

VSDB

# **NOSQL**

Alexander Rieppel      Dominik Backhausen

30. März 2014

5AHITT

# Inhaltsverzeichnis

<b>1</b>	<b>Aufgabenstellung</b>	<b>3</b>
1.1	Anforderungen . . . . .	3
<b>2</b>	<b>Designüberlegungen</b>	<b>5</b>
2.1	Zu implementierende Anforderungen . . . . .	5
2.2	3er- und 4er-Gruppen Anforderungen . . . . .	5
<b>3</b>	<b>Arbeitsaufteilung</b>	<b>6</b>
<b>4</b>	<b>Arbeitsdurchführung</b>	<b>7</b>
4.1	Resultate . . . . .	7
4.2	Niederlagen . . . . .	7
<b>5</b>	<b>Testbericht</b>	<b>8</b>
<b>6</b>	<b>Quellen</b>	<b>9</b>

# 1 Aufgabenstellung

Entwerfen Sie ein System zur Verwaltung von alten Prüfungsfragen und Aufgabenstellungen. Beachten Sie dabei eine einfache und modulare Art der Datenspeicherung. Es soll die Möglichkeit bestehen, Fragen zu unterschiedlichen Hauptkategorien zuzuordnen (Unterrichtsfächer bzw. Kompetenzbereich, z.B. INSY, SYT bzw. "Dezentrale Systeme etc.). Weiters soll es aber auch möglich sein den Fragen und Aufgaben weitere Informationen beizufügen (Tags, z.B. JEE, Datawarehouse, XML, Hibernate etc.). Fragen werden Kandidaten bzw. Klassen zur Vorbereitung an einem bestimmten Termin zugewiesen. Die Liste aller Fragen und Aufgaben soll sortierbar sein und als solche auch ausgedruckt werden können. Da die endgültige Definition der Archivierung nicht abgeschlossen ist, muss eine einfache Erweiterungsmöglichkeit gegeben sein.

Die Anzeige und Verwaltung der Fragen soll nur authentifizierten Benutzern gestattet werden.

Achten Sie beim Design und der Implementierung des Systems auf Modularität und Erweiterbarkeit. Verwenden Sie für die graphische Visualisierung ebenfalls gängige Frameworks, wie zum Beispiel Bootstrap . Verwenden Sie für die Implementierung ein passendes Web-Framework (Python, Java, PHP, ...) und ein der Aufgabenstellung entsprechendes NoSQL-Datenbankmanagementsystem (MongoDB, ...).

Gruppenarbeit von 2 bis 4 Mitgliedern, wobei die Anforderungen (graphische Oberfläche, funktionale Anforderungen) der Anzahl der Gruppenmitgliedern angepasst wird. Erweiterungen werden in Klammern entsprechend deklariert (z.B. 4er Gruppe muss es zusätzlich implementieren, die anderen müssen es nicht realisieren aber die technische Möglichkeit im Protokoll erörtern), sonst ist es ein Must-Have-Requirement.

## 1.1 Anforderungen

- Anzeige / Editierbarkeit [multiplizität]

Aufgabenart (Maturafrage, praktische Aufgabenstellung, schriftliche Mitarbeitensüberprüfung, Jahresprüfung, ...) [1..\*]

Kategorie (SYT, INSY, Dezentrale Systeme, ...) [1..\*]

Tags (JEE, Hibernate, XML, ...) [0..\*]

Author [1..\*]

erstellt / zuletzt geändert [1]

zuletzt verwendet [0..\*]

zugeteilt [0..\*]

Angabentext [1]

Hinzufügen von zusätzlichen Keys und deren Values (4er)

Zusätzliche Attachments (Bilder, Videos, etc.) (3er)

- Suche

Aufgabenart, Kategorie und Tags

Datum

Volltextsuche über alle Values (4er)

- Administration

Benutzer verwalten

Logging von Benutzeraktionen (4er)

Snapshots und Wiederherstellung der Datenbasis (3er)

## 2 Designüberlegungen

### 2.1 Zu implementierende Anforderungen

Als Datenbank wird MongoDB verwendet und für die Administrationsoberfläche wird ein Servlet erstellt, welches in Java EE implementiert wird. Die Darstellung wird mit dem Framework Twitter Bootstrap 3 erfolgen. Für die Speicherung von Aufgabenart, Kategorie, Tags, Autor und Angabentext, werden simple Strings in der Datenbank abgelegt. Für erstellt/zuletzt geändert sowie zuletzt verwendet wird ein Datum im entsprechenden Format abgespeichert. Die Suche wird über die Datenbank mit Abfragen laufen. Die Benutzerverwaltung läuft über eine eigene Administrationsseite ab, die über den Browser aufgerufen werden kann.

### 2.2 3er- und 4er-Gruppen Anforderungen

Für das Hinzufügen von zusätzlichen Keys und deren Values wird der Datensatz gesucht und ein update auf diesen durchgeführt. Für zusätzliche Attachments von z.B. Bildern und Videos, wird einfach ein Button erstellt hinter den das entsprechende File per link angehängt wird. Dies könnte unter Umständen auch ein FTP Link sein. Die Volltextsuche würde im Prinzip genauso funktionieren wie die normale Suche, nur dass die Suchparameter bei der DB-Abfrage erweitert werden müssen. Das Logging der Benutzeraktionen könnte z.B. über ein eigenes Framework realisiert werden (Log4j). Snapshots und Wiederherstellung der Datenbasis würde in erster Linie über die Datenbank selbst funktionieren, wobei eine entsprechende Verwaltung der Snapshots wiederum, über JavaEE möglich wäre. Siehe hierzu auch: MongoDB Replikation Manual

### 3 Arbeitsaufteilung

Name	Arbeitspaket	Time Estimated	Time Spent
Alexander Rieppel	Datenbank installieren	1h	0.5h
Alexander Rieppel	Datenbank konfigurieren	1h	1h
Alexander Rieppel	JavaEE Servlet schreiben	2h	1h
Alexander Rieppel	Seiten Design mit Bootstrap	2h	3h
Dominik Backhausen	JavaEE Servlet schreiben	2h	4h
Dominik Backhausen	Datenbank Schnittstelle schreiben	2h	1.5h
Gesamt		10h	11h

# 4 Arbeitsdurchführung

## 4.1 Resultate

- MongoDB wurde erfolgreich installiert und konfiguriert
- Das Servlet wurde funktionalistisch so wie in der Designüberlegung angegeben implementiert
- Die Speicherung der Strings funktioniert problemlos
- Die Daten für erstellt, zuletzt verwendet etc. werden einwandfrei angezeigt und gespeichert
- Die Suche funktioniert so wie in der Designüberlegung angegeben
- Die Benutzerverwaltung läuft über eine Adminseite
- Das Design wurde mit Hilfe von Twitter Bootstrap 3 implementiert

## 4.2 Niederlagen

- Keine Niederlagen

## 5 Testbericht

Sobald das Servlet und der Dienst von MongoDB gestartet ist, kann auf der lokalen Maschine folgende Seite aufgerufen werden: `localhost:8080/NOSQL/Aufgaben/admin`. Hier ist zunächst ein Login notwendig, wo sich mit dem Default-Administrator Benutzer `admin` und dem Passwort `admin` eingeloggt werden muss. Durch Eingabe eines Benutzernamens und eines Passworts können hier neue Benutzer angelegt werden. Zusätzlich hat man hier die Möglichkeit nach bestimmten vorhandenen Benutzern zu suchen und Benutzer zu löschen. Sollte der gewünschte Benutzer bereits vorher angelegt sein, kann der Schritt des neuen Benutzers übersprungen und direkt die Seite `localhost:8080/NOSQL/Aufgaben` aufgerufen werden. Nach erfolgreichem Login mit dem entsprechenden Nutzer erscheinen die Datenbankeinträge, wobei mit einem Klick auf das Plus in der oberen Leiste ein neuer Datensatz erstellt werden kann. In dieser Ansicht werden alle erforderlichen Informationen eingegeben und danach kann der Datensatz abgespeichert oder durch einen Klick auf zurück verworfen werden. Analog hierzu funktioniert die Bearbeitungsfunktion durch einen Klick auf den Stift. Auch existiert hier die Möglichkeit eine Suche nach Art, Kategorie oder bestimmten Tags, bzw. nach Datum durchzuführen. Auch können hier Datensätze durch einen Klick auf das Kreuz gelöscht werden. Zuletzt ist es möglich sich durch den Logout-Button in der rechten oberen Ecke auszuloggen.



## 6 Quellen

- [1]<http://docs.mongodb.org/manual/reference/>
- [2]<https://www.mongodb.org/>
- [3]<http://api.mongodb.org/java/>
- [4]<http://getbootstrap.com/>
- [5]<http://getbootstrap.com/components/>