



Dokumentation

Sprechstunden Internet Verwaltung

cdeinlein, tdubovik, spahl, ndiadowitz



Inhaltsverzeichnis

1	Vorwort	2
2	Funktionale Anforderungen	3
3	Style-Guide	4
3.1	Farbschema	4
3.2	Typographie	4
3.3	Layout	4
3.4	Navigation	4
4	Datenbankmodell	5
5	REST-Schnittstelle	6
5.1	Überblick	6
5.2	Details	7
5.2.1	Allgemein	7
5.2.2	Login, Register und Reset Password	7
6	Implementierung	8
7	Allgemein	9
8	Testfälle	10
8.1	Backend	10
8.2	Frontend	10
9	Benutzerhandbuch	11
9.1	Installation für Entwicklung und zum Ausprobieren (unter Ubuntu 17.04)	11
9.2	Dozenten aktualisieren	11



1 | Vorwort

In dieser Studienarbeit geht es um die Erstellung einer Website zur Sprechstundenverwaltung an der Hochschule Hof. Als Projektname wurde Schiv gewählt das eine Abkürzung von (S)pre(ch)stunden (I)nternet V(erwaltung) ist. Für die Implementierung dürfen folgende Hilfsmittel verwendet werden:

- HTML 5
- CSS 3
- Javascript (Angular JS, Bootstrap, Ajax, jQuery)
- PHP (Laravel)
- MySQL



2 Funktionale Anforderungen

Eine Liste von Funktionalen Anforderungen die während des Interviews mit dem Auftraggeber beschlossen wurden:

1. Konten sind nur registrierbar mit E-Mail-Adressen (@hof-university.de) der Hochschule Hof.
2. Zur Registrierung wird nur die E-Mail-Adresse und ein Passwort benötigt.
3. Konten werden nach der Registrierung als „Studenten“ behandelt.
4. Eine Liste der aktiven Dozenten wird über die Schnittstelle der iOS-Stundenplan-App¹ abgefragt. Hierbei muss aus den Namen des Dozenten die E-Mail-Adresse abgeleitet werden, da die diese nicht direkt abfragbar ist. Für jeden Dozenten wird ein deaktiviertes Konto angelegt.
5. Konten können nur „Dozent“ werden, wenn bereits ein deaktiviertes Konto diesen Typs angelegt ist.
6. Jedem Konto sind ein oder mehrere Fakultäten zugeordnet.
7. Aktivierung eines Kontos erfolgt über eine E-Mail die nach der Registrierung mit einem Aktivierungs-Link geschickt wird.
8. Die Anwendung soll mindestens auf normalen Desktop-PCs mit den weitverbreitetsten Browsern laufen.
9. Es werden generell keine Datensätze gelöscht, sondern nur weich gelöscht („unsichtbar gemacht“).
10. Am Ende eines Semesters werden Studenten aus dem System gelöscht und müssen sich dann wieder erneut registrieren
- 11.

¹<https://github.com/HochschuleHofStundenplanapp/iOS-App/wiki/Schnittstellen-zum-Server>



3 | Style-Guide

3.1 Farbschema

(TODO)

Es wird sich am Farbschema der Hochschule Hof orientiert.

3.2 Typographie

(TODO)

Als Schriftart wird Roboto¹ verwendet, da diese klar lesbar ist.

3.3 Layout

(TODO)

Beim Layout muss zwischen Student und Dozent unterschieden werden. Das Layout der Seite soll möglichst einfach und übersichtlich sein.

3.4 Navigation

Es wird nur eine Navigationleiste am oberen Rand geben, die immer sichtbar ist. Alle Funktionen sind darüber mit wenigen Klicks erreichbar.

¹<https://fonts.google.com/specimen/Roboto?selection.family=Roboto>



4 Datenbankmodel

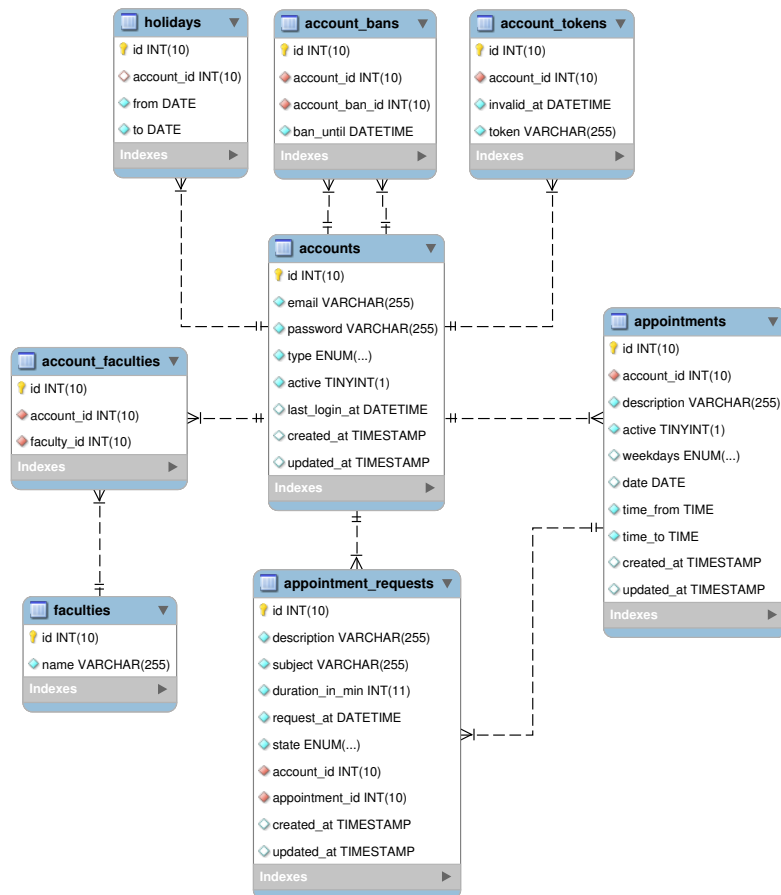


Abbildung 4.1: Datenbankmodel von schiv (Erstellt mit mysql-workbench)



5 REST-Schnittstelle

5.1 Überblick

Tabelle 5.1: Zeigt den Aufbau der REST-Schnittstelle

Beschreibung	[A D] Method:Url (Body)
Login	put:login (email, password)
Logout	put:logout (email, password)
Reset Passwort	put:reset (email, password)
Registrieren	post:register (email, password) put:register (token)
Suche	get:docents
Informationen bzw. Termine von Dozenten	get:docents/{docent_id}
Terminanfragen abrufen	D get:appointment_requests
Termin Einschreibung	A post:appointment_requests (...)
Termine annehmen bzw. ablehen	D put:appointment_requests (id, state) A delete:appointment_requests/{id}
Termine Dozenten	D get:appointments D get:appointments/{count} D get:appointments/{from}/{to} D post:appointments (...) D delete:appointments/{appointment_id}
Einstellungen	A get:settings A put:settings (email, password)
Ausgeschlossene Konten	D get:account_bans D post:account_bans (account_ban_id) D delete:account_bans/{id}

Anmerkungen:

- (...) bedeutet das alle Felder, wie im Model definiert, verwendet werden können.
- get:appointments/{from}/{to}: from und to müssen im Datumsformat YYYY-MM-DD HH:II:SS angegeben werden.
- A: Zugriff nur wenn man als Student oder Dozent eingeloggt.
- D: Zugriff nur wenn man als Dozent eingeloggt.



5.2 Details

5.2.1 Allgemein

Die Nachrichten sind im JSON-Format. Im Erfolgsfall kommt der HTTP-Status **200** zurück, im Fehlerfall kann **401**, **404**, **422** oder **500** zurückgegeben werden:

- **401:** (TODO)

```
{"message": "login unsuccessful"}
```

- **404:** Die angefragten Objekte sind nicht in der Datenbank vorhanden, oder es kann auf sie nicht zugegriffen werden, da die Berechtigung fehlt. Als Antwort kommt eine leere Nachricht zurück. Beispiel:

```
[]
```

- **422:** Die Anfrage enthält nicht alle nötigen Felder, oder ein Feld ist falsch formatiert. Die Antwort enthält als Attribute die Feldnamen bei denen ein Fehler festgestellt wurde. Als Wert der Attribute wird ein Array mit Fehlermeldungen zurückgegeben. Beispiel:

Anfrage:

```
{"email": "foo", "password": "foo"}
```

Antwort:

```
{"email": ["validation.required"], "password": ["validation.min.string"]}
```

- **500:** Ein allgemeiner Fehler. Als Antwort wird eine Fehlermeldung im Attribut message zurückgegeben. Beispiel:

```
{"message": "TODO"}
```

Eine Nachricht kann zum genaueren spezifizieren des Fehlers das Attribut „message“ enthalten.

5.2.2 Login, Register und Reset Password

(TODO)



6 | Implementierung



7 | Allgemein

Die REST-Schnittstelle kann manuell über `test.html` ausprobiert bzw. getestet werden. Implementierungsdetails für die REST-Schnittstelle werden bereits in Abschnitt 5.2 beschrieben.



8 | Testfälle

8.1 Backend

Es sind für jeden Controller Testfälle im Verzeichnis tests/Feature angelegt.

8.2 Frontend

(TODO)



9 Benutzerhandbuch

9.1 Installation für Entwicklung und zum Ausprobieren (unter Ubuntu 17.04)

```
1 $ sudo apt update
2 $ sudo apt install git composer unzip php php-mbstring php-xml
3 $ sudo apt install php-sqlite3 sqlitebrowser
4 $ cd $HOME
5 $ git clone https://github.com/studentcstn/schiv
6 $ cd schiv
7 $ composer install
8 $ cp .env.sqlite .env
9 $ touch /tmp/db
10 $ php artisan migrate:refresh --seed
11 $ php artisan serve
12 $ php artisan retrieve:docents <username> <password>
13 $ firefox http://localhost:8000
```

9.2 Dozenten aktualisieren

Am Ende des Semester können die Dozenten-Konten aktualisiert werden. Dazu dient der Befehl `php artisan retrieve:docents`. Es werden alle vorhanden Dozenten-Kontos deaktiviert und das Passwort zurückgesetzt. Nur Dozenten die noch Zugriff auf ihr E-Mail-Konto haben können sich erneut registrieren und anmelden. Die Zugangsdaten für die Schnittstelle der iOS-Stundenplan-App müssen in der `.env`-Datei eingetragen werden (Schlüssel: `IOSAPP_USERNAME` und `IOSAPP_PASSWORD`).

Zur Vereinfachung kann man das Task-Scheduling aktivieren, wie in dieser Anleitung beschrieben <https://laravel.com/docs/5.4/scheduling>. Der Task würde am 15. März bzw. 31. September jedes Jahr ausgeführt.

Zum Testen kann noch die Option `--from-cache` verwendet werden. Hier werden die Daten nur einmal vom Server geholt und dann in einer lokalen Datei `/tmp/docents` zwischengespeichert.