

Turniermanagment

Ein Webinterface zur einfachen Eingabe und Verwaltung von Fechtturnieren.

Funktionsumfang: Abspeichern der Turniere in einer Datenbank, Mailversand an zutreffende Sportler und das Abspeichern von Turnieren in waffenspezifischen Google Kalendern.

Turnier hinzufügen	Fechter hinzufügen	Turnierlisten exportieren
--------------------	--------------------	---------------------------

Name des Turniers:

Link zur Ausschreibung:

Veranstaltungsort:

Datum:

Altersklasse(n):

Schüler: <input type="checkbox"/>	Junioren: <input type="checkbox"/>
B-Jugend: <input type="checkbox"/>	Aktive: <input type="checkbox"/>
A-Jugend: <input type="checkbox"/>	Senioren: <input type="checkbox"/>

Waffe(n):

Säbel: <input type="checkbox"/>	Florett: <input type="checkbox"/>
------------------------------------	--------------------------------------

Pflichtturnier:

Anklicken für "Ja"
☐

Abschicken

Inhaltsverzeichnis

Nutzungshinweise.....	3
Anlegen von Turnieren:.....	3
Einen neuen Fechter anlegen:.....	3
Export der angelegten Turniere.....	3
Python.....	4
Emails versenden.....	4
Kalender befüllen.....	4
Installation und Updates.....	5
Voraussetzungen:.....	5
Installation.....	5
Vorbereitung:.....	5
MySQL Datenbank und Benutzer anlegen:.....	5
Webinterface installieren:.....	6
Pythonscripte:.....	6
Datenbank importieren:.....	7
Zeitsteuerung aktivieren.....	7
Aktualisieren.....	7
Datenbankaufbau.....	8
Entity-Relationship-Diagramm:.....	9
Beziehungsdiagramm:.....	9
Fazit.....	10

Nutzungshinweise

Anlegen von Turnieren:

Neue Turniere können im Webinterface im Reiter "Turnier hinzufügen" angelegt werden. Die Felder mit dem Namen, der Ausschreibung und dem Veranstaltungsort sind Pflichtfelder. Sind diese nicht ausgefüllt schlägt das Hinzufügen fehl. Das Datum lässt sich einfach mithilfe des Kalenderformulars eingeben, sobald man auf die Eingabezeile hierfür klickt. Auf diesem Weg wird auch sicher gestellt, dass das Datumsformat eingehalten wird.

Bei den Altersklassen und den Waffen muss mindestens ein Kästchen angekreuzt werden, ansonsten schlägt auch hier der Prozess fehl.

Anders sieht es beim Pflichtturnier aus. Dieses kann ankreuzt werden, es ist jedoch nicht zwingend erforderlich. Sofern es ausgewählt ist wird sich im generierten Mailtext ein zusätzlicher Hinweis auf die Relevanz dieses Turniers befinden.

Einen neuen Fechter anlegen:

Das Anlegen eines neuen Fechter ist unter "Fechter hinzufügen" möglich. Hier muss in jedem Feld eine Angabe gemacht werden und auch hier wieder mindestens ein Kästchen angekreuzt werden. Auch dieses Formular gibt Rückmeldung ob das Hinzufügen erfolgreich gewesen ist oder nicht.

Sofern Eltern informiert werden sollen sind diese mit dem Jahrgang des Kindes einzugeben, um eine Benachrichtigung nach Altersklassen zu ermöglichen.

Export der angelegten Turniere

Es ist möglich die erfassten Turniere nach Waffe, Altersklasse und Datum sortiert in ein Tabellendokument zu exportieren. Hierfür ist ein Klick auf den Button im Reiter "Turnierlisten exportieren" nötig und ein anschließender Klick auf den „Turnierliste herunterladen“ Button. Nach kurzer Zeit sollte sich ein Download öffnen, welche eine CSV Tabelle herunterlädt. Diese lässt sich mit herkömmlichen Tabellenkalkulationsprogrammen öffnen und dort grafisch aufarbeiten und ausdrucken.

Python

Emails versenden:

Der in Python verfasste "Turnierkurier" ist ein Script, welches die kommenden Turniere der nächsten 6 Wochen aus der Datenbank abfragt, die Altersklassen und Waffen für dieses Turnier ermittelt und anschließend mit den gespeicherten Daten der Fechter abgleicht um diese bei Übereinstimmung per Mail zu informieren.

Sollte dieses Script zu Demonstrationszwecken gestartet werden empfiehlt es sich den Testmodus durch setzen der Variable "test = 1" am Anfang des Scriptes zu aktivieren. Die Folge ist, dass es die Mail nicht zu versenden versucht und stattdessen in einer Konsole den Mailtext ausgibt.

Kalender befüllen:

Das Script "calendar-push" schreibt die Turniere nach Waffen sortiert in eigene Google Kalender, welche auf nahezu allen Plattformen eingebunden werden können. (Outlook, Thunderbird, iOS, Android). Dies ist auch der Grund, weshalb die Entscheidung auf ein proprietäres Kalenderformat gefallen ist: Unter Android ist es nicht nicht ohne erweiterten Aufwand möglich einen anderen Kalender importieren. Dies ist mit vorinstallierten Apps nur mit Google Calendar und den Google-internen Freigaben möglich. Da die Nutzer möglichst geringe technische Hürden nehmen sollen fiel die Entscheidung auf Google Calendar. Hinzu kommt, dass eine Integration in andere Software bereits umfassend dokumentiert ist.

Installation und Updates

Voraussetzungen:

- Ein Webserver wie Apache oder Nginx (getestet)
- PHP Unterstützung (Version 5.5 oder höher)
- MySQL und die notwendigen PHP-MySQL Erweiterungen
- Python Version 2.7.* (3.4 ist inkompatibel aufgrund der Google O-Auth Bibliotheken)
- Python MySQL-Connector (→ Installation)
- Python Google Oauth Pakete (→ Installation)
- Ein lokaler Mailserver

Installation

Vorbereitung:

1. Git installieren (Nicht notwendig, vereinfacht aber Installation und Updates)
2. Den Webserver mit den zugehörigen (PHP-)Paketen nach der offiziellen Dokumentation installieren. (Ausschließlich für lokale Testsysteme kann auch XAMPP genutzt werden.)
3. MySQL Server installieren und einen „root“-Benutzer inklusive Passwort anlegen.
4. Python 2.7 inklusive „pip“ installieren. (
5. Die für den Kalender notwendigen Python Bibliotheken installieren:
pip install --upgrade google-api-python-client python-gflags
6. Den Python MySQL Connector von <https://dev.mysql.com/downloads/connector/python/> für das entsprechende Betriebssystem herunterladen und installieren

MySQL Datenbank und Benutzer anlegen:

1. Öffnen einer Kommandozeile und MySQL starten (mysql -u root -p und bei der Aufforderung das Passwort eingeben)
2. Einen neuen Datenbankbenutzer anlegen und ein Passwort zuweisen:
CREATE USER 'turniere'@'localhost' IDENTIFIED BY '\$password';

3. Erstelle die Datenbank:

```
CREATE DATABASE IF NOT EXISTS `turniermanagment` DEFAULT CHARACTER SET `utf8`  
COLLATE `utf8_unicode_ci`;
```

4. Weise dem neu erstellten Nutzer die Berechtigungen für die Datenbank zu:

```
GRANT ALL PRIVILEGES ON `turniermanagment`. * TO 'turniere'@'localhost';
```

5. Aus MySQL ausloggen:

```
\q
```

Webinterface installieren:

1. Das öffentliche Verzeichnis des Webserver aufrufen und eine Kommandozeile öffnen.

2. "git clone https://git.scimeda.de/thore/turniermanagment.git" ausführen und die aktuelle Version herunterladen. (Bei einer Installation ohne git kann von der Website auch ein Archiv heruntergeladen und dann lokal extrahiert werden.

3. Die Datenbankeinstellungen im Ordner subpages von Tournament.php und AddFencer.php sowie in libs/ExportCSV.php Nutzernamen und Passwort durch die zuvor angelegten Nutzerdaten ersetzen.

4. Die Änderungen in git sichern für spätere Updates:
git commit -am „Datenbankdaten der Website angepasst“

Pythonscripte:

Die Scripte befinden sich im Unterordner Python

1. Die Datenbankparameter in calendar-push.py und turnierkurier.py ebenfalls anpassen.

2. In calendar-push.py müssen zudem die Kalendereinstellungen angepasst werden.

Zunächst werden Google API Zugangsdaten aus der Developers Console benötigt:

<https://developers.google.com/google-apps/calendar/firstapp#register>

Die dort erstellten Zugangsdaten für die Calendar API müssen bei

```
client_id = "your client ID here"
```

```
client_secret = "your client secret here"
```

eingefügt werden.

3. Außerdem sind zwei öffentliche Google Kalender notwendig. Deren IDs sind müssen in jeweils eines der:

Calendar*ID = "your calendar ID here"

Felder eingefügt werden. Die KalenderID sit in der jeweiligen Kalendereinstellung unter "Kalenderadresse" aufzufinden.

4. Auch diese Änderungen wieder in git sichern für spätere Updates:
git commit -am "Variablen der Pythonscripte gesetzt"

Datenbank importieren:

1. Im Ordner "Datenbanken" befinden sich zwei Dateien:

- db-sample.sql
- db.sql

Erstgenannte enthält einige Beispiele für Fechter und Turniere.

2. Aufruf des Ordners Datenbanken mit einer Konsole

3. Import der Datenbank mit

mysql -u turniere -p turniermanagment < db.sql

oder für die Beispieldatenbank

mysql -u turniere -p turniermanagment < db-sample.sql

Nach einer Eingabe des Passworts importiert MySQL die die nötigen Tabellen und gegebenenfalls mit Daten

Zeitsteuerung aktivieren

1. Für das jeweilige Betriebssystem unterschiedliche zeitgesteuerte Aufrufe der Pythonscripte aktivieren (Empfohlen: wöchentlich)

Aktualisieren

Der bevorzugte Weg ein Update einzuspielen ist ebenfalls über git. Die Alternative ist ein erneutes Herunterladen der aktuellen Version und ein erneutes ausführen der Installationsanweisungen. Eine Datenbankeinrichtung entfällt in diesem Falle jedoch.

1. Neue Version herunterladen und zwischenspeichern:

git checkout -b updates

git fetch origin updates

2. Updates in die aktuelle Konfiguration eingliedern (\$Version durch die zu installierende Version ersetzen)

git checkout master

git merge update

Datenbankaufbau

Die Datenbank besteht aus vier Tabellen, welche Daten beinhalten und drei weiteren Zwischentabellen, welche Verknüpfungen abspeichern.

Kerntabellen:

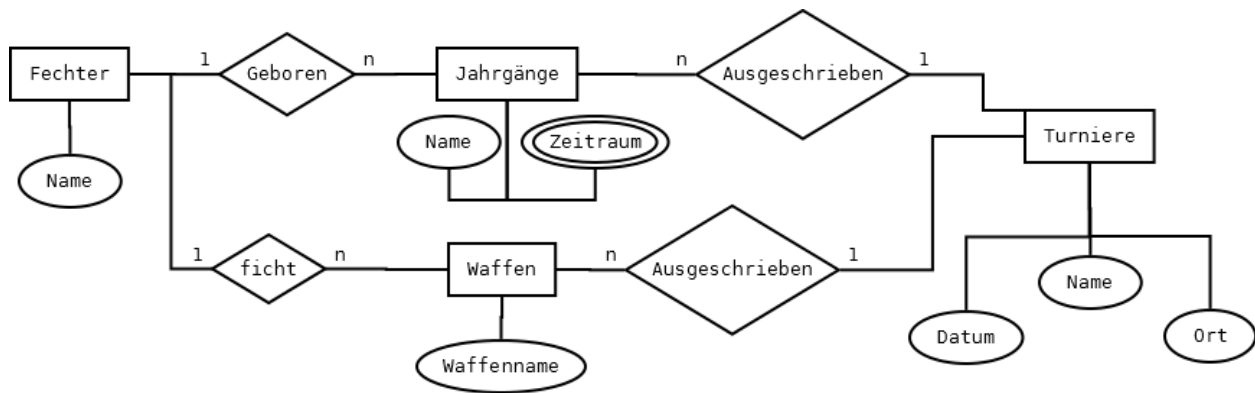
- turnier
 - Allgemeine Daten des Turniers (Ort, Name, Datum, Ausschreibung) und ob es bereits per Mail bearbeitet wurde.
- fechter
 - Daten die über den Fechter erforderlich sind (Name, Email) sowie das Geburtsjahr. Die Tabelle ist nicht mit "jahrgaenge" verknüpft
- waffe
 - Name der Waffe und eine zugehörige ID
- jahrgaenge
 - Die einzelnen Jahrgänge mit Beginn und Ende definiert nach Jahren. Die Daten verschieben sich jedes Jahr.

Zwischentabellen:

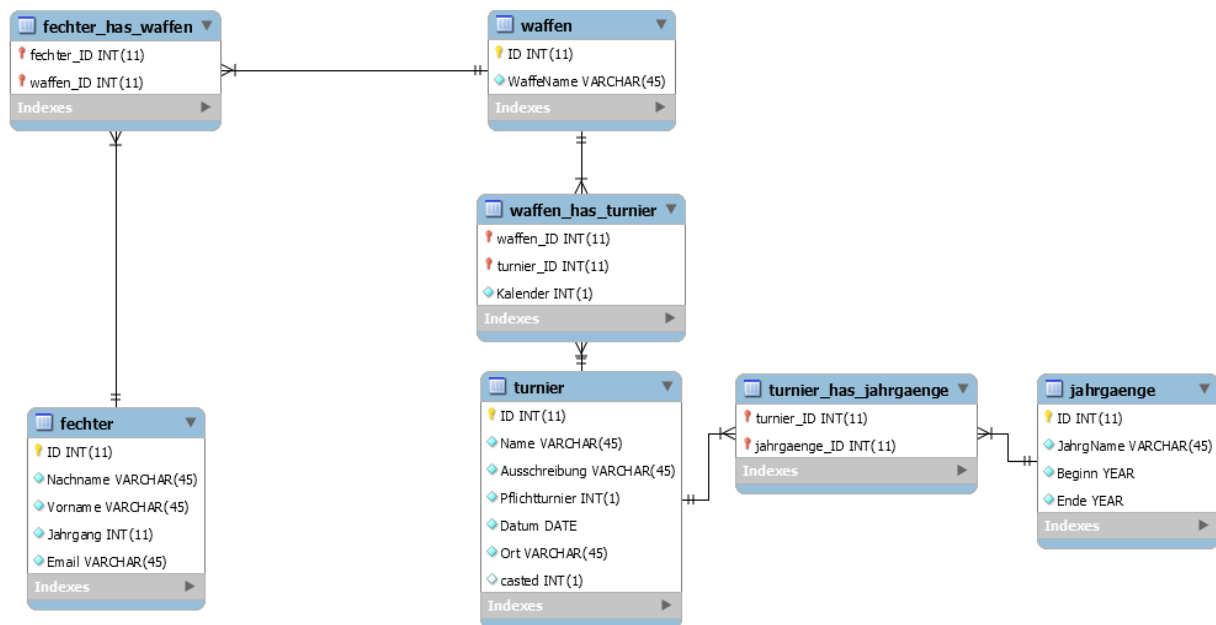
- fechterwaffe
 - Die FechterID und die zugehörige(n) WaffeID(s)
- waffetur
 - Die TurnierID und die IDs der zugewiesenen Waffen. Darüber hinaus wird in dieser Tabelle abgespeichert, ob es bereits einen Schreibvorgang in den Kalender gegeben hat.
- altersklassen
 - Die Verknüpfung zwischen den Turnieren und den möglichen Jahrgängen.

Zwischen der Tabelle Fechter und Jahrgänge gibt es keine Verknüpfung, da die Jahrgänge sich jährlich verschieben. Deshalb wird bei den Fechtern ein Geburtsjahr gespeichert und anschließend ein Abgleich ausgeführt, ob dieses innerhalb des definierten Bereiches des Jahrganges liegt.

Entity-Relationship-Diagramm:



Beziehungsdiagramm:



Fazit

Das Projekt begann als einfache Website, welche nur Turniere speichern sollte.

Daraus ist jedoch relativ schnell weitaus mehr entstanden. Auf der einen Seite, dass sich das Design von Standard HTML Formularen hin zu jQuery-UI veränderte, was die gesamte Entwicklung des Webinterfaces vereinfachte. Darüber hinaus kam mit dem Datepicker eine einfache Möglichkeit zu erzwingen, dass ein Nutzer nicht gegen das von MySQL erwartete Datumsformat verstößt.

Zum Anderen wuchs das Projekt auf Nachfrage dann von der einfachen Eingabe um den Mailversand. Dies verursachte die Notwendigkeit in einer Sprache zu arbeiten, die sowohl auf Servern als auch auf einem lokalen Entwicklungssystem ohne große Einrichtung ausgeführt werden kann. Da Python eine einfach zu lesen und zu schreibende Scriptsprache ist, welche unter Linuxsystemen mit geringem Aufwand installiert und aktuell gehalten werden kann, fiel die Entscheidung hierauf.

Eine Problematik stellte die Verbindung zwischen Python und MySQL her, da der offizielle Client nur mäßig dokumentiert ist, besonders die Ausgabeformate. Dies ließ sich aber durch die mehr oder weniger saubere Lösung in einer separaten Funktion lösen.

Die Idee die vorhandenen Daten in einen Kalender zu schreiben bestand schon zu Anfang des Projektes, kam aber erst im späteren Verlauf hinzu. Die Möglichkeit Daten zentral zu sammeln und von dort aus auf verschiedene Informationskanäle zu verteilen ist überaus praktisch, besonders da die Verteilung aktuell zum Teil über verschiedene Personen läuft, was zu Verzögerungen führt. Besonders bei der Übergabe in Kalender. Dies lässt sich mit diesem Projekt bequem lösen.

Erweiterungsmöglichkeiten wären zum Beispiel weitere Formulare, welche die Überarbeitung der Turniere und Fechter ermöglichen. Ansätze hierfür sind jedoch an nicht nachvollziehbaren Problemen mit jQuery gescheitert. Eine andere Idee wäre es noch Zu- und Absagen zu erfassen und auch diese wieder in eine Tabelle zu exportieren.

Abschließend lässt sich sagen, dass dieses Projekt einen hohen Lernfaktor hatte, da mehrere verschiedene Sprachen wie HTML, PHP, JavaScript bzw. jQuery, Python und SQL verwendet wurden, was eine vielseitige Einarbeitung erforderte. Besonders die Tatsache, dass das Produkt so einfach wie möglich sein musste und Fehlerquellen, wie beispielsweise die Datumseingabe, so weit wie möglich umgangen werden mussten stellte eine besondere Herausforderung dar.