

# Dokumentation Ticketshop MVC

PROJEKTARBEIT MODUL 151 – DATENBANKEN IN  
WEBAUFTTRITT EINBINDEN

KLINGLER JOËL

## Inhalt

|        |                                       |    |
|--------|---------------------------------------|----|
| 1      | Beschreibung .....                    | 3  |
| 2      | Designstudie .....                    | 3  |
| 2.1    | Grundriss .....                       | 3  |
| 2.2    | Detailliert .....                     | 4  |
| 3      | Terminplanung .....                   | 5  |
| 4      | Technologien .....                    | 5  |
| 5      | MVC Ablaufplan .....                  | 6  |
| 6      | Benutzerfluss .....                   | 7  |
| 6.1    | Startseite (Home) .....               | 7  |
| 6.2    | Login .....                           | 7  |
| 6.3    | Registrieren .....                    | 8  |
| 6.4    | Veranstaltung erstellen .....         | 8  |
| 6.5    | Veranstaltung bearbeiten .....        | 9  |
| 6.6    | Veranstaltung löschen .....           | 10 |
| 6.7    | Veranstaltung anzeigen .....          | 10 |
| 7      | Funktionsbeschreibung .....           | 10 |
| 7.1    | Zwingend .....                        | 10 |
| 7.2    | Zusatz .....                          | 10 |
| 7.3    | Ausbau .....                          | 11 |
| 8      | Berechtigungskonzept .....            | 11 |
| 9      | Test-Benutzer .....                   | 11 |
| 10     | Serveranforderungen .....             | 11 |
| 11     | Datenbank .....                       | 12 |
| 12     | Dateistruktur .....                   | 12 |
| 13     | Installationsanleitung .....          | 13 |
| 13.1   | XAMPP .....                           | 13 |
| 13.2   | XAMPP konfigurieren und starten ..... | 13 |
| 13.3   | Website .....                         | 14 |
| 13.3.1 | Datenbank .....                       | 14 |
| 13.3.2 | PhpMyAdmin .....                      | 14 |
| 13.3.3 | Datenbank zurücksetzen .....          | 14 |
| 13.3.4 | Datenbank ändern .....                | 14 |
| 14     | Bedienungsanleitung .....             | 14 |
| 14.1   | Registrieren .....                    | 14 |
| 14.2   | Einloggen .....                       | 15 |

|       |   |    |
|-------|---|----|
| 14.3  | Ausloggen .....                                     | 15 |
| 14.4  | Profil betrachten.....                              | 15 |
| 14.5  | Profil bearbeiten.....                              | 15 |
| 14.6  | Veranstaltung erfassen.....                         | 15 |
| 14.7  | Veranstaltungen in Veranstalter-Sicht anzeigen..... | 16 |
| 14.8  | Veranstaltungen bearbeiten .....                    | 16 |
| 14.9  | Veranstaltung löschen.....                          | 16 |
| 14.10 | Veranstaltungen in Kunden-Sicht anzeigen .....      | 17 |
| 15    | Testing .....                                       | 18 |

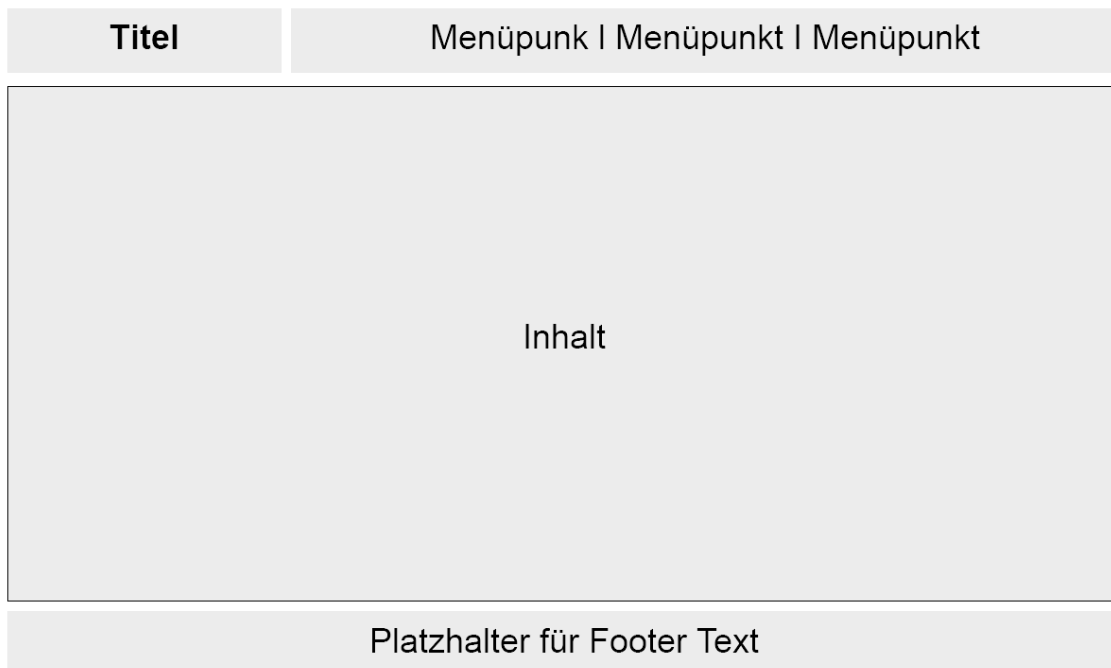
## 1 Beschreibung

Im Rahmen des Modules M151 – Datenbanken in Webauftritt einbinden der Überbetrieblichen Kurse für Informatiker Applikationsentwickler, müssen wir eine Web-Applikation planen und realisieren. Die Web-Applikation muss über einen Login, sowie CRUD-Funktionalitäten (Create, Read, Update, Delete) verfügen. Der Schwerpunkt dieses Modul liegt aber auf dem MVC-Design-Pattern. MVC steht für **M**odel-**V**iew-**C**ontroller. Der Ablaufs-Fluss für die Webapplikation wird in einem weiteren Artikel beschrieben.

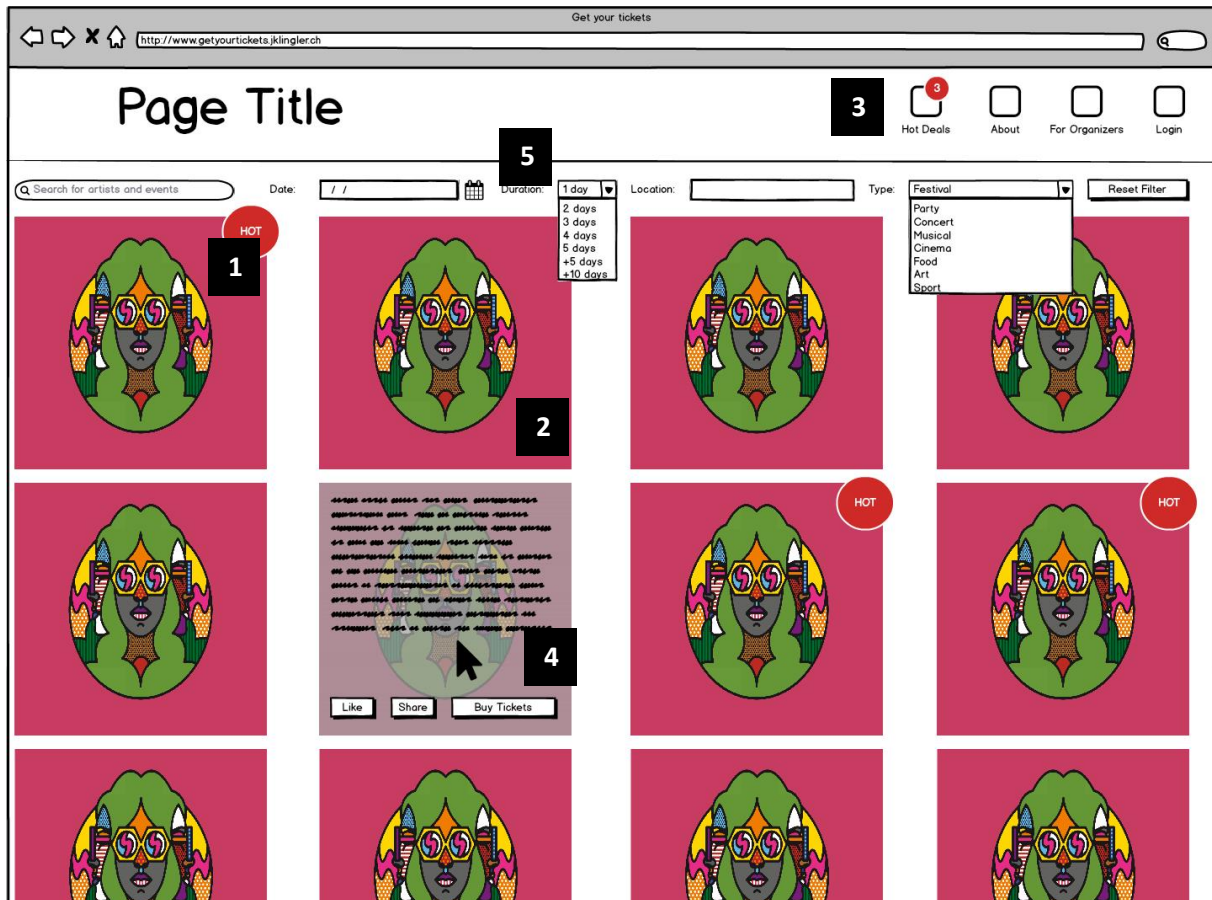
Für diese Projektarbeit wählte ich den Ticketshop, welchen ich im letzten ÜK-Modul M131 entwickelt habe. Die groben Funktionalitäten waren bereits vorhanden und ich brauchte diese bloss in mein MVC-Framework zu implementieren.

## 2 Designstudie

### 2.1 Grundriss



## 2.2 Detailliert



- 1 Hot-Deal: Veranstaltung mit Tickets zum Sonderpreis. (*Nice-to-have*)
- 2 Veranstaltung: Reguläre Veranstaltung.
- 3 Navigation: Für Unangemeldete Besucher, Kunden und Veranstalter unterschiedlich.
- 4 Veranstaltung Hover: Kurzbeschreibung. Funktionen zum Liken, Teilen und Ticketkauf. (*Nice-to-have*)
- 5 Filterbar: Kunden kann nach Veranstaltung suchen, filtern etc. (*Nice-to-have*)

### 3 Terminplanung

| Tag              | Ziel  |
|------------------|---|
| Freitag          | <ul style="list-style-type: none"><li>• Basistemplate-Fertig</li><li>• Datenbankzugriff vorhanden</li><li>• Login + Registrierung</li></ul> |
| Montag           | <ul style="list-style-type: none"><li>• Strukturierung der App fertig.</li></ul>  |
| Dienstag         | <ul style="list-style-type: none"><li>• Veranstaltungen erstellen / anzeigen (Veranstalter) / löschen / bearbeiten fertig.</li></ul>        |
| Abgabe 14:00 Uhr |   |

### 4 Technologien

Die Website wird Serverseitig mit **PHP** entwickelt. Clientseitig kommt **HTML 5**, sowie **JavaScript** mit der **jQuery-Library** zum Einsatz.

<http://php.net/>

<https://www.w3.org/TR/html5/>

<https://www.javascript.com/>

<https://jquery.com/>

Damit das MVC-Design-Pattern nicht von Grund auf neu erfunden werden muss, wird das **CodeIgniter PHP MVC Framework** mit verschiedenen Zusatz-modulen wie «Database» und «Session» verwendet.

<http://www.codeigniter.com/>

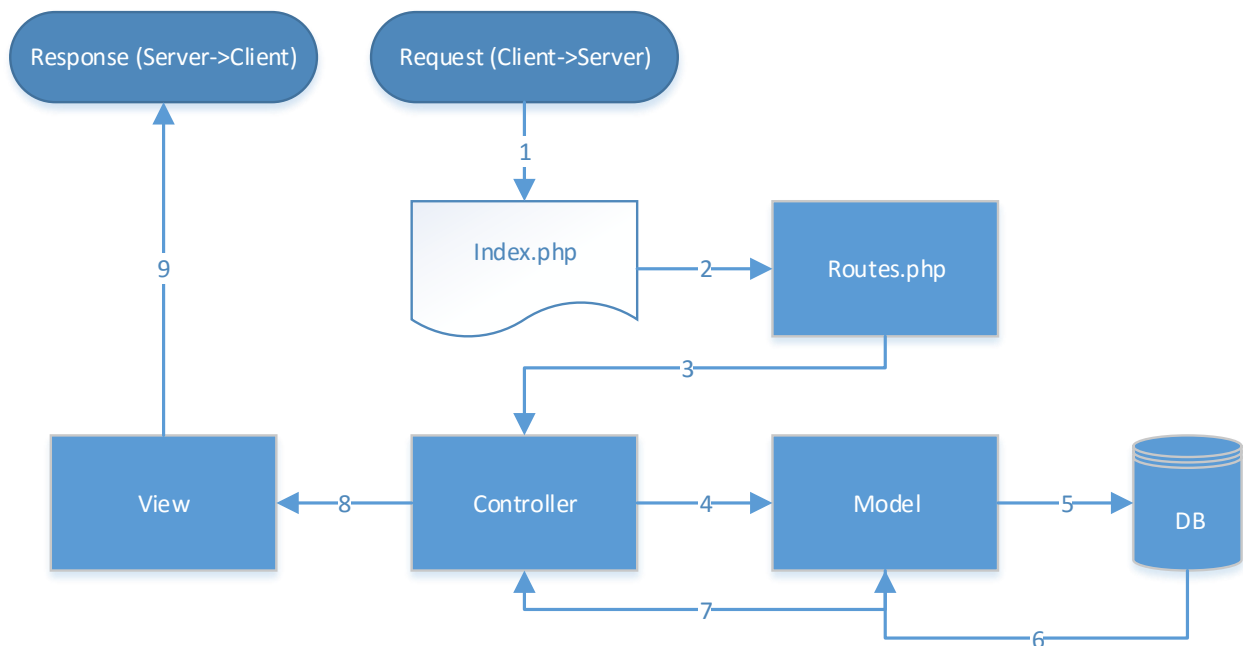
Zum Einsatz kommt die Adaption von Google's Material Design Lite, **Materialize.css**. Dies ist ein CSS-Framework.

<http://materializecss.com/>

Für die Interaktiven Charts wird die JavaScript Library **char.js** verwendet.

<http://www.chartjs.org/>

## 5 MVC Ablaufplan



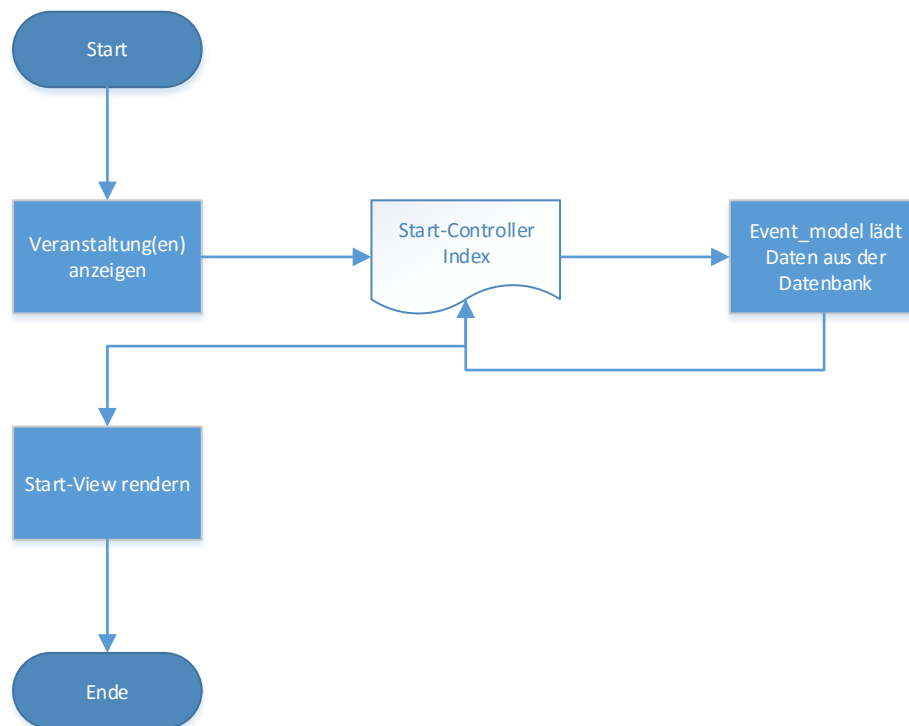
### Legende

1. Der Browser des Users sendet eine HTTP-Request an den Webserver.
2. Das index.php-file leitet die benötigten Informationen (GET / POST – Daten, URL etc.) an das route-File weiter. (Siehe Kapitel X.X Datenstruktur).
3. Das Route-File entscheidet aufgrund der Informationen welcher Controller mit welcher Methode aufgerufen werden soll und leitet die Daten an diesen weiter.
4. Der Controller sendet einen Aufruf an das Model um Daten zu lesen oder zu schreiben.
5. Das Model führt die Datenbankoperation aus.
6. Die Datenbank liefert bei Schreib-Anfragen Daten an das Model zurück.
7. Das Model bereitet die Daten auf und sendet diese an den Controller zurück.
8. Der Controller leitet die Daten an die View. Die View setzt sich aus mehreren Komponenten (Header, Footer, Loader...) zusammen und interpretiert die gelieferten Daten.
9. Das aus der View generierte HTML wird als Response zurück an den Browser des Clients gesendet.

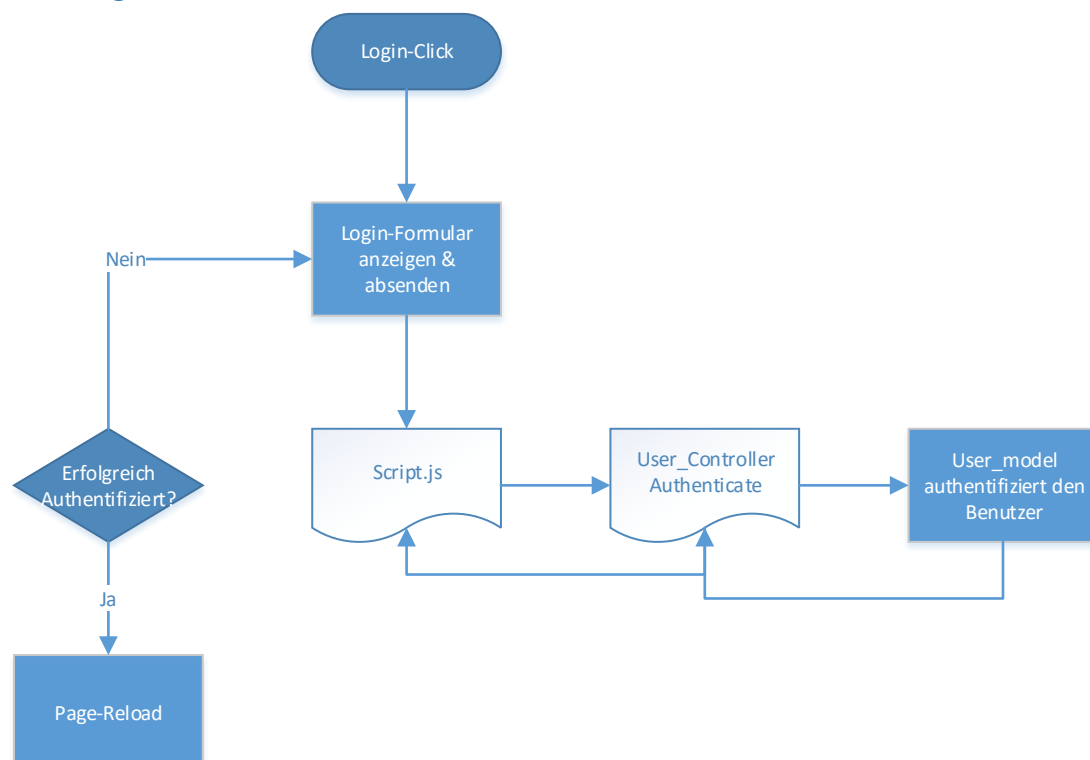
Schritte 4-7 sind optional und werden nur wenn benötigt ausgeführt.

## 6 Benutzerfluss

### 6.1 Startseite (Home)

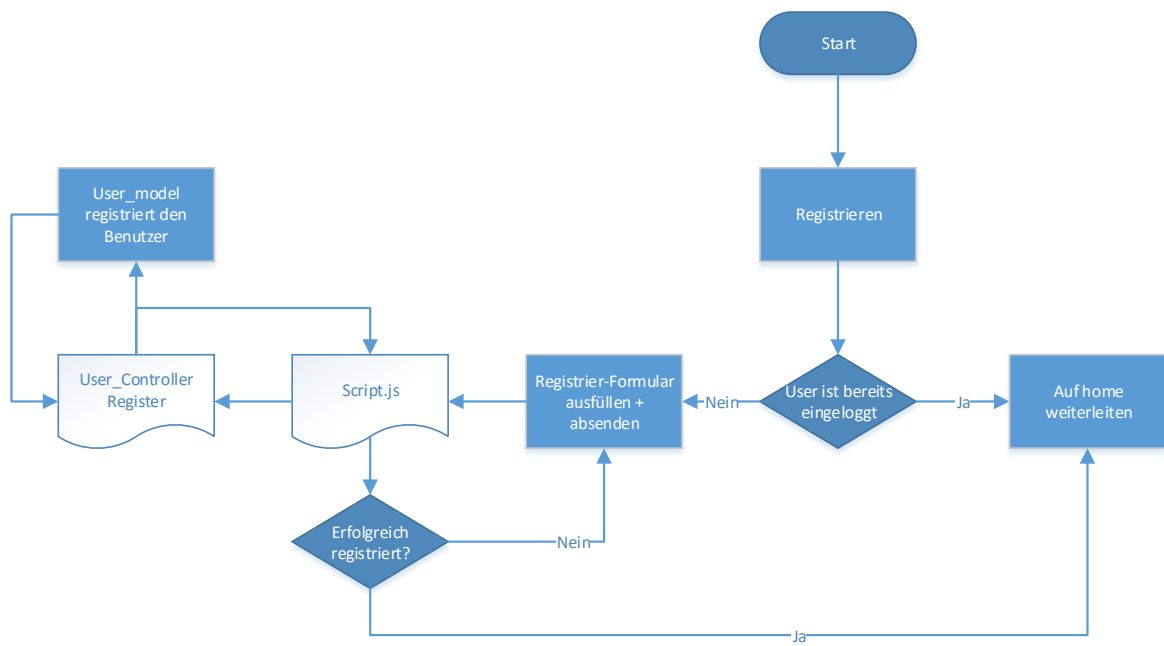


### 6.2 Login

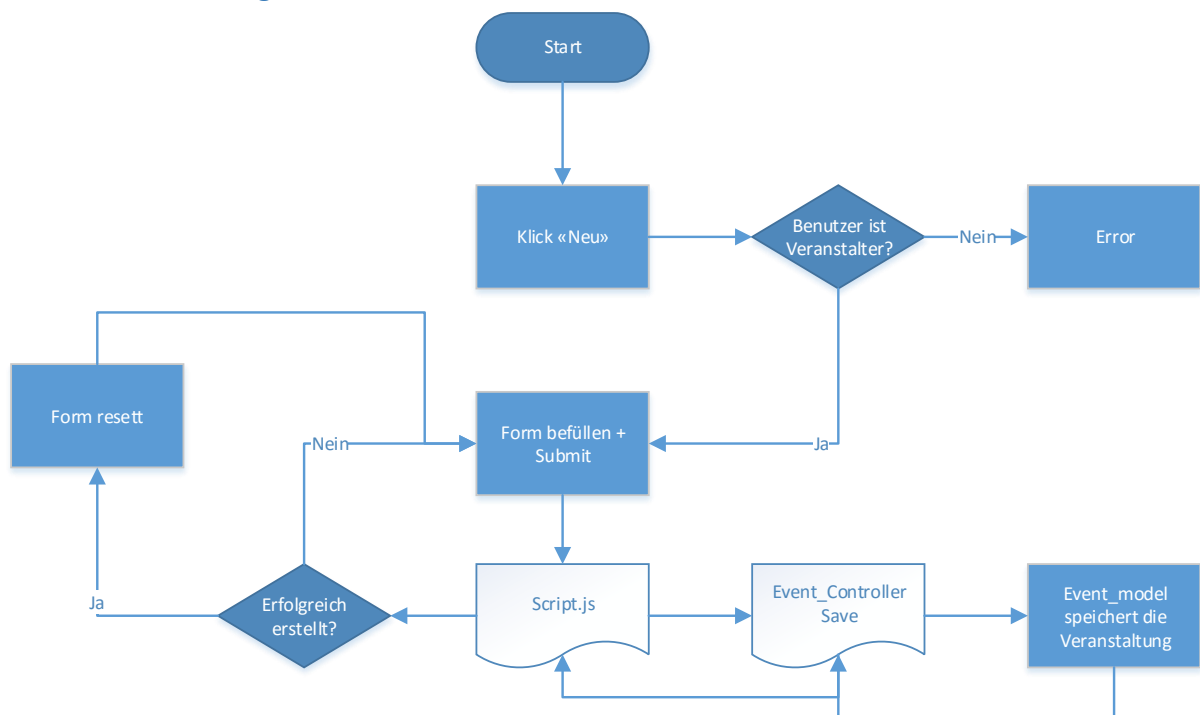




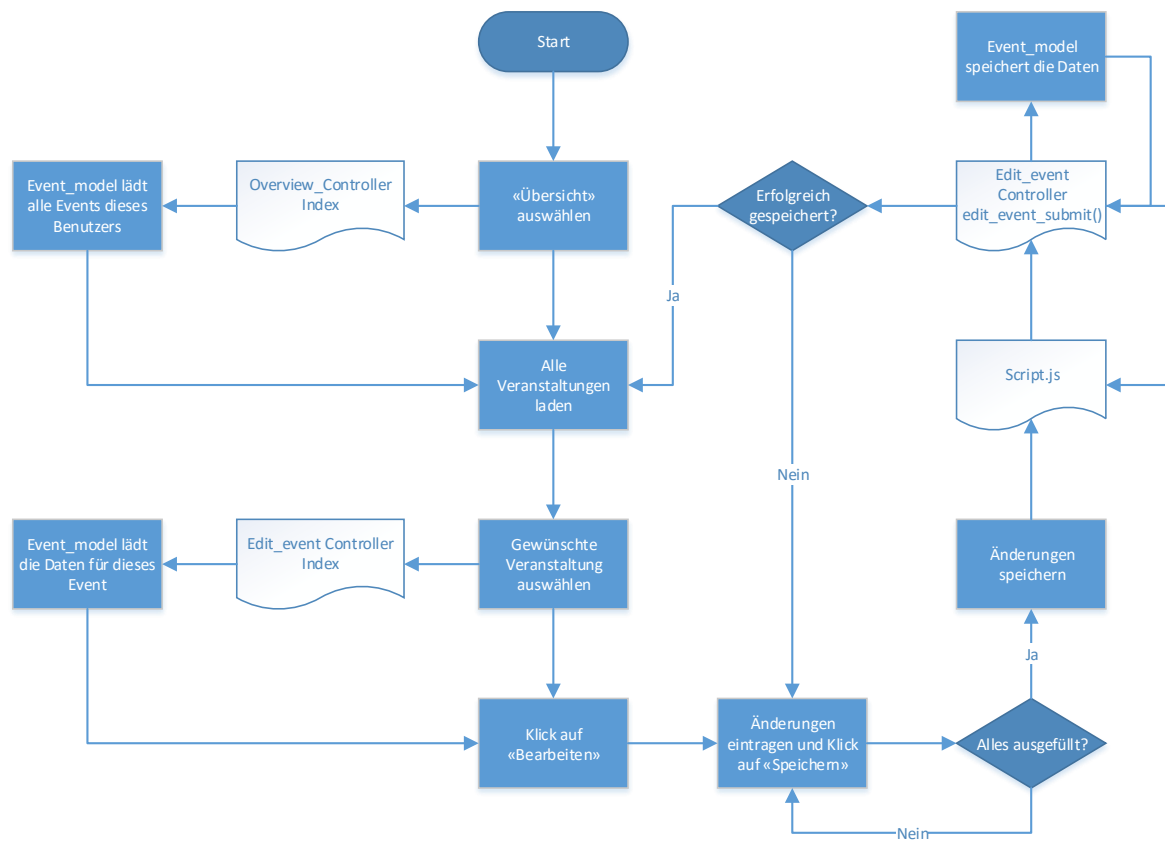
### 6.3 Registrieren



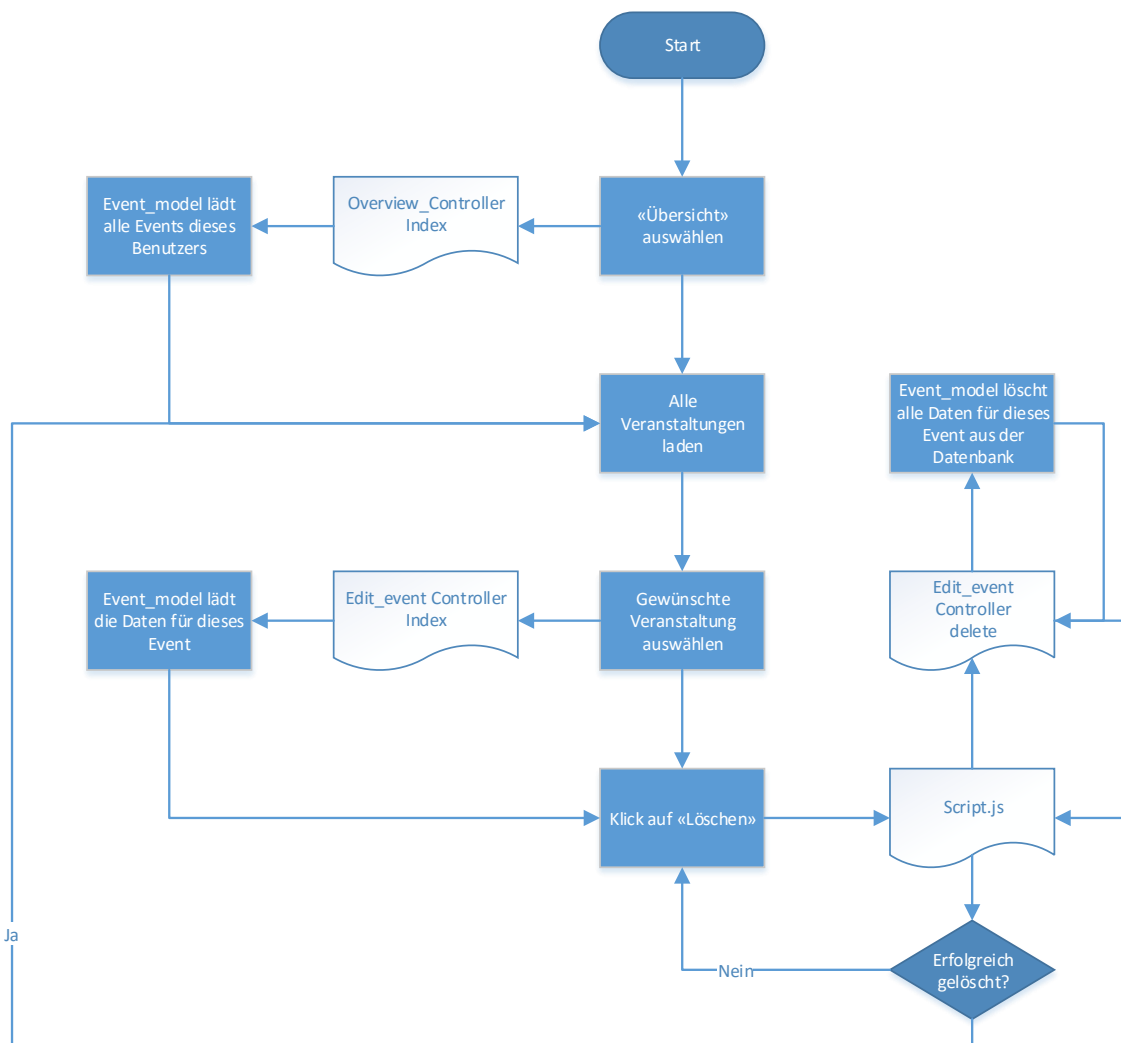
### 6.4 Veranstaltung erstellen



## 6.5 Veranstaltung bearbeiten



## 6.6 Veranstaltung löschen



## 6.7 Veranstaltung anzeigen

Dies wird in «Veranstalter Dashboard», sowie für die Kundenstartseite und die Edit-Event-Seite durchgeführt. Das Model «Event\_model» ist zuständig für das Schreiben und Lesen von Veranstaltungen

## 7 Funktionsbeschreibung

### 7.1 Zwingend

Login / Registrierung

Kunden und Veranstalter können sich Registrieren und Einloggen.

Veranstaltungen CRUD

Veranstalter können Veranstaltungen erstellen, anzeigen, bearbeiten und löschen.

### 7.2 Zusatz

Veranstaltungen kaufen

Kunden können Tickets für eine Veranstaltung erwerben und in Ihrem Profil anzeigen.

Veranstaltung's Dashboard

Veranstalter haben einen Überblick über verkaufte Tickets.

Hot-Deal

Wenn der Ticket-Preis sinkt, werden Veranstaltungen als «Hot Deal» angezeigt.

Filter / Suche

Kunden können Veranstaltungen filtern und nach Kriterien durchsuchen.

### 7.3 Ausbau

Einbinden einer Pay-API

Kunden Bezahlen per PayPal, Kreditkarte etc.

Werbung

Veranstaltungen werden auf externen Seiten durch eine angebotene API beworben.

## 8 Berechtigungskonzept

| Rolle        | Beschreibung                         |
|--------------|--------------------------------------|
| Kunde        | Normal-Sterblicher Kunde.            |
| Veranstalter | Veranstalter. Organisator für Events |

| Rolle        | Funktion  |
|--------------|---|
| Kunde        | Kaufen, Events anzeigen                         |
| Veranstalter | Events erstellen, löschen, bearbeiten, anzeigen |

## 9 Test-Benutzer

| Username  | Rolle        | Passwort  |
|-----------|--------------|-----------|
| Jklingler | Veranstalter | Passw0rd% |
| Kunde     | Kunde        | Passw0rd% |

## 10 Serveranforderungen

PHP (Apache) und MySQL müssen vom Webserver unterstützt werden da:

1. Die Seite PHP basiert ist. (PHP 5.6.15)
2. Daten in einer MySQL-Datenbank liegen.

Die Website wird in XAMPP v3.2.1 programmiert.

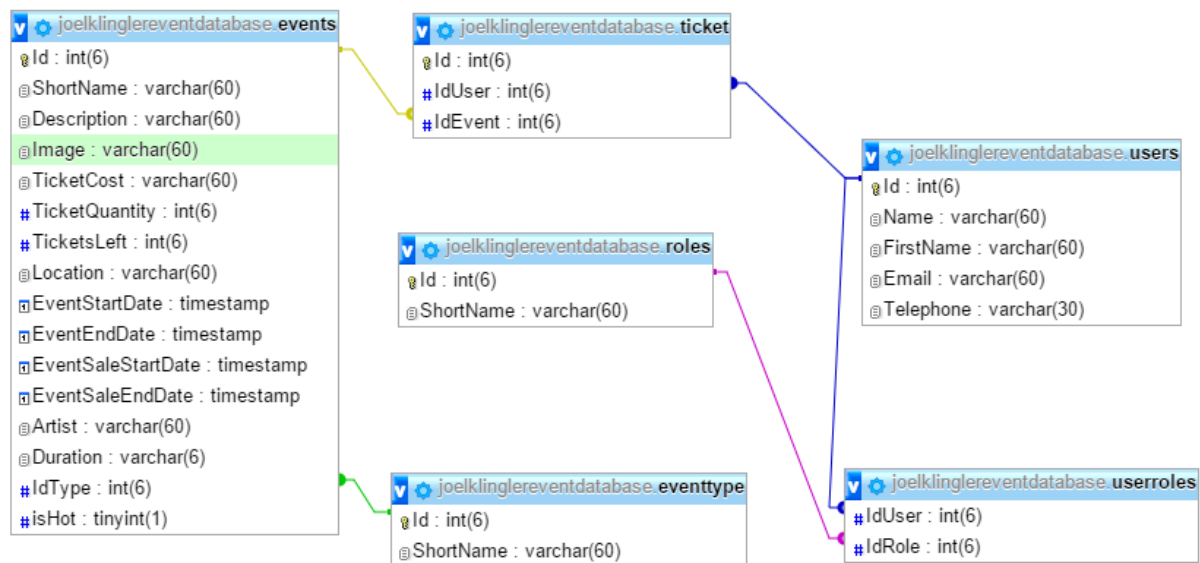
Benötigte Ports:

Apache        80, 443

MySQL        3306

**WICHTIG:** Bei der Projektinstallation install.php bei aktivem Apache und MySQL-Service über den Browser aufrufen. Dieses Skript initialisiert die benötigten Datenbanken, Tabellen und Test-Daten.

## 11 Datenbank



## 12 Dateistruktur

Die Dateistruktur wird auf vier Ebenen genau dokumentiert. Der ebenfalls auf der zweiten Ebene vorhandene «system»-Ordner wird nicht dokumentiert, da dieser für das Projekt irrelevante Daten enthält.

- |                                |   |
|--------------------------------|---|
| - /root (m151/ticketshop_m151) | Root-Pfad (Unterste Ebene)              |
| - application                  | Enthält alle Applikations-Dateien       |
| - cache                        | Caching-Daten                           |
| - config                       | Konfigurationsdateien                   |
| - controllers                  | Die Controller                          |
| - Core                         | Core-Module. Integration Framework      |
| - helpers                      | Helper-Klassen (DB, Utility...)         |
| - hooks                        | Hooking-Files für Framework manip.      |
| - Language                     | Multi-Language                          |
| - English                      |   |
| - Libraries                    | Templates und zusatz-funktionen         |
| - Logs                         | Log-Files                               |
| - Models                       | Die Models.                             |
| - third_party                  | Ordner für third-party-produkte.        |
| - Views                        | Die Views                               |
| - Errors                       | Die Error-Views                         |
| - Partial                      | Die partiellen Views (Login, Loader...) |
| - assets                       | Eigene Anpassungen und Files            |
| - css                          | Cascading-Style-Sheet Files             |
| - font                         | Schriftarten                            |
| - material-design-icons        | MDL Icons für Buttons etc.              |
| - roboto                       | Roboto Content-Schrift                  |
| - img                          | Bilder                                  |
| - js                           | JavaScript-Files                        |
| - php                          | PHP-Files                               |

## 13 Installationsanleitung

Damit die Website lauffähig ist müssen XAMPP, sowie die Website selbst Installiert werden.

### 13.1 XAMPP

1. XAMPP Herunterladen von <https://www.apachefriends.org/de/index.html>
2. Die Heruntergeladene .exe starten
3. Die Meldung bestätigen (XAMPP ist nicht schädlich für Ihren Computer)
4. Den Setup-Anweisungen folgen.

Sobald XAMPP installiert worden ist geht es weiter mit:

### 13.2 XAMPP konfigurieren und starten

Öffnen Sie Ihr XAMPP-Installationsort. Dies ist meistens **C:\xampp**. Navigieren Sie zu **xampp\htdocs**.

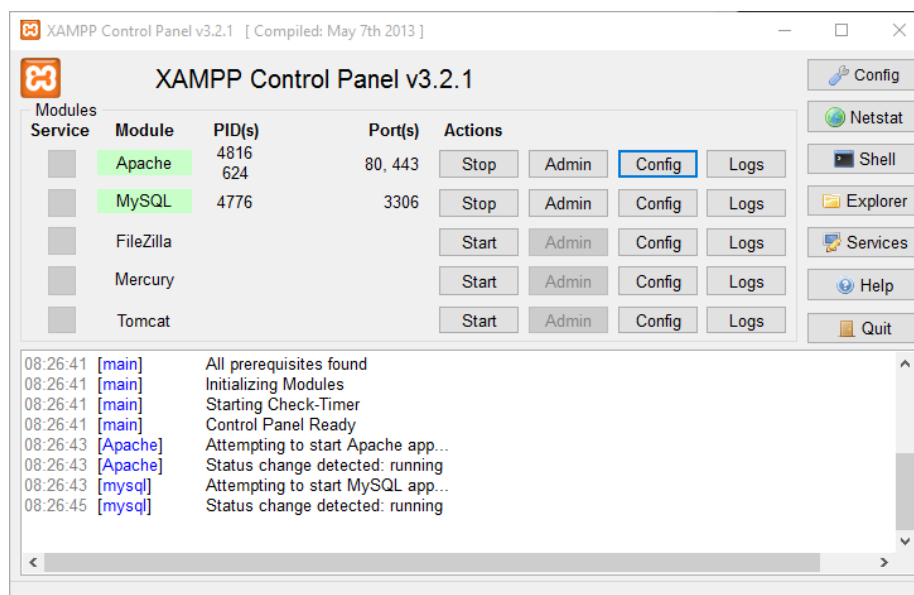
Kopieren Sie den Ordner «Ticketshop» in diesen **htdocs-ordner**.

Navigieren Sie zurück zu **xampp** und starten Sie das **xampp-control.exe**.

Starten Sie die Dienste **Apache** (Webserver) und **MySQL** (Datenbank). Eventuell müssen Sie die Firewall ausschalten oder gewisse Ports freigeben. Skype und IIS können den Port 80 des Apache-Servers bereits besetzen. Sie haben nun drei Möglichkeiten:

1. Den Prozess, welcher den Port besetzt beenden (PID in Taskmanager). ACHTUNG: Gewisse Prozesse sind für die Funktionalität des Betriebssystems essentiell.
2. Besetzt IIS den Port 80 kann IIS auf einen anderen Port um konfiguriert werden. ACHTUNG: Gewisse Web-Dienste können dadurch eingeschränkt werden und müssen ebenfalls angepasst werden.
3. XAMPP Port um konfigurieren.
  1. Klicken Sie auf den Button «Config» in der Apache-Spalte im xampp-Control.
  2. Klicken Sie auf **Apache (httpd.conf)**.
  3. Scrollen Sie zur Zeile «Listen 80» und Ändern Sie diese zu «Listen 8080»  
ACHTUNG: Webseiten welche auf Apache-xampp laufen sind nun nur noch über Port 8080 erreichbar.  
Beispiel eines Seitenaufrufes: localhost:**8080**/ticketshop

Das XAMPP-Control Panel sollte nun folgendermassen aussehen:



### 13.3 Website

Vergewissern Sie sich, dass der Ordner «**ticketshop**» imhtdocs Ordner des XAMPP-Verzeichnis befindet. (Siehe [Kapitel 10.2 XAMPP konfigurieren und starten](#)).

Navigieren Sie in Ihrem Browser zur folgender URL: **localhost/ticketshop**

Sollte auf Ihrem MySQL-Server noch keine Datenbank existieren, wird die Website beim ersten Aufruf selbständig eine Datenbank mit Tabellen und Testdaten anlegen. Die Benutzer, welche in [Kapitel 3 Test-Benutzer](#) beschrieben sind, werden nun ebenfalls erstellt.

#### 13.3.1 Datenbank

Die Datenbank «joelklinglereventdatabase» umfasst folgende Tabellen:

| Datenbanktabelle | Funktion   |
|------------------|--|
| events           | Alle Veranstaltungen   |
| eventtype        | Die zur Auswahl stehenden Veranstaltungs-Typen.                    |
| eventtypes       | Assoziation zwischen Veranstaltung und ihren Veranstaltungs-Typen. |
| orders           | Die Ticket-Käufe.  |
| ticket           | Die Tickets.   |
| userroles        | Die verschiedenen User-Rollen.                                     |
| users            | Die Benutzer.  |

Die Datenbank umfasst fünf Veranstaltungen als vordefinierte Test-Daten. Diese können als Veranstalter auf der Übersicht-Seite bearbeitet und gelöscht werden.

#### 13.3.2 PhpMyAdmin

Sollten Sie manuelle Änderungen an der Datenbank vornehmen wollen, können Sie die Datenbankoberfläche über **localhost/phpmyadmin** aufrufen. Beachten Sie das Strukturelle Änderungen an der Datenbank Einfluss auf die Funktionsfähigkeit der Website haben.

#### 13.3.3 Datenbank zurücksetzen

Um die Datenbank wieder auf ihre Ursprungswerte zurückzusetzen muss die Datenbank entfernt werden. Dies geschieht beim zurücksetzen der Daten automatisch. Klicken Sie dazu auf den Link «**Datenbank initialisieren**» im Footer. Sie erhalten Informationen zum Fortschritt der Operation und können anschliessend die Seite neu laden.

#### 13.3.4 Datenbank ändern

Wird eine Änderung an der Datenbank vorgenommen, müssen diese an zwei Orten eingetragen werden:

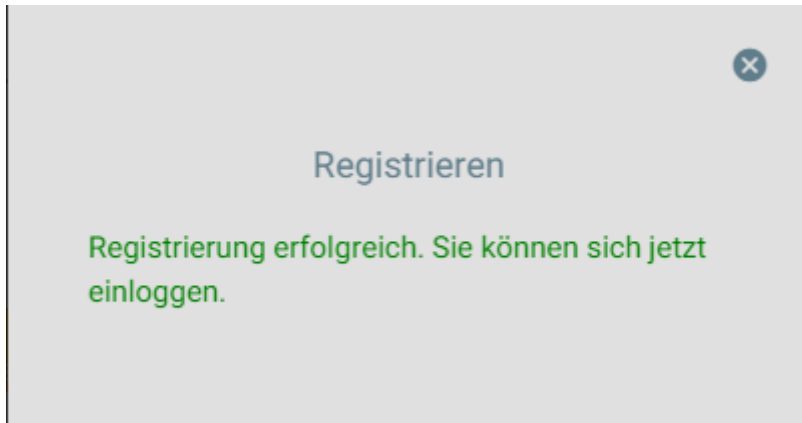
1. In der **application/config/database.php** File. Das Skript stellt die Verbindung zur Datenbank für die weiteren Skripte her. Es ist daher essentiell, dass hier die korrekten Informationen eingetragen sind.
2. Im **application/helpers/database\_helper.php** Skript. Dieses Skript erstellt die Datenbank und alle Tabellen. Auch hier müssen die Änderungen nachgetragen werden.

## 14 Bedienungsanleitung

### 14.1 Registrieren

1. Klicken Sie auf der Startsite auf **Login**. Das Login-Fenster öffnet sich.
2. Klicken Sie auf **Registrieren** unterhalb des Login-Button.
3. Füllen Sie alle Felder im Formular aus. Beachten Sie das eine E-Mail-Adresse nur einmal registriert sein kann.

4. Wählen Sie aus ob Sie ein Normal-Kunde oder ein Veranstalter-Konto eröffnen wollen, indem Sie den Slider auf die jeweilige Seite schieben.
5. Klicken Sie auf «Registrieren». Sie werden über den Status benachrichtigt.

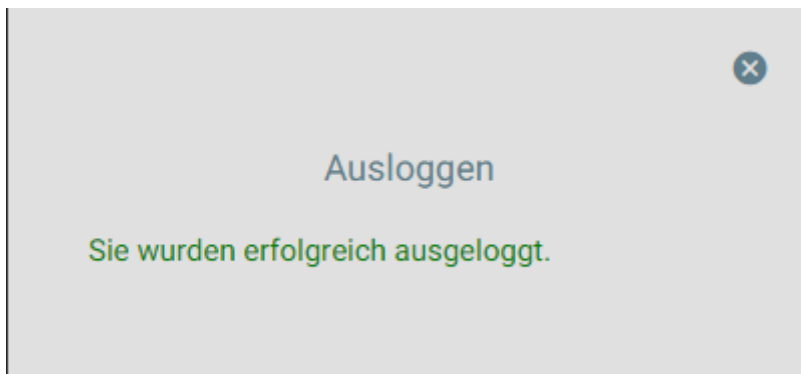


#### 14.2 Einloggen

1. Klicken Sie auf der Startseite auf **Login**. Das Login-Fenster öffnet sich.
2. Geben Sie Ihre Benutzerdaten ein.
3. Drücken Sie auf den Button «**Login**».
4. Bei einem Fehler wird Ihnen eine Fehlermeldung angezeigt.

#### 14.3 Ausloggen

1. Klicken Sie in der Navigation auf «**Ausloggen**»



#### 14.4 Profil betrachten

Diese Funktion ist nur für eingeloggte Benutzer verfügbar.

1. Klicken Sie in der Navigation auf «**Profil**».

#### 14.5 Profil bearbeiten

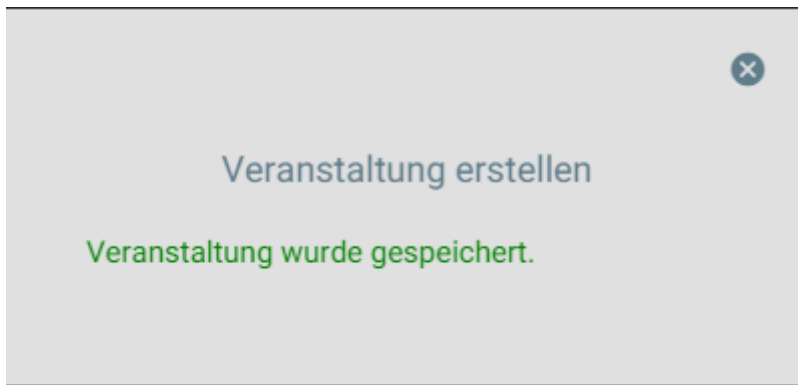
1. Klicken Sie in der Navigation auf «**Profil**».
2. Klicken Sie in der Profil-Ansicht auf den Button «**Profil bearbeiten**».
3. Füllen Sie alle Formular-felder aus.
  - Wird das Passwort-feld leer gelassen, wird das Passwort nicht geändert.

#### 14.6 Veranstaltung erfassen

Diese Funktion ist nur für eingeloggte Benutzer mit der Rolle **Veranstalter** verfügbar.

1. Klicken Sie in der Navigation auf «**Neu**».
2. Füllen Sie alle Felder im Formular aus. Diese Daten können im Nachhinein geändert werden.
3. Klicken Sie auf «**Veranstaltung eröffnen**».





#### 14.7 Veranstaltungen in Veranstalter-Sicht anzeigen

Diese Funktion ist nur für eingeloggte Benutzer mit der Rolle **Veranstalter** verfügbar.

Der Veranstalter sieht nur Veranstaltungen, welche er selbst erstellt hat.

1. Klicken Sie in der Navigation auf «**Übersicht**».

Oder

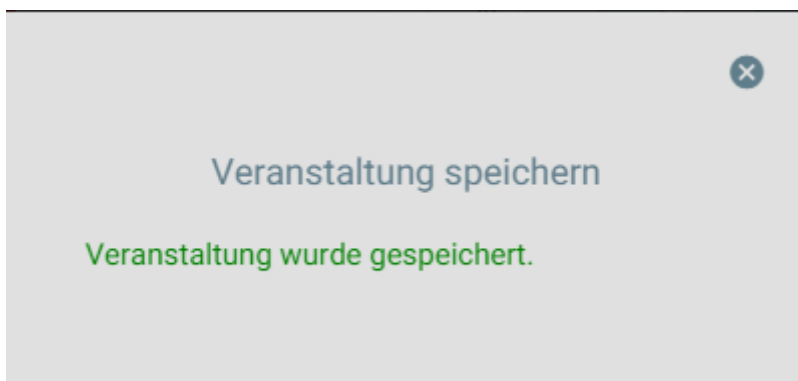
1. Klicken Sie in der Navigation auf «**Profil**».
2. Klicken Sie im Profil auf «**Zu meinen Veranstaltungen**».

#### 14.8 Veranstaltungen bearbeiten

Diese Funktion ist nur für eingeloggte Benutzer mit der Rolle **Veranstalter** verfügbar.

Sie können nur Ihre eigenen Veranstaltungen bearbeiten.

1. Klicken Sie in der Navigation auf «**Übersicht**».
2. Klicken Sie auf der gewünschten Veranstaltung auf «**Bearbeiten**».
3. Klicken Sie in der Detail-Ansicht der Veranstaltung auf «**Bearbeiten**».
4. Füllen Sie die Formular-Felder aus.
  - Die Beschreibung muss bei jeder Änderung erneut eingegeben werden.
  - Die Veranstaltungstypen können nicht entfernt werden. Veranstaltungstypen, welche in diesem Formular ausgewählt werden, werden der Veranstaltung hinzugefügt.
5. Klicken Sie auf «**Speichern**».

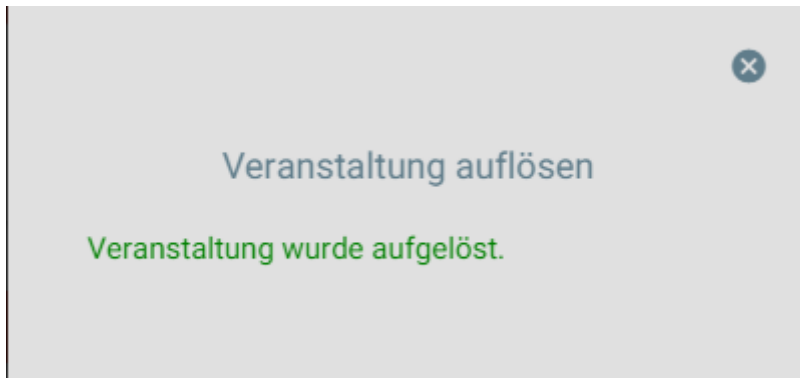


#### 14.9 Veranstaltung löschen

Diese Funktion ist nur für eingeloggte Benutzer mit der Rolle **Veranstalter** verfügbar.

Sie können nur Ihre eigenen Veranstaltungen löschen.

1. Klicken Sie in der Navigation auf «Übersicht».
2. Klicken Sie auf der gewünschten Veranstaltung auf «**Bearbeiten**».
3. Klicken Sie in der Detail-Ansicht der Veranstaltung auf «**Veranstaltung auflösen**».



#### 14.10 Veranstaltungen in Kunden-Sicht anzeigen

Diese Funktion ist für alle Benutzer (Aus- / und Eingeloggt) verfügbar.

1. Klicken Sie in der Navigation auf «**Ticketshop**».

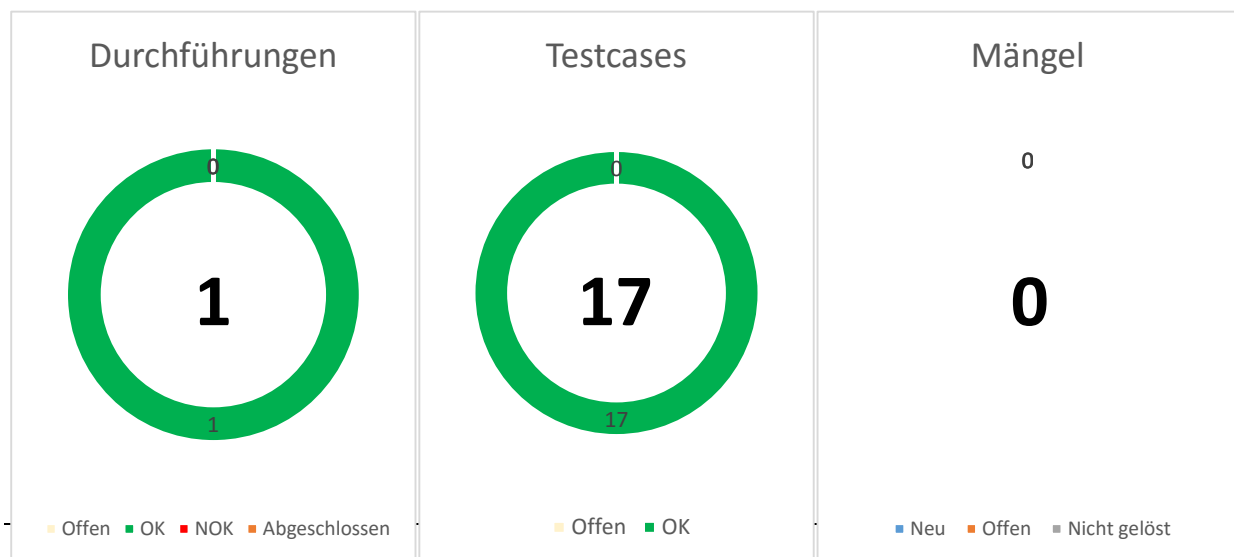
## 15 Testing

Die Testfälle und die dazugehörigen Resultate befinden sich unter «**Testprotokol\_M151.xlsx**»

### Testprotokoll

|                |   |
|----------------|---|
| Requirement    | Ticketshop  |
| Typ            | Story   |
| Applikation    | Ticketshop  |
| Testgebiet     | Web   |
| Testsystem     | <a href="ftp://192.168.13.15/PC08">ftp://192.168.13.15/PC08</a> |
| Verantwortlich | Joël  |

### Testresultate



1 OK  
0 NOK  
0 Abgeschlossen

#### 17 Testcases

0 Offen  
17 OK

#### 0 Mängel

0 Neu  
0 Offen  
0 Nicht gelöst