

# Projektdefinition sqrrl

**Web-Applikation zum Üben und automatisierten Korrigieren von SQL**

Alexander-Michael Kühnle      Matthias Michael Döpmann  
Mark Umnus

5. Mai 2019

# Inhaltsverzeichnis

<b>1. Übersicht</b>	<b>4</b>
1.1. Anforderungen . . . . .	4
1.1.1. Funktionale Anforderungen . . . . .	4
1.1.2. Nichtfunktionale Anforderungen . . . . .	4
1.2. Termine . . . . .	5
<b>2. Konkrete Umsetzung</b>	<b>6</b>
2.1. Verwendete Technologien . . . . .	6
2.2. Liefergegenstände . . . . .	6
<b>A. Protokolle</b>	<b>7</b>

## Erläuterungen

Fragen an Betreuer

Fragen an uns

## Einführung

Dieses Dokument dient der Planung und Durchführung des im Titel genannten Projektes im Rahmen des Moduls „*Softwareentwicklungsprojekt I*“. Im Kapitel 1.1 werden die Anforderungen aufgezählt, die an dieses Projekt gestellt werden. Kapitel 1.2 zeigt den initial erstellten Zeitplan.

# 1. Übersicht

## 1.1. Anforderungen

### 1.1.1. Funktionale Anforderungen

- Kontenverwaltung von Studierenden und Dozierenden
- Dozierende können Aufgaben erstellen
- Studierende können Aufgaben bearbeiten
  - Aufgabenstellung in natürlicher Sprache
  - Entgegennahme von SQL-Anfragen
  - Anzeige des Ergebnisses der Anfrage (auf zufälliger/ Beispieltabelle)
  - Erkennung der Korrektheit der Anfrage bezüglich der gestellten Frage
  - bei Unsicherheit → Ersteller informieren
- freies Üben
- Studierende sollen virtuelle Abzeichen erwerben können → von Dozenten erstellt
- eigene Datenbank für jeden Studenten (i.e. Sollen Änderungen an der „Spiel-DB“ auch nach dem Ausloggen erhalten bleiben?)
- Adminaccount zur Benutzerverwaltung?

### 1.1.2. Nichtfunktionale Anforderungen

**Einfachheit** Das Programm soll durch Benutzer ohne das Lesen einer Anleitung bedienbar sein.

**Wartbarkeit** Das Programm soll auch für projektexterne Entwickler verständlich, wartbar und erweiterbar sein.

**Qualität** Das Programm soll durch Tests grundlegenden Qualitätsansprüchen genügen.

**Freiheit** Das Programm soll unter Linux funktionieren und möglichst nur auf freie/offene Software zurückgreifen.

## 1.2. Termine

Datum	Ziele
01.05.2019	Technologien bestimmt
08.05.2019	Technologien installiert und lauffähig; Zeit- und Architekturplan erstellt
15.05.2019	
22.05.2019	Prototypen erstellt; Vortragsfolien erstellt
29.05.2019	Konzept für automatische Kontrolle erarbeitet
05.06.2019	
12.06.2019	
19.06.2019	
26.06.2019	
03.07.2019	Projekt fertig; Abschlussvortragsfolien erstellt
10.07.2019	kleine Verbesserungen (z. B. Dokumentation)

## 2. Konkrete Umsetzung

### 2.1. Verwendete Technologien

- Ruby **Version 2.5.5?**
- Ruby on Rails **Version 5.2.3?**
- Apache HTTP Server **Version?**
- PostgreSQL **Version?**
- Bootstrap **Version?**
- weitere noch zu bestimmende JS-/CSS-Bibliotheken

### 2.2. Liefergegenstände

Hier sind lediglich Anforderungen aufgelistet, die vom Betreuer gefordert und akzeptiert wurden.

#### **A Benutzerverwaltung**

ID	Liefergegenstand	Erläuterung
A.1	Studierendenaccounts	
A.1.1	Nutzernamen	
A.1.2	Passwörter	
A.1.3	Aufgabenübersicht	Markierung für Bearbeitungsstand
A.1.4	Abzeichenübersicht	
A.2	Dozentenaccounts	
A.2.1	Nutzernamen	
A.2.2	Passwörter	
A.2.3	Aufgabenübersicht	Markierung für Bearbeitungsstand und Fragen/Probleme
A.2.4	Studentenübersicht	

#### **B Aufgabenbearbeitung**

# A. Protokolle

## 01. Mai 2019

### Ziele

- zu verwendende Technologien bestimmt
- grundlegenden Zeitplan entworfen

## 08. Mai 2019

### Ziele

- Programmbibliotheken installiert und eingerichtet
- Liefergegenstände spezifiziert
- Zeitplan daran angepasst
- Zuständigkeiten geklärt

## 15. Mai 2019

### Ziele

- Prototypen erstellt
  - Dozenten registrieren
  - Aufgabe anlegen
  - Studenten registrieren
  - Aufgabe auswählen
  - Query eintragen
  - Ergebnis wird angezeigt zusammen mit "Richtig!"
- Vortrag ausgearbeitet
  - Aufgabenstellung
  - Vorgehen/Vision
  - Stand
  - Zukunftspläne
  - Vortragsfolien erstellt