode des EntityManager aufruft welcher die DataModels direkt aus der PostgreSQL Datenbank zurückliefert. Hierbei findet ein Übergang von „middle tier“ zu „data-server tier“ statt. Diese Models werden über den NewsBoardService an die NewsRessource zurückgeliefert. Nun erfolgt die Rückgabe der Daten an den FrontendService als JSON Objekte bevor sie weiter an den FrontendController gegeben werden, der für das Databinding zuständig ist und die Darstellung im Frontend entsprechend aktualisiert.

Der Sequenzablauf für die Suche und die Bewertung von Artikeln erfolgt analog dazu.

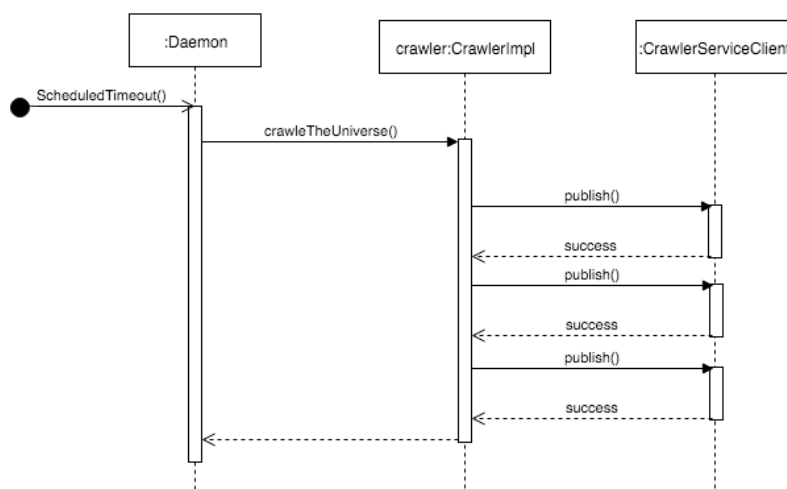
Beispielhafte Analyzer-Sequenz



Dieses beispielhafte Sequenzdiagramm zeigt eine mögliche Implementierung des NLTK (Native Language Toolkit) für die Klassifizierung von Sätzen. Von einer Implementierung unsererseits wurde vorerst abgesehen.

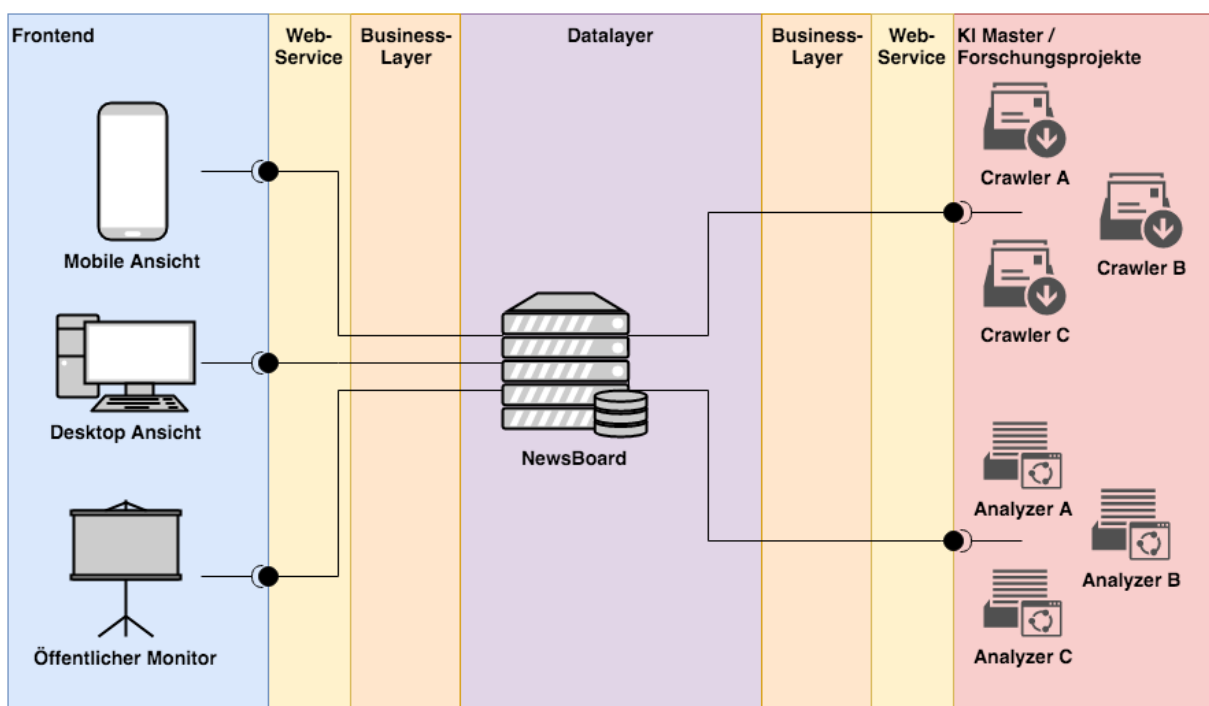
In diesem würde die Analyzer Klasse vom AnalyzerServiceClient neue unklassifizierte Newseinträge abrufen, diese Newsbeiträge werden über die classify() Methode an das NLTK übergeben. Die bewerteten Beiträge werden zurückgeliefert und wiederum über die publishResult() Methode an den AnalyzerServiceClient zurückgegeben.

Beispielhafte Crawler-Sequenz



Dieses beispielhafte Sequenzdiagramm zeigt, wie ein regelmäßiger Crawling-Prozess angestoßen werden könnte. In regelmäßigen Abständen würde ein Daemon die `crawlTheUniverse()` Methode der Crawler Implementierungsklasse „CrawlerImpl“ ausführen. Die Ergebnisse des Crawlers werden dann über die `publish()` Methode der Klasse CrawlerServiceClient veröffentlicht. Somit würde durch die Implementierung der Crawler-Sequenz sichergestellt, dass die Datenbank regelmäßig mit neuen Newseinträgen erweitert wird.

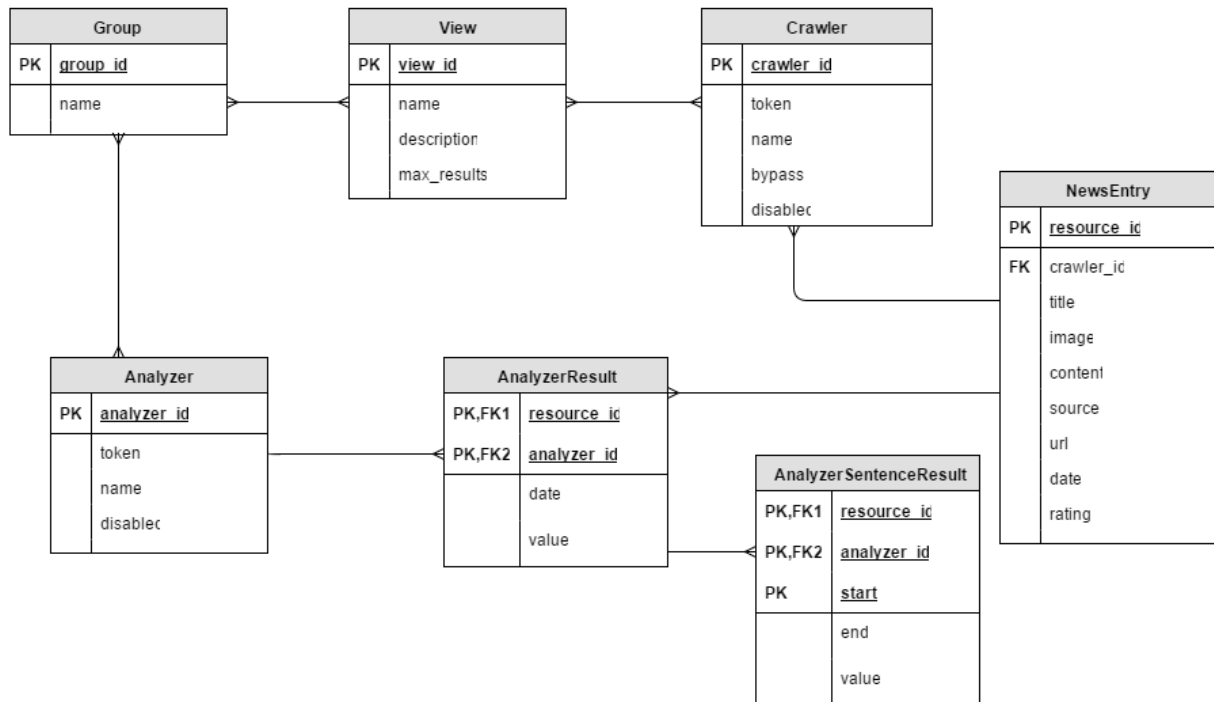
Software-Architektur



Das KI-Newsboard verteilt sich auf 3 Stufen (tier), die aus dem „data server tier“, in unserem Fall PostgreSQL, dem „middle tier“ und dem „client tier“ besteht. Das „middle tier“ wird auf die Datenzugriffs-, Geschäftslogik- und Webservice-Schicht verteilt. Das „client tier“ beinhaltet die Präsentations- und die Client-Kommunikations-Schicht.

ER-Modell

Das Design der Datenbank in der Barker-Notation ist der folgenden Abbildung zu entnehmen.



Die Entität *NewsEntry* stellt einen Nachrichtenbeitrag dar und verfügt somit über Attribute, die eine allgemeine Struktur eines Beitrages beschreiben. Dazu zählen der Titel, der Inhalt, das Bild, sofern vorhanden, die Quelle, das Datum und die URL des Beitrages. Zusätzlich wird die Bewertung des Beitrages, welche durch Benutzer der Software im Browser oder in der mobilen Applikation erfolgt, in der Datenbank gespeichert. Um zu identifizieren, welcher Crawler den Beitrag abgerufen hat, wird die Id des jeweiligen Crawlers in dieser Tabelle hinterlegt.

Ein Crawlerzugang wird durch die Entität *Crawler* beschrieben. Dieser hält neben seinem Namen noch seinen aktuellen Aktivitätsstatus und das Attribut *bypass*, welches angibt, ob die Crawlerergebnisse durch die Analyzer behandelt werden sollen. Außerdem ist ein Token für den Crawlerzugang notwendig, damit der Crawler dem System gegenüber authentifiziert ist und somit kein unbefugter Datenzugriff stattfindet.

Auf der anderen Seite ist die Entität *AnalyzerResult* über einen Fremdschlüssel auf den Primärschlüssel *resource_id* und einen Fremdschlüssel auf den Primärschlüssel *analyzer_id* mit den beiden Entitäten *NewsEntry* und *Analyzer* verknüpft. Die Entität *AnalyzerResult* repräsentiert das Ergebnis der Sentiment-Analyse eines Beitrages. Um die Eindeutigkeit eines Ergebnisses zu gewährleisten, wird ein zusammengesetzter

Primärschlüssel aus den beiden Fremdschlüsseln erstellt. Zudem verfügt die Tabelle über das Datum und das Ergebnis der Analyse.

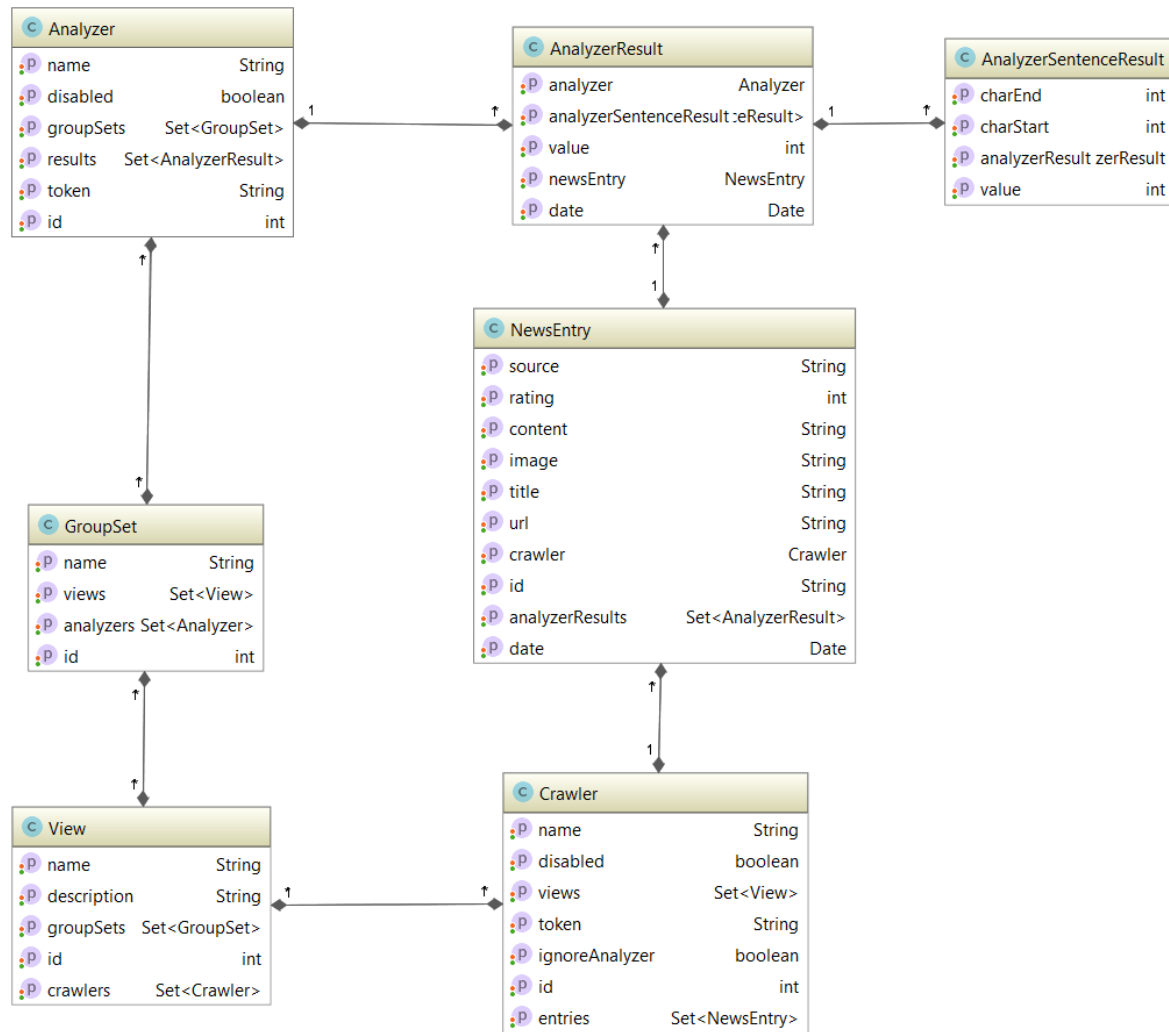
Die Entität *AnalyzerSentenceResult* steht für das Ergebnis der Sentiment-Analyse auf Satzebene. Zusätzlich zu den Fremdschlüsseln auf Entitäten *NewsEntry* und *Analyzer* wird der Startindex des analysierten Satzes als zusammengesetzter Primärschlüssel verwendet. So kann verhindert werden, dass ein Analyseergebnis nicht eindeutig einem Satz zugeordnet wird. Auch werden der letzte Index des analysierten Satzes und das Ergebnis der Sentiment-Analyse zu dem analysierten Satz in dieser Entität gespeichert.

Ein Analyserzugang wird durch die Entität *Analyzer* dargestellt. Diese speichert neben ihrem Namen und ihrem Aktivitätsstatus einen Token, welcher zum Authentifizieren dem System gegenüber dient.

In Zukunft soll jedem KI-Masterstudenten ein Analyzer zugeordnet werden. Damit nicht jeder einzelne Analyzer einer Ansicht, hier durch die Entität *View* repräsentiert, zugeordnet werden muss, werden die Analyzer gruppiert. Die Entität *Group* stellt die Gruppierung der Analyzer dar. Eine Gruppe verfügt lediglich über einen Namen und einer Id und ist mit den Entitäten *Analyzer* und *View* verknüpft.

Da im Produktiveinsatz mehrere Crawler eingesetzt werden können und nicht immer die Ergebnisse aller Crawler in einer Ansicht angezeigt werden sollen, kann in einer Ansicht dies definiert werden. Außerdem kann die Menge der angezeigten Beiträge ebenfalls eingestellt werden. Eine Ansicht besitzt auch einen Namen und eine Beschreibung.

Das Klassendiagramm der Datenzugriffsschicht ist in der folgenden Abbildung ersichtlich. Auffällig dabei ist, dass die Verknüpfungen und Attribute der einzelnen Klassen analog zum ER-Diagramm modelliert werden.



Powered by yFiles

Projektverlauf

Treffen 1, 22.10.2016

Thema der Besprechung/Anlass:	Projektplanung, Lastenheft
Datum:	22.10.2016
Anwesende:	Cem, Kevin, Azad, Veronika
Leitung:	-
Protokollführer:	Veronika

TOPIC	Diskussion	Ergebnis	Zuständigkeit/Aufgaben
1	Lastenheft	Projekt, Ziele, Produkteinsatz, funktionale und nichtfunktionale Anforderungen beschreiben. Offene Punkte finden und Vorgehensmodell für Entwicklung des Projektes wählen.	-
2	Architekturübersicht		-
3	Grober Zeitplan	GANTT Diagramm erstellt	-

Treffen 2, 29.10.2016

Thema der Besprechung/Anlass:	Frontend Prototypes, Konzeptionierung
Datum:	29.10.2016
Anwesende:	Cem, Kevin, Azad, Veronika
Leitung:	-
Protokollführer:	Veronika

TOPIC	Diskussion	Ergebnis	Zuständigkeit/Aufgaben
1	Frontend Prototype	Drei Prototypen wurden erstellt: Mobile App, Web und große Bildschirme.	Wahl eines Prototypen für große Bildschirme
2	ER-Diagramm		-

Treffen 3, 03.11.2016

Thema der Besprechung/Anlass:	Grundlegende Codestruktur
Datum:	03.11.2016
Anwesende:	Cem, Kevin, Azad, Veronika
Leitung:	-
Protokollführer:	Veronika

TOPIC	Diskussion	Ergebnis	Zuständigkeit/Aufgaben
1	Grundlegende Projektstruktur, Auswahl der Frameworks und Komponenten (Frontend)	Endgültige Wahl auf AngularJS, sowie Bootstrap für Frontend	-

Treffen 4, 10.11.2016

Thema der Besprechung/Anlass:	WebService, Abhängigkeitsverwaltung
Datum:	10.11.2016
Anwesende:	Cem, Kevin, Azad, Veronika
Leitung:	-
Protokollführer:	Veronika

TOPIC	Diskussion	Ergebnis	Zuständigkeit/Aufgaben
1	WebService Signaturen	Wurden erstellt	Weitere Implementierung: Cem + Kevin
2	Abhängigkeitsverwaltung	Wahl fällt auf bower statt npm wegen statischem Deployment auf glassfish	-

Treffen 5, 17.11.2016

Thema der Besprechung/Anlass:	Wöchentliches Treffen
Datum:	17.11.2016
Anwesende:	Cem, Kevin, Azad, Veronika
Leitung:	-
Protokollführer:	Veronika

TOPIC	Diskussion	Ergebnis	Zuständigkeit/Aufgaben
1	Wo Konfigurationen speichern?	persistence.xml und payara-server sollten für unsere Anforderungen genügen	Ressourcen im Payara Server definieren
2	Wie Webservice Dokumentieren?	Entscheidung: swagger.io kann auch Client-Code generieren	Cem erstellt Schnittstellenbeschreibung für folgende Implementierungen
3	Wie soll Authentifizierung implementiert werden?	Als Webservice Filter	Azad recherchiert wie Filter implementiert werden können

Treffen 6, 24.11.2016

Thema der Besprechung/Anlass:	Wöchentliches Treffen
Datum:	24.11.2016
Anwesende:	Cem, Kevin, Azad, Veronika
Leitung:	-
Protokollführer:	Veronika

TOPIC	Diskussion	Ergebnis	Zuständigkeit/Aufgaben
1	QR-Code Client oder Serverseitig?	Mit dem entsprechenden Modul Clientseitig generieren.	Kevin sucht Modul für QR Code Generierung
2	Masken im Adminbereich aufteilen als Master-/ Detail- Page?	Einfache Masken nebeneinander, sonst extra Seite für Details.	-

Treffen 7, 01.12.2016

Thema der Besprechung/Anlass:	Wöchentliches Treffen
Datum:	01.12.2016
Anwesende:	Cem, Kevin, Azad, Veronika
Leitung:	-
Protokollführer:	Veronika

TOPIC	Diskussion	Ergebnis	Zuständigkeit/Aufgaben
1	Exception Handling im Webservice	Einsatz von Exception Mapper	Cem
2	Schönere Ansicht im öffentlichen Monitor	CSS erweitern/verbessern	Kevin

TOPIC	Diskussion	Ergebnis	Zuständigkeit/Aufgaben
3	Datenmodelle kapseln oder durchschleifen	Datenmodelle durchschleifen	Azad
4	Aussehen der Crawler-Administrationsoberfläche	Master-/Slave-Ansicht auf einer Seite	Veronika

Treffen 8, 08.12.2016

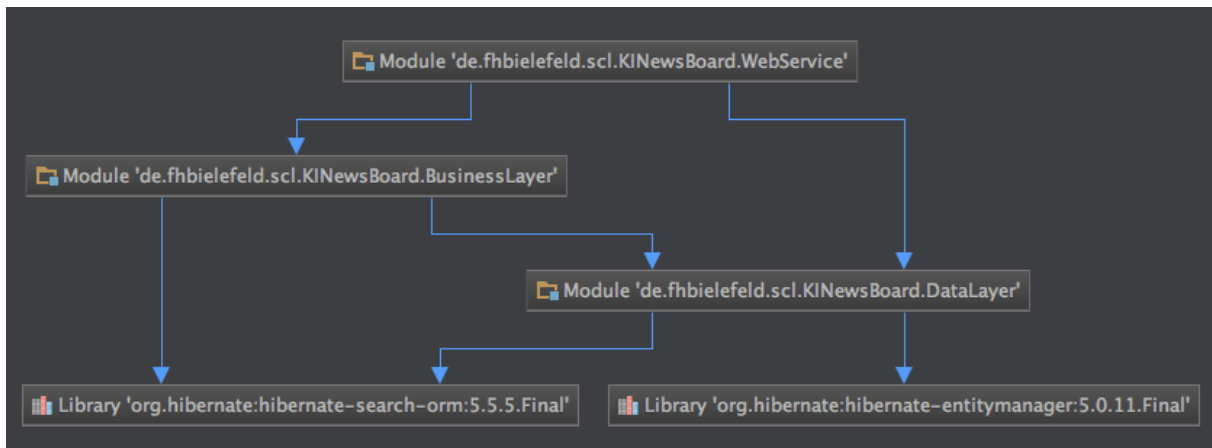
Thema der Besprechung/Anlass:	Wöchentliches Treffen
Datum:	08.12.2016
Anwesende:	Cem, Kevin, Azad, Veronika
Leitung:	-
Protokollführer:	Veronika

TOPIC	Diskussion	Ergebnis	Zuständigkeit/Aufgaben
1	Bibliothek zur Validierung von Eingaben	Bean Validation API	Cem
2	Aussehen der Detailansicht	Beim Klicken auf einen Beitrag wird dieser größer und steht im Vordergrund	Kevin
3	Wie soll JavaScript dokumentiert werden?	JSDoc	Kevin, Veronika

Implementierung

Middle-tier

Um eine hohe Kohäsion zu erreichen, werden die einzelnen Schichten in dem „middle tier“ auf die Package- und Module- Organisation wiedergespiegelt.

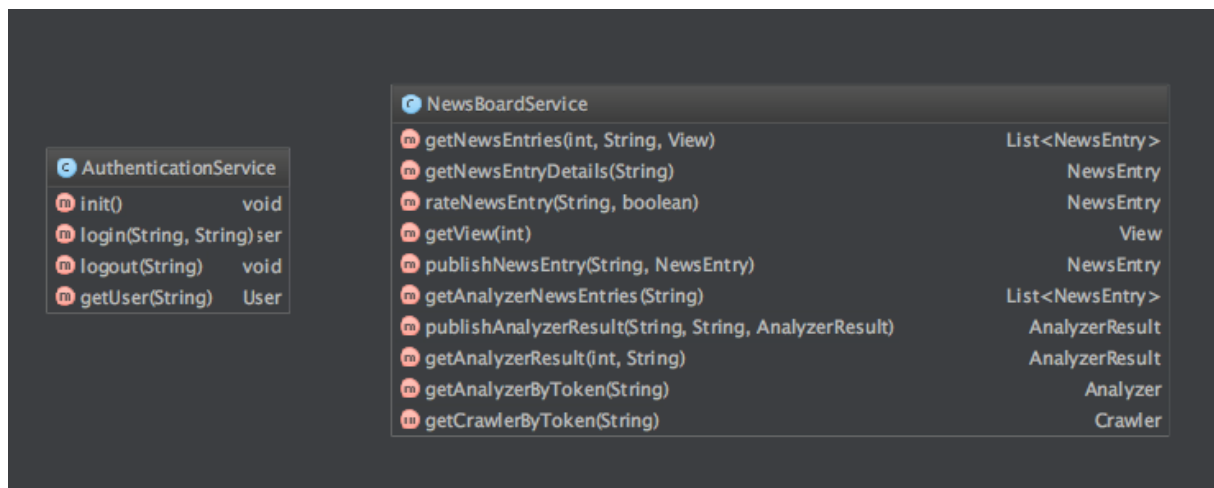


Die Abhängigkeit zwischen der Webservice-Schicht und der Datenzugriffs-Schicht beschränkt sich auf die Verwendung der in der letzteren Schicht implementierten Datenmodelle.

Um die Kohäsion weiter zu steigern, kann diese Abhängigkeit durch die Implementierung von Datentransfer-Objekten in der Geschäftslogik-Schicht aufgelöst werden. Aufgrund des geringen Mehrwerts gegenüber dem Mehraufwand, vor allem bei der Weiterentwicklung, wurde auf die weitere Separation verzichtet.

Datenzugriffs- und Geschäftslogikschicht

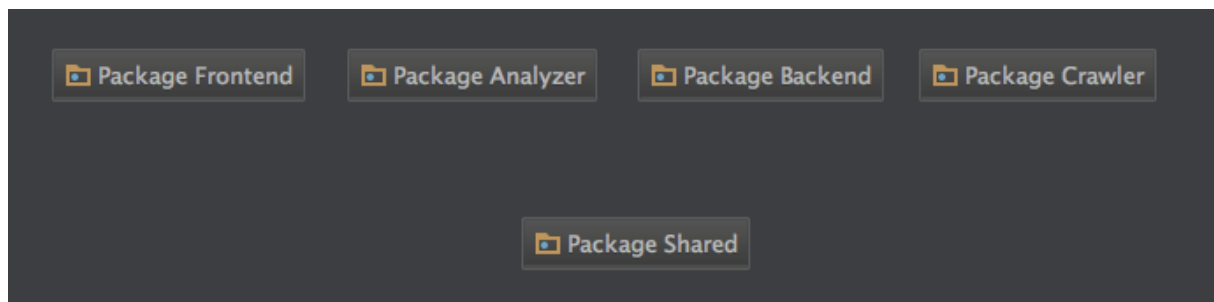
Durch die Verwendung der Java Persistence API, minimiert sich die Datenzugriffsschicht auf die Implementierung der im ER-Diagramm modellierten Entitäten. Diese werden dann in der Geschäftslogikschicht verwendet, um die Anforderungen aus den Use-Cases zu implementieren.



Darüber hinaus übernimmt die Geschäftslogikschicht die Authentifizierung der Administratoren, gegenüber dem SCL-Authentication-Service und sichert den Zugriff auf Daten, vor unautorisierten Zugriff.

Webservice-Schicht

Die Webservice-Schicht ist zur besseren Wartbarkeit in 5 weitere Packages unterteilt.



Jedes Package entspricht der Schnittstelle des Actors aus den Use-Cases und implementiert einen eigenen Webservice. Gemeinsame Funktionalitäten sind in dem Package „Shared“ implementiert.

Die Webservice-Schicht bildet die Schnittstelle zu der Geschäftslogikschicht und implementiert lediglich, die für diese Schicht notwendigen Anforderungen. Die von der Geschäftslogikschicht bereitgestellten Dienste werden, wie folgender Codeausschnitt zeigt, mithilfe der EJB-Annotation injiziert.

```
@EJB
private NewsBoardService newsBoardService;
```

Der Webservice wird mittels JAX-RS-API realisiert und nutzt zur Übertragung die Javascript Object Notation.

```
@Provider
public class AuthenticationFilter implements ContainerRequestFilter {

    @EJB
    AuthenticationService service;

    @Override
    public void filter(ContainerRequestContext requestContext) {

        final MultivaluedMap<String, String> headers = requestContext.getHeaders();
        final List<String> authorization = headers.get(HttpHeaders.AUTHORIZATION);

        if (authorization == null || authorization.isEmpty()) {
            requestContext.abortWith(Response.status(Response.Status.UNAUTHORIZED)
                .entity(new ErrorModel("Bitte geben Sie einen Authentication Token an.")).build());
            return;
        }

        final String token = authorization.get(0).replaceFirst("BASIC ", "");

        User user = service.getUser(token);
        if (user == null) {
            requestContext.abortWith(Response.status(Response.Status.UNAUTHORIZED)
                .entity(new ErrorModel("Der Zugriff ist nicht erlaubt.")).build());
            return;
        }

        requestContext.setSecurityContext(...);
    }
}
```

Ebenfalls beinhaltet es Funktionalitäten, um sich mittels dem Hypertext Transfer Protocol (HTTP) zu authentifizieren. Dazu wird die „ContainerRequestFilter“-Schnittstelle aus dem JAX-RS Framework wie folgt implementiert.

Somit wird bei jedem Webservice-Aufruf zunächst zentral die Authentifizierung geprüft, bevor die Anfrage an die verantwortliche Routine weitergereicht wird.

Ein weitere Aufgabe Webserivce-Schicht ist das Umformen der auftretenden Exceptions in der Datenzugriffs- und Geschäftslogik-Schicht, um diese mittels HTTP-Statuscode zu reflektieren.

```
@Provider
public class ValidationExceptionMapper implements ExceptionMapper<ConstraintViolationException> {

    @Override
    public Response toResponse(ConstraintViolationException e) {
        String msg = "";
        ...
        return Response.status(Response.Status.BAD_REQUEST)
            .entity(new ErrorModel(msg))
            .build();
    }
}
```

Dazu wird für jede mögliche Exception, das generische Interface „ExceptionMapper“ implementiert.

Der Vorteil in der Verwendung des ExceptionsMappers ist die zentrale Behandlung von Exceptions. Damit wird sichergestellt dass die Rückmeldungen einheitlich sind.

Client-tier

Präsentations-Schicht

Die Präsentations-Schicht ist mittels dem Model View Viewmodel Pattern, unter Zuhilfenahme des AngularJS-Frameworks implementiert. Dazu besitzt jede View ein Controller der wiederum über die Variable „\$scope“ Inhalte für die Datenbindung bereitstellt.

```
angular.module('nwb.frontend', ['ui.router'])
  .controller('FrontendViewController', ['$scope', 'FrontendService',
    function ($scope, FrontendService) {

      $scope.entries = [];

      $scope.updateView = function () {
        $scope.isLoading = true;

        FrontendService.getNewsEntries(...).then(function (entries) {
          $scope.entries = entries;
        });
      };
    }
  ]);
```

Alle in der View angezeigten Informationen werden sowohl durch Einweg- als Zweiweg-Datenbindungen realisiert.

Client-Kommunikations-Schicht

Für den Datenaustausch zwischen der Präsentations-Schicht und der Webservice-Schicht in der „middle tier“, ist die als Service Pattern implementierte Client-Kommunikations-Schicht verantwortlich. Diese abstrahiert die Logik für die Übermittlung der Daten an und vom REST-Webservice.

```
angular.module('nwb')
  .factory('FrontendService', ['$http', '$q', function ($http, $q) {
    var api = "WebService/frontend/";
    var service = {};

    service.getNewsEntries = function (page, keyword, view) {

      var params = {};
      ...
      return $http.get(api + "news", {params: params});
    };
  }]);
```

Der Service wiederum wird mittels Injektion in die entsprechenden Controller injiziert.

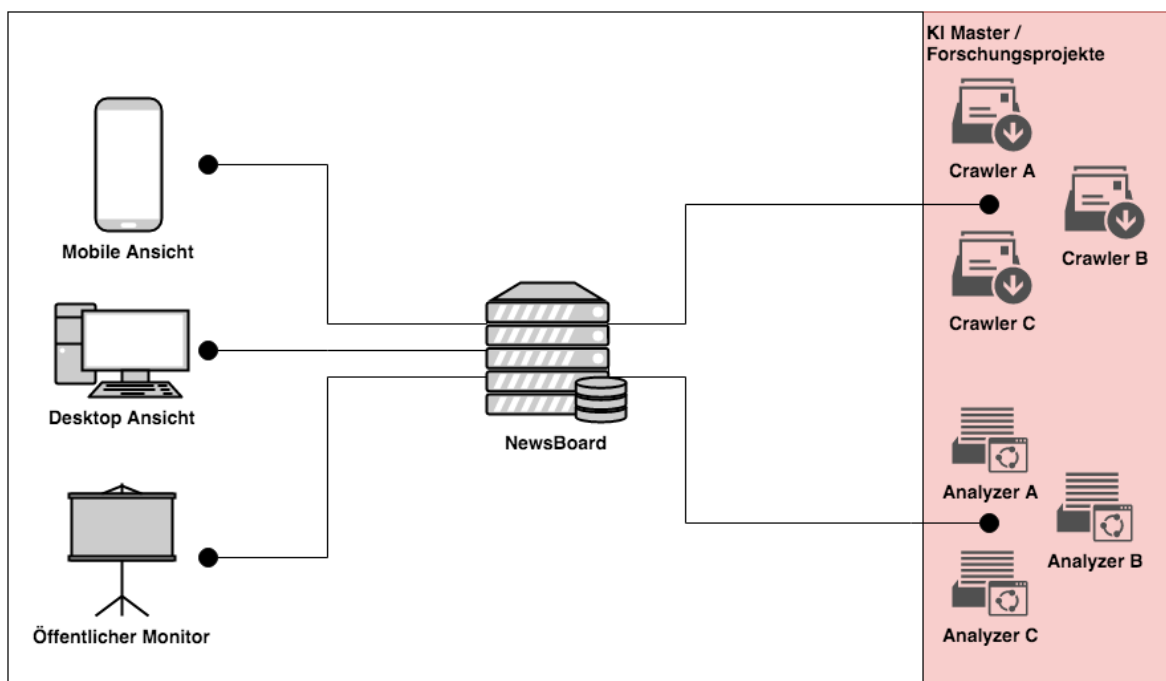
Ausblick

Eines der Anforderungen an das Projekt war es, dass ein modularer Aufbau der Software vorhanden ist. So soll ermöglicht werden, dass einzelne Module ausgetauscht werden können. Ein Modul, welches dafür in Frage kommt, ist z.B. Apache Lucene.

Auch soll das Ranking der Suchergebnisse, die die Suchleiste liefert, ersetzbar sein. Hierdurch soll die Möglichkeit geschaffen werden, die Suchergebnisse nach verschiedenen Merkmalen sortieren zu können.

Während des Projekts wurden lediglich Testcrawler und Testanalyzer implementiert, für den Produktiveinsatz sollen mindestens zwei echte Crawler entwickelt und angebunden werden (siehe untenstehende Grafik).

Das Builden soll von der IDE losgelöst über ein Buildscript erfolgen und abhängig vom Continuous Deployment gewählt werden. Um die korrekte Funktionalität der Software zu gewährleisten, ist die Integration von Unit-Tests erforderlich.



Fazit

Das Projekt hat Mitte Oktober 2016 gestartet und endete nach ca. drei Monaten im Januar. Nach einer umfangreichen Anforderungsanalyse deren Ergebnisse im Pflichtenheft festgehalten wurden und durch die definierten Meilensteine einen genauen Zeitrahmen festgelegt haben, konnte die Implementierung der verschiedenen Komponenten (Frontend, Webservices, Backend) unabhängig voneinander vonstattengehen.

In dem Projekt sind alle Webengineering-Aspekte behandelt worden, das heißt, die Implementation von der Datenbank, der Logik der Anwendung, das Frontend der Anwendung und dessen Schnittstellen waren vollständig dem Team überlassen worden. Dadurch wurde das Interesse an dem Projekt viel größer, jedoch gestaltete sich auch das Projekt um einiges umfangreicher.

Das Ziel der Projektarbeit war es, ein modulares webbasiertes System zur Anzeige von FH-bezogenen Nachrichten zu implementieren. Dabei ging es weniger darum, eigene Analyzer und Crawler zu erstellen, sondern vielmehr um die Implementierung eines Systems, welches dazu dient, Beiträge anzuzeigen, die durch Crawler aus dem Web aggregiert, daraufhin von Analyzern einer semantischen Untersuchung unterzogen und im Anschluss in die Datenbank geschrieben wurden. Dabei lag ein Hauptaugenmerk darauf, dass für Crawler und Analyzer eine Schnittstelle zur Verfügung stehen musste, die eine zukünftige Erweiterung erlaubt.

Natürlich läuft kein Projekt von Anfang bis Ende ohne Schwierigkeiten, so auch dieses. Um die Funktionalität der Software zu testen, war ein umfangreicher Testdatensatz erforderlich. Die Beispieldaten wurden durch Testcrawler aggregiert, wobei auffiel, dass die Entwicklung von generischen Crawlern sehr komplex ist. Ein Crawler muss auf die jeweilige Seite stark angepasst sein, damit sichergestellt ist, dass keine anderen Inhalte der Quellseite neben den eigentlichen Artikeldetails erfasst werden. Für unsere Beispieldatensätze war somit ein Crawler, der RSS-Feeds auslesen kann, relativ schnell und fehlerfrei umzusetzen.

Insgesamt lässt sich sagen, dass das Projekt die gewünschten Anforderungen realisiert hat, wodurch es schon praktisch einsatzfähig ist.

Installationshinweise und Benutzerhandbuch

Für die Installation des KI-Newsboards werden die im folgenden genannten Anwendungen auf dem Server und der Entwicklungsmaschine vorausgesetzt.

Voraussetzungen Server

- Payara Server Version 4.1
- PostgreSQL Server Version 9.X
- PostgreSQL Treiber Version (postgresql-9.X.jar) in das Payara Verzeichnis 'glassfish/domains/XXX/lib/' kopieren
- PostgreSQL Datasource in Payara mit dem Namen 'jdbc/newsboard_readwrite' erstellen
- Das Verzeichnis, das in der persistence.xml unter 'hibernate.search.default.indexBase' definiert ist muss existieren. Alternative kann auch die Pfadangabe derselbigen Einstellung angepasst werden. Wichtig: Der Benutzer unter dem der Payara Server ausgeführt wird, benötigt Schreibrechte auf das Verzeichnis.

Voraussetzungen Entwicklungsmaschine

Falls Änderungen an der Anwendungen vorgenommen und diese erneut kompiliert werden muss, müssen folgende Voraussetzungen erfüllt werden.

- IntelliJ IDEA ab Version 2016.2.5
- Node.JS ab Version 4.4.3
- Bower Global installieren: npm -g install bower

Kompilieren der Anwendung

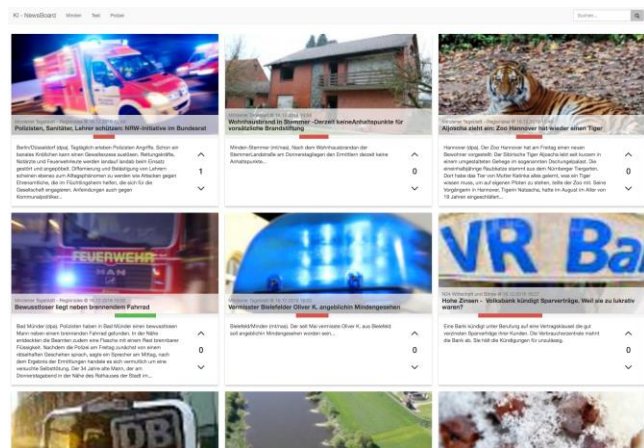
- In das Verzeichnis de.fhbielefeld.scl.KINewsBoard.Web wechseln
- In Kommandozeile folgenden Befehl ausführen: bower install
- Projekt in IntelliJ öffnen und mit Build Artifacts... das Artifact ‚NewsBoard‘ erstellen

Installation

Das Applikationsarchiv ‚NewsBoard.ear‘ über die Payara-Administration auf den Server deployen.

Handbuch

Web-/Mobile- Frontend



Die Web-/Mobile- Ansicht zeigt alle gecrawlten und analysierten Elemente der als Standard definierten View an. Ist keine Standard View definiert, werden alle Elemente angezeigt. Die Standard View kann im Administrationsbereich unter Views definiert werden.

Suche



Über die Textbox in der Navigationsbar können die angezeigt Elemente durchsucht werden. Zusätzlich gibts es vordefinierte Suchbegriffe, die in der Navigationsbar als Menüeinträge angezeigt werden.

Bewertung

Über die Auf- und Ab-Symbole, kann ein Element bewertet werden. Dies kann je nach Implementierung die



Sortierung der Elemente beeinflussen. Zusätzlich kann diese Bewertung als Feedback für die Analyzer herangezogen werden, um die analysen weiter zu optimieren.


Analyseergebnis



Jedes Element zeigt das kummulierte Analyseergebnis in Form eines positiven/negativen Fortschrittsbalken an. Dabei wird ist der negative Fortschrittsbalken rot und verläuft von der mitte des Elements nach links. Ist das Element positiv analysiert worden so ist der Fortschrittsbalken grün und verläuft von der mitte nach rechts.

Detailansicht

Durch klicken der Überschrift oder des Vorschaubilds, öffnet sich die Detailansicht des Elements. Hier wird neben den Informationen aus der Übersicht und dem kompletten Inhalt, ebenfalls die satzweise Analyse des Elements je nach Analyzer angezeigt.




Polizisten, Sanitäter, Lehrer schützen: NRW-Initiative im Bundesrat

Erstellt: 16.12.2016 12:48
 Externer Link: [Mindener Tageblatt - Regionales](#)


Berlin/Düsseldorf (dpa). Tagtäglich erleben Polizisten Angriffe. Schon ein banales Knöllchen kann einen Gewaltexzess auslösen. Rettungskräfte, Notärzte und Feuerwehrleute werden landauf landab beim Einsatz gestört und angepöbelt. Diffamierung und Belästigung von Lehrern scheinen ebenso zum Alltagsphänomen zu werden wie Attacken gegen Ehrenamtliche, die im Flüchtlingsheim helfen, die sich für die Gesellschaft engagieren. Anfeindungen auch gegen Kommunalpolitiker...

Test Analyzer 1	4
Test Analyzer 3	78
Test Analyzer 2	12


Öffentlicher Monitor




Autobahnen in Bayern - Hätten Sie gewusst, wozu diese Schilder gut sind?
 Sie sind rund, schwarz-weiß und 70 Zentimeter breit: Auf den Autobahnen 9 und 93 wurden 13 neue Verkehrsschilder aufgestellt. Dabei handelt es sich um keine gewöhnlichen Hinweise für Autofahrer.




#insidefacebook - Facebook-Mitarbeiter erzählen von ihren Arbeitsbedingungen
 "SZ"-Reporter sprachen trotz der Schweigepflicht mit Facebook-Mitarbeitern und berichten über deren psychische Belastungen. "Ich habe Sachen gesehen, die mich am Guten zweifeln lassen", sagt einer.




Zuwachs von 1,8 Prozent - Wirtschaftsforscher vom RWI senken Wachstums-Prognose
 Essen (dpa)- Das Wirtschaftsforschungsinstitut RWI hat seine Prognose für das Wirtschaftswachstum in Deutschland gesenkt. Für das zu Ende gehende Jahr werde nur noch mit einem Zuwachs von 1,8...



Geflügelpest rückt näher
 Sternwede/Minden (mt/GB). Der Kreis Minden-Lübbecke weitet seine Maßnahmen zur Verhinderung der Ausbreitung der Geflügelpest aus. Im benachbarten Landkreis Vechta ist vorgestern ein Fall von Vogelgrippe in Damme/Rüschendorf amtlich festgestellt worden. In einem Umkreis von zehn




Hier blitzt die Polizei heute im Kreis Minden-Lübbecke
 Minden (mt). Die Polizei NRW will für mehr Sicherheit auf den Straßen sorgen, deswegen werden vorgesehene Geschwindigkeitsmessungen vorab...




Neue Elektronik-Firma Ceconomy - Metro kommt bei
 Aufnahmen von...

Campus Minden Newsboard

Aktuelle Meldungen zum Campus Minden

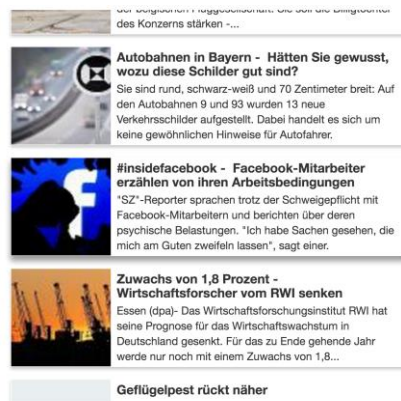




N24 Wirtschaft und Börse @ 16.12.2016 06:41
#insidefacebook - Facebook-Mitarbeiter erzählen von ihren Arbeitsbedingungen
 "SZ"-Reporter sprachen trotz der Schweigepflicht mit Facebook-Mitarbeitern und berichten über deren psychische Belastungen. "Ich habe Sachen gesehen, die mich am Guten zweifeln lassen", sagt einer.

Die öffentliche Ansicht zeigt alle gecrawlten und analysierten Elemente der einer View. Die View kann im Administrationsbereich unter Views definiert werden. Die öffentlich Ansicht ist auf den Präsentationsmodus d.h. ohne Benutzerinteraktion ausgelegt.

Nachrichtenliste



Die linke Liste zeigt eine Vorschau der vorangegangenen und folgenden Nachrichteneinträge an. Der aktuelle im Detail sichtbare Nachrichteneintrag wird dabei etwas hervorgehoben. Neben dem Vorschaubild wird die Überschrift und ein Ausschnitt des Inhalts angezeigt.

Detailansicht



Wird ein Nachrichteneintrag in der Liste gewählt, so wird diese im Detail auf der rechten Seite angezeigt. Hier wird nun zusätzlich die Quelle des Nachrichteneintrags, das Veröffentlichungsdatum und das Analyseergebnis angezeigt.

Da der Text bei der öffentlichen Ansicht auch aus größerer Entfernung gelesen werden muss, wurde die Schriftgröße entsprechend gewählt. Da es zudem nicht möglich ist Inhalte zu scrollen, wird der Text bei Überschreiten des verfügbaren Anzeigebereichs abgeschnitten.

Analyseergebnis



Jedes Element zeigt das kummulierte Analyseergebnis in Form eines positiven/negativen Fortschrittsbalken an. Dabei wird ist der negative Fortschrittsbalken rot und verläuft von der Mitte des Elements nach links. Ist das Element positiv analysiert worden so ist der Fortschrittsbalken grün und verläuft von der Mitte nach rechts.

Automatischer Wechsel

Die Nachrichteneinträge werden automatisch nach einer vorgegeben Zeit gewechselt. Somit ist es möglich ohne Benutzerinteraktion die Nachrichten im Detail zu lesen. Der Fortschrittsbalken in der rechten oberen Ecke Zeit dabei die verbleibende Zeit des aktuellen Nachrichteneintrags an.



Administration

Die Administration ermöglicht den Administratoren Crawler und Analyzer Zugänge zu verwalten, sowie das Erstellen von Views für unterschiedliche öffentliche Monitore.

KI-NewsBoard

Cem, Basoglu

Dashboard

Crawler

Analyzer

Gruppen

Views

Dashboard

Statistik Übersicht

130

News Beiträge

387

Analyze Ergebnisse

3

Aktive Analyzer

2

Aktive Crawler

Letzten News Einträge

Datum	Inhalt	Quelle
09.01.2017 13:02	Darum müssen Sie bei Glätteis ab 7 Uhr streuen Schnee schippen, Wasser abdrehen, Dachlawinen, Sturmschäden: Wird's kalt, müssen sich Millionen Bürger kümmern. Tun sie das nicht, springt die Versicherung im Ernstfall nicht ein. Eine Expertin gibt acht Tipps.	N24 Wirtschaft und Börse
09.01.2017 12:55	So lebt ihr Smartphone bei Kälte länger Smartphone-Akkus sind meistens genau dann leer, wenn man sie am dringendsten braucht. Besonders empfindlich reagieren sie auf Kälte. Der Akku kann aber länger durchhalten, wenn man ein paar Dinge beachtet.	N24 Wirtschaft und Börse

Login

Benutzername

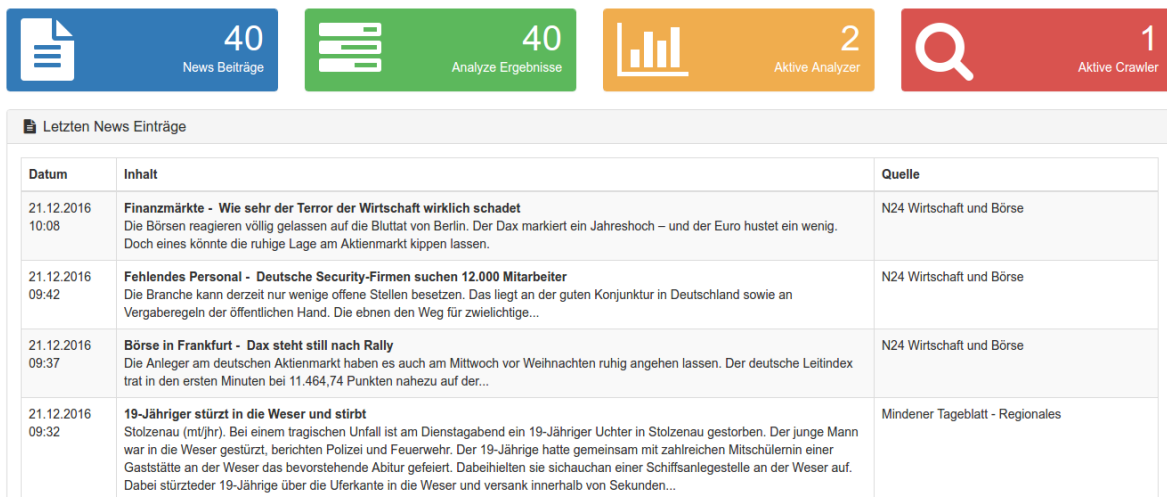
Password

Anmelden

Der Login in den Adminbereich erfolgt über das SolarComputingLab Authentifizierungssystem der FH Bielefeld mit den normalen FH Anmeldedaten.

Dashboard

Dashboard Statistik Übersicht



Das Dashboard bietet eine schnelle Übersicht über die Anzahl der momentan aktiven Newsartikel, Analyzer Ergebnissen, aktive Analyzer und aktive Crawler.

Zusätzlich werden die zuletzt hinzugefügten Newsartikel mit Zeit und Quelle angezeigt.

Crawler

Crawler

[Neuer Crawler](#)

#	Bezeichnung	Status	Analyzer
3	eins	Aktiv	
4	zwei	Aktiv	ByPass
6	drei	Inaktiv	

Die Crawler Unterseite beinhaltet eine Liste mit angelegten Crawlern, deren ID, Bezeichnung, Status (aktiv oder inaktiv) sowie ein Flag (Analyzer) ob die Ergebnisse des Crawlerns an die Analyzer weitergegeben werden sollen.

Neue Crawler anlegen

Bei dem Anlegen neuer Crawler können einige Einstellungen gesetzt werden:

- **Bezeichnung:** Name des Crawler
- **Token:** wird automatisch generiert
- **Status:** Crawler kann aktiv oder inaktiv gesetzt werden
- **Analyzer:** Das Setzen von ByPass weist an, dass Ergebnisse des Crawler nicht an Analyzer weitergereicht werden sollen

Bezeichnung

vier

Token

tln94eo9jsmcx41zrehtzkt9u8xgormlkbys4m93il51att9

Status ☐ InaktivAnalyzer ☐ ByPass

Löschen

Abbrechen

Speichern

Analyzer

Analyzer

[Neuer Analyzer](#)

#	Bezeichnung	Status
1	eins	Aktiv
2	zwei	Inaktiv

Die Analyzer Unterseite beinhaltet eine Liste mit angelegten Analyzern, deren ID, Bezeichnung und Status (aktiv oder inaktiv).

Neue Analyzer anlegen

Bei dem Anlegen neuer Analyzer können einige Einstellungen gesetzt werden.

- **Bezeichnung:** Name des Analyzer
- **Token:** wird automatisch generiert
- **Status:** ein Analyzer kann aktiv oder inaktiv gesetzt werden
- **Gruppen:** ein Analyzer kann einer oder mehreren Gruppen zugeordnet werden, die Gruppen werden dann wiederum den Views zugeordnet

Bezeichnung

eins

Token

56589lt5app3bmvggxt46ajor6z5vfgpnmpk50

Status ☐ Inaktiv

Groups

#	Bezeichnung
7	Stable Analyzer
8	Beta Analyzer
9	Master Informatik

Löschen

Abbrechen

Speichern

Gruppen

Gruppen

[Neue Gruppe](#)

#	Bezeichnung
7	Stable Analyzer
8	Beta Analyzer
9	Master Informatik

Die Gruppen Unterseite beinhaltet eine Liste mit angelegten Gruppen.

Neue Gruppen anlegen

Außerdem können neue Gruppen angelegt werden.

Bezeichnung[Löschen](#)[Abbrechen](#)[Speichern](#)

Views

Views

[Neue View](#)

#	Bezeichnung	Altionen
10	testview erste Testview	↗ 📱 ✎ 🗑
11	testview2 zweite testview	↗ 📱 ✎ 🗑

Die View Unterseite beinhaltet eine Liste mit angelegten Views. Auf der rechten Seite sind Buttons für das Anzeigen in Web/Mobile und in der öffentlichen Ansicht, sowie bearbeiten und löschen der ausgewählten View vorhanden.

Neue View anlegen

Bei dem Anlegen einer neuen View können einige Einstellungen gesetzt werden:

- **Bezeichnung:** Name des View
- **Beschreibung:** Beschreibung der View
- **Gruppe:** Gruppen denen die View angehört
- **Crawler:** Crawler aus denen die View Nachrichteneinträge bezieht

Views

Bezeichnung

testview

Beschreibung

erste Testview

Zuordnungen

Gruppen

#	Bezeichnung
7	Stable Analyzer
8	Beta Analyzer
9	Master Informatik

Crawler

#	Bezeichnung
3	eins
4	zwei
6	drei

[Abbrechen](#)
[Speichern](#)

Anlagen

Teamarbeitsplan

Milestone 1 (21.11.2016)

Utilities #1	Kevin Schima
<ul style="list-style-type: none"> • Konfigurations-Management 	
Data-Layer	Azad Sino
<ul style="list-style-type: none"> • SQL Schema • Datamodels (JPA) • DAO 	
Business-Layer #1	Azad Sino
<ul style="list-style-type: none"> • Business Logic Part <ul style="list-style-type: none"> ◦ Views • SCL-Authentication Client Library 	
Webservice-Layer #1	Cem Basoglu
<ul style="list-style-type: none"> • Frontend Service Signaturen • Backend Service Signaturen • Dummy Implementierung 	
UI-Layer #1 (Backend)	Veronika Sliesarenko
<ul style="list-style-type: none"> • Backend Client Service (AngularJS) • Crawler <ul style="list-style-type: none"> ◦ Erstellen ◦ Bearbeiten ◦ Löschen ◦ Sperren ◦ Analysesperre 	
UI-Layer #2 (Frontend)	Kevin Schima
<ul style="list-style-type: none"> • Frontend Client Service (AngularJS) • Öffentlicher Monitor 	
Webservice-Layer #2	Cem Basoglu
<ul style="list-style-type: none"> • Frontend Service Implementierung <ul style="list-style-type: none"> ◦ Abrufen aller Beiträge (fester Zeitraum) ◦ Abrufen der Beiträge einer View ◦ Abrufen der Details eines Beitrag ◦ Suchen von Beiträgen • Backend Service Implementierung <ul style="list-style-type: none"> ◦ Authentifizierung ◦ CRUD Crawler ◦ CRUD Analyzer ◦ CRUD Views 	

Milestone 2 (05.12.2016)

Business-Layer #2

Azad Sino

- Business Logic
 - Analyzer
 - Crawler

UI-Layer #3 (Backend)

Veronika Sliesarenko

- Analyzer
 - Erstellen
 - Bearbeiten
 - Löschen
 - Sperren
- Gruppen
 - Erstellen
 - Bearbeiten
 - Löschen
 - Analyzer zuordnen
 - Analyzer Zuordnung aufheben

UI-Layer #4 (Frontend)

Kevin Schima

- Web-Ansicht
 - Übersicht
 - Suchbegriff Filterung
 - Bewertung
- Öffentlicher Monitor
 - QR Code anzeigen

Webservice-Layer #3

Cem Basoglu

- Token basierte Authentifizierung
- QR-Code Generierung
- Crawler Service
 - Beiträge veröffentlichen
- Analyzer Service
 - Noch nicht analysierte Beiträge abrufen (fester Zeitraum)
 - Analyseergebnis zum Beitrag veröffentlichen

Milestone 3 (19.12.2016)

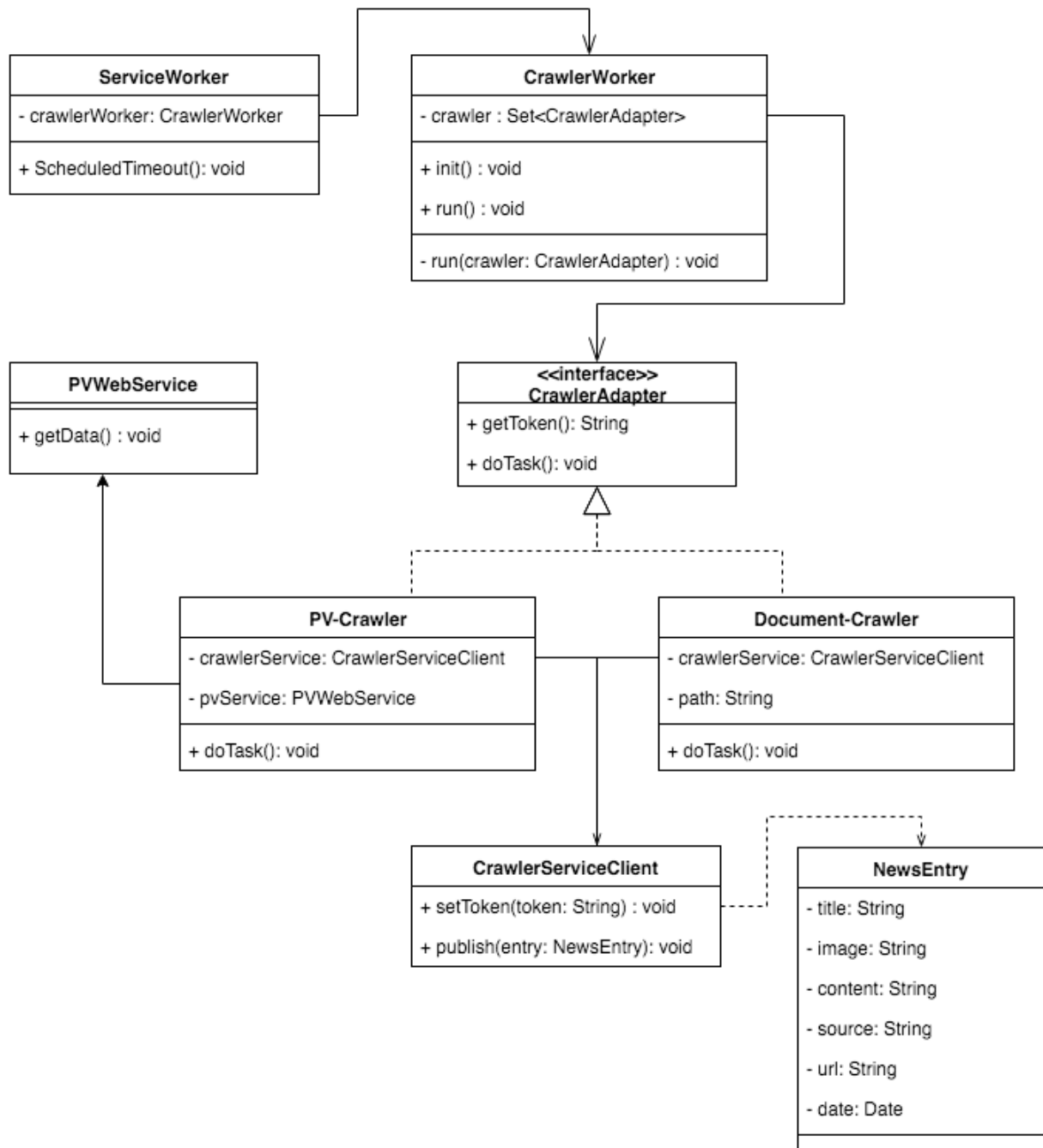
Python Analyzer Client Library	Azad Sino
<ul style="list-style-type: none"> • Abrufen der Beiträge • Analyseergebnisse veröffentlichen 	
UI-Layer #3 (Backend)	Veronika Sliesarenko
<ul style="list-style-type: none"> • Ansichten (öffentlicher Monitor) <ul style="list-style-type: none"> ○ Erstellen ○ Bearbeiten ○ Löschen ○ Anzuzeigende Crawler Ergebnisse einschränken ○ Anzuzeigende Analyzer-Gruppen einschränken ○ Eindeutige URL zur Ansicht anzeigen 	
UI-Layer #4 (Frontend)	Kevin Schima
<ul style="list-style-type: none"> • Web-Ansicht <ul style="list-style-type: none"> ○ Detailansicht ○ Analyseergebnisse (Details) • Mobile-Ansicht optimieren <ul style="list-style-type: none"> ○ Übersicht ○ Bewertung ○ Detailansicht ○ Analyseergebnisse (Details) 	
Apache Nutch Plugin	Cem Basoglu
<ul style="list-style-type: none"> • Crawler Client Library (Java) • Plugin Implementierung <ul style="list-style-type: none"> ○ IndexingFilter ○ IndexWriter 	

Feature Complete (09.01.2017)

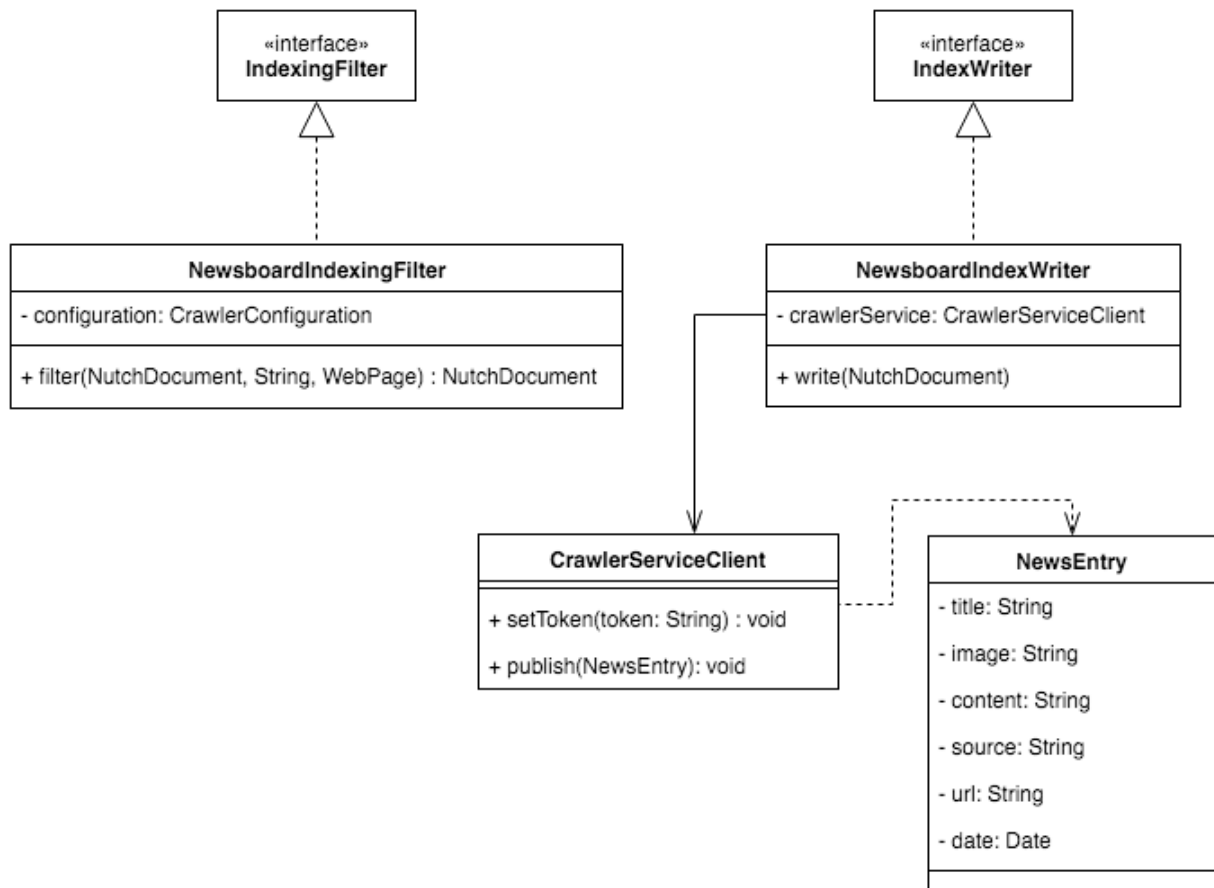
Abschlussarbeiten	alle
<ul style="list-style-type: none"> • Refactoring • Code Convention • Javadoc 	
Dokumentation	Veronika Sliesarenko, Kevin Schima, Azad Sino, Cem Basoglu
<ul style="list-style-type: none"> • Bedienungsanleitung der Administrationsoberfläche • Installationsanleitung <ul style="list-style-type: none"> ◦ Konfiguration Payara ◦ Installation Web-Crawler • Javadoc Generierung 	
Packaging / Deployment	Cem Basoglu Azad Sino
<ul style="list-style-type: none"> • Deployment Archiv 	

Klassendiagramm

Crawler



Crawler - Apache Nutch



Analyzer

