

VSDB

# **NOSQL**

Alexander Rieppel      Dominik Backhausen

27. 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>
<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**

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

### 3 Arbeitsaufteilung

Name	Arbeitssegment	Time Estimated	Time Spent
Alexander Rieppel	Datenbank installieren	1h	0.5h
Alexander Rieppel	Datenbank konfigurieren	1h	1h
Alexander Rieppel	JavaEE Servlet schreiben	2h	1h
Dominik Backhausen	JavaEE Servlet schreiben	2h	3h
Dominik Backhausen	Datenbank Schnittstelle schreiben	2h	1.5h
Gesamt		8h	7h

## **4 Arbeitsdurchführung**

## 5 Testbericht



## 6 Quellen

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