|  |
| --- |
| TH Köln Campus Gummersbach |
| Social Events |
| Praxisprojekt Dokumentation |

|  |
| --- |
| Kevin Röttel (Matr. Nr. 11139133) |

Inhaltsverzeichnis

[1. Projektidee 2](#_Toc129159268)

[2. Projektziel 2](#_Toc129159269)

[3. Genutzte Technologien 2](#_Toc129159270)

[4. Domain Model 3](#_Toc129159271)

[5. Funktionalitäten 4](#_Toc129159272)

[5.1 Login 4](#_Toc129159273)

[5.2 Account Einstellungen 4](#_Toc129159274)

[5.3 Event Erstellen 4](#_Toc129159275)

[5.3.1 Venue Erstellen 4](#_Toc129159276)

[5.3.2 Künstler Erstellen 4](#_Toc129159277)

[5.4 Event Ansicht 5](#_Toc129159278)

[5.4.1 Entfernungsrechnung 5](#_Toc129159279)

[5.5 Event Bearbeiten 5](#_Toc129159280)

[5.6 Locations Page 5](#_Toc129159281)

[5.7 Artist Page 5](#_Toc129159282)

[5.8 Freunde System 5](#_Toc129159283)

[5.9 Search Bar 6](#_Toc129159284)

[5.10 Dashboard 6](#_Toc129159285)

[6. Fazit 7](#_Toc129159286)

[7. Ausblick auf Bachelorarbeit 7](#_Toc129159287)

# Projektidee

Durch den Abfall von insbesondere jüngeren Nutzern bei Facebook, werden dort ebenfalls Einträge für diverse Veranstaltungen immer weniger. Mit der Entwicklung einer webbasierten Anwendung, soll eine Alternative geschaffen werden, um sich mit anderen Besuchern besser zu vernetzen und um ggf. Fahrgemeinschaften oder Treffen zu organisieren. Die Anwendung soll ein allgemeines Freunde-System beherbergen, sowie das Erstellen von Veranstaltungen möglich machen. Nutzer sollen ihr Interesse bekunden können, zu Veranstaltungen zusagen können sowie mit anderen Nutzern über dedizierte Bereiche kommunizieren können. Ebenfalls soll ein geeigneter Vorschlagsalgorithmus zum Einsatz kommen, um neue Veranstaltungen an die Nutzer heranzutragen. Ebenfalls soll nach einer Möglichkeit gesucht werden, neue Veranstaltungen automatisch von verschiedenen Quellen abzufragen und in der Anwendung einzuspeisen.

# Projektziel

Das Ziel ist eine Plattform über welche Nutzer sich über Veranstaltungen informieren, sowie mit anderen Nutzern über diese Veranstaltungen kommunizieren können. Die Anwendung soll gegen Ende des Projektes ein Minimal Viable Product sein mit Explorativen Features wie dem Vorschlagealgorithmus.

# Genutzte Technologien

Als Grundgerüst und Backend für die Webplattform wird Laravel genutzt. Laravel ist ein PHP-Framework welches das MVC-Muster[[1]](#footnote-1) nutzt. Es ist in dieser Applikation für die Datenstruktur, sowie für alle Datenbank-Manipulationen verantwortlich. Ebenfalls stellt Laravel den Einstiegspunkt für die Frontend-Frameworks zur Verfügung, sowie die Login- und Register-Seiten. Auch werden über Laravel alle internen API Routen definiert.

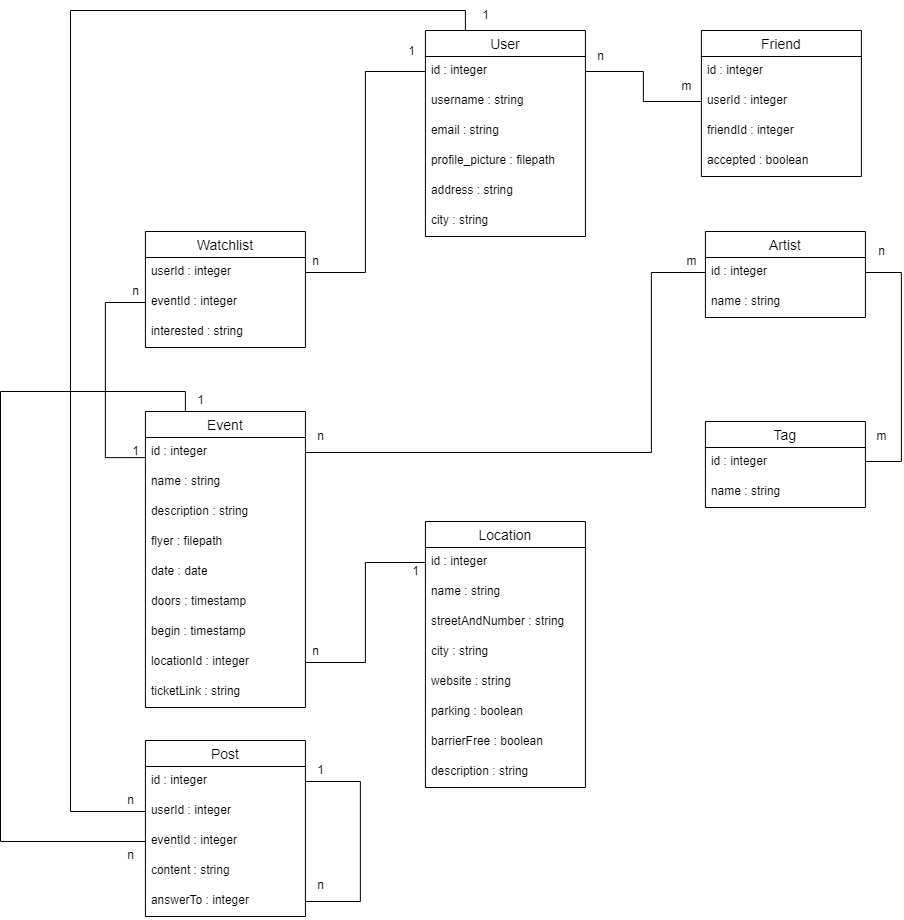
Für die Frontend Interaktionen ist Vue verantwortlich. Vue ist ein JavaScript Framework für Single-Page-Webanwendungen[[2]](#footnote-2). Eine Vue-Applikation besteht aus verschiedenen Components, welche mehrfach getrennt voneinander wiederverwendet werden können. Jeder Component besteht aus einem Template- und einem Script-Teil.

Um auf der Single-Page-Webanwendung auch Browserfunktionalitäten zur Verfügung zu stellen, wurde Vue-Router genutzt. Vue-Router ist ein Add-on um verschiedene Components je nach URL zu laden. Somit kann auch die Zurück-Funktion des Browsers genutzt werden, und beim neuladen der Seite wird nicht automatisch auf die Startseite zurückgeleitet.

Da mehrere Vue-Components die gleichen Daten nutzen müssen wird Pinia verwendet. Pinia ist ebenfalls ein Add-on, um seitenübergreifend Daten in einem virtuellen Storage zu speichern. Hierdurch müssen nicht bei jedem Seitenwechsel alle Daten neugeladen werden. Somit werden nur bei einem Page-Refresh alle Daten aus der Datenbank abgefragt.

Damit alles etwas angenehmer aussieht, wird Bootstrap verwendet um mit Hilfe von CSS-Klassen Elemente besser hervorzuheben. Zudem können durch Bootstrap Elemente wie z.B. Modale wesentlich schneller erstellt werden.

# Domain Model



# Funktionalitäten

## Login

Die Login-Seite ist die allgemeine Startseite für nicht angemeldete Benutzer. Hierrüber können sich neue Nutzer registrieren sowie bestehende einloggen.

Bei fehlerhaften Login-Versuchen wird dem Nutzer mitgeteilt das seine Anmeldedaten nicht korrekt sind. Ebenfalls werden bei der Registrierung Informationen angezeigt, sollte eine angegebene E-Mail oder ein Username bereits im System vorhanden sein.

## Account Einstellungen

Durch die Account Seite lassen sich die Informationen des Nutzers bearbeiten. Dazu zählen das Ändern des Usernames, der E-Mail sowie das Passwort und das Hochladen eines Profilbildes.

Außerdem können Nutzer ihren Wohnort mit Adresse festlegen. Diese Angaben sind freiwillig, und dienen rein für die Berechnung von Entfernungen zu Veranstaltungsorten.

## Event Erstellen

Über die Navbar[[3]](#footnote-3) lässt sich die Seite für die Eventerstellung öffnen. Diese besteht aus dem Formular und einer Vorschau der Eventansicht.

Das Formular besteht aus neun Eingabefeldern, welche die wichtigsten Informationen für eine Veranstaltung abdecken sollen. Dazu zählt der Name der Veranstaltung, eine Beschreibung, ggf. ein Flyer sowie das Datum mit Einlass- und Beginn-Zeitpunkten. Ebenfalls lässt sich eine Verlinkung hinterlegen für Tickets zur Veranstaltung, sowie der Veranstaltungsort und die auftretenden Künstler.

### Venue Erstellen

Innerhalb des Formulars lassen sich zudem neue Veranstaltungsorte anlegen. Dies war zuerst ein losgelöster Teil der Eventerstellung, wurde dann aber so integriert das eine Eventerstellung nicht unterbrochen werden muss, um einen neuen Veranstaltungsort hinzuzufügen.

Ein Veranstaltungsort (ff. Venue), besteht aus dem Namen, einer Adresse, dem Ort und einer optionalen Webseite. Zudem wird abgefragt ob es ausreichend Parkplätze vor Ort gibt, sowie der Zustand der Barrierefreiheit innerhalb der Venue. Diese können dann auch über eine Beschreibung weiter erläutert werden.

### Künstler Erstellen

Für eine Veranstaltung müssen auch Künstler ausgewählt werden. Dies funktioniert über ein Text-Eingabefeld, welches mehrere Werte gleichzeitig tragen kann. Die ausgewählten Künstler können dann auch per Klick wieder entfernt werden. Gleichzeitig funktioniert dieses Eingabefeld auch als Suche, um vorhandene Künstler zu finden, und direkt auszuwählen.

Sollte ein Künstler noch nicht in der Datenbank vorhanden sein, so muss der Nutzer den vollständigen Namen eingeben und per Eingabe hinzufügen. Hier wird automatisch eine Benachrichtigung angezeigt, dass die Korrektheit des Künstlers überprüft werden soll. Zudem wird auch eine weitere Information angezeigt, sollte der Name sehr ähnlich zu einem anderen Künstler sein.

## Event Ansicht

Durch das Anklicken eines kleinen Event-Teasers auf dem Dashboard, oder nach der Erstellung, wird das Event als eigenständige Seite geöffnet. Dort sind alle hinterlegten Informationen abzurufen, inklusive der ungefähren Entfernung vom Nutzer zur Venue, sowie der Anzahl an Interessens-bekundungen sowie Zusagen.

Ebenfalls sind Verlinkungen der Venue und den einzelnen Künstlern vorhanden, welche zu den jeweiligen Seiten führen.

Es gibt auch die Möglichkeit Beiträge zu einem Event zu posten. Dies ist am unteren Ende der Eventinformationen möglich. Dies kann genutzt werden, um eventuelle Fahrgemeinschaften zu bilden, oder weitere Informationen zum Event hinzuzufügen (z.B. Zeitpläne), ohne das Event selbst Bearbeiten zu müssen. Auch können zu jedem Beitrag Kommentare verfasst werden.

### Entfernungsrechnung

Für den Fall, dass der Nutzer seine Adresse im System hinterlegt hat, wird innerhalb des Events die ungefähr kürzeste Route zur Venue berechnet. Dies setzt ebenfalls voraus, dass die Venue namentlich auf den genutzten Karten hinterlegt ist, oder ebenfalls eine Adresse hinterlegt hat.

Zunächst müssen für die korrekte Berechnung die Längen- und Breitengrade des Nutzers und der Venue herausgefunden werden. Dies geschieht über die Nominatim-API, was grundsätzlich eine Search Engine für OpenStreetMap[[4]](#footnote-4) ist.

Mit diesen Daten wird dann über die API von OSRM[[5]](#footnote-5) eine Route abgefragt, wovon nur die Distanz benötigt wird, um sie schließlich dem Nutzer anzuzeigen.

## Event Bearbeiten

Die Eventbearbeitung ist fast identisch mit dem Aufbau der Erstellung. Nur das Ändern des Line-Ups ist in ein Modal ausgelagert worden, um Fehlklicke zu vermeiden.

## Locations Page

Jede Location kann auch als eigene Seite angezeigt werden, um weitere Infos zu erhalten. Dort können ebenfalls Details bearbeitet werden, um z.B. auf neue Parkmöglichkeiten hinzuweisen.

Ebenfalls werden hier alle bevorstehenden und vergangene Veranstaltungen in einer Liste angezeigt.

## Artist Page

Auch jeder Künstler wird auf einer eigenen Seite angezeigt. Hier sind ebenfalls alle bevorstehenden und vergangenen Veranstaltungen hinterlegt.

Zudem besitzen Künstler ein Tag-System. Hier können beliebig beschreibende Tags hinzugefügt und entfernt werden. Diese Tags dienen den Vorschlagssystemen auf dem Dashboard, und um Nutzern ein Gefühl zu geben, was sie bei dem Künstler erwartet.

## Freunde System

Ein rudimentäres System für Nutzerfreundschaften ist ebenfalls vorhanden. Dieses beschränkt sich zurzeit auf das Senden, Annehmen und Löschen von Freundschaftsanfragen.

Ebenfalls ist auf dem Dashboard ein Abschnitt vorhanden für Events an denen Freunde interessiert sind oder zugesagt haben.

## Search Bar

Über die Navbar lassen sich Suchanfragen starten. Diese Suchanfragen vergleichen den String mit dem Namen von Künstlern, Events und Venues. Übereinstimmungen werden dann als Liste auf einer Ergebnis-Seite angezeigt, und verlinken auf die jeweils eigenständigen Seiten der Künstler, Events und Venues.

## Dashboard

Das Dashboard ist die allgemeine Startseite für eingeloggte Nutzer. Hier sind die Events der Nutzer-Watchlist zu finden, sowie Vorschläge, die darauf basieren.

Die Vorschläge werden zurzeit mit Hilfe der Cosine Similarityoder des Jaccard Index generiert. Dafür werden zu jedem Event, alle Tags der Künstler in ein gemeinsames Array gesteckt. Dieses Array wird dann von den Algorithmen genutzt um Vorschläge zu generieren.

Die Kosinus-Ähnlichkeit berechnet die Ähnlichkeit zweier Vektoren. Dies geschieht durch die Berechnung des Kosinus-Winkels zwischen den beiden Vektoren. Damit können wir auch berechnen wie ähnlich sich zwei Dokumente oder auch Texte sind anhand von Häufigkeits-Vektoren.

Dies könnte folgendermaßen in Code aussehen:  
Die Funktion *cosineSimilarity* ist in zwei weitere Funktionen gegliedert. Einmal das Skalarprodukt (*dotProduct*) als Zähler und zweimal die Größe (*magnitude*) der Vektoren als Nenner.

const dotProduct = (a, b) => {

    let dotProduct = 0;

    for (let i = 0; i < a.length; i++) {

        dotProduct += a[i] \* b[i];

    }

    return dotProduct;

}

const magnitude = (vector) => {

    let sum = 0;

    for (let i = 0; i < vector.length; i++) {

        sum += vector[i] \* vector[i];

    }

    return Math.sqrt(sum);

}

const cosineSimilarity = (a, b) => {

    return dotProduct(a, b) / (magnitude(a) \* magnitude(b));

}

Der Jaccard Index beschreibt die Ähnlichkeit von zwei Gruppen. Definiert wird dies durch die Größe der Schnittmenge geteilt durch die Größe der Vereinigung der jeweiligen Gruppen. Je höher der Index ist, und je näher er an 1 liegt, desto höher ist die Ähnlichkeit der Mengen.

Würde man dies nun in JavaScript umsetzen, könnte dies z.B. so aussehen:  
Zunächst werden die Überschneidungen gefiltert, und danach die Berechnung aus der obigen Formel angewendet.

const jaccardIndex = (s1, s2) => {  
 let intersection = new Set([...s1].filter(i => s2.has(i)));  
 return intersection.size / (s1.size + s2.size - intersection.size);  
}

# Fazit

Das Projekt ist nun zum Ende ein Minimal Viable Product mit den wichtigsten Funktionalitäten realisiert. Dies umfasst die Eventerstellung und -bearbeitung, inklusive der Erstellung von Venues und Künstlern. Ebenfalls wurden explorative Features implementiert, wie Vorschlagsalgorithmen.

Die Plattform ist somit in einem test- und erweiterbaren Zustand. Erweiterungen würde unter anderem folgende Punkte enthalten:

* Mobile Optimierung und interessanteres Design
* Web Accessibility/Behindertengerechtes Internet
* Freunde-System präsenter machen (z.B. Privat-Chat, Posts von Freunden hervorheben)
* Benachrichtigungen bei neuen Event-Zusagen von Freunden oder Posts zu Interessierten Events
* Dateneffizienteres Speichern und Laden von Flyern und Profilbildern
* Events absagen
* Event-Grabbing von z.B. Tour Flyern oder Social-Media Posts
* Bewertung von Künstler-Tags

# Ausblick auf Bachelorarbeit

Für die Bachelorarbeit soll der Fokus auf die Vorschlagssysteme gelegt werden. Bislang wurden auf der Plattform nur Content-basierende Algorithmen genutzt. Eine Möglichkeit wäre z.B. ein Nutzer-basierendes oder Hybrides Vorschlagssystem zu implementieren. Diese Systeme könnten miteinander verglichen werden, wie komplex die Implementierung ist und wie gut jeweils Vorschläge generiert werden. Dies könnte auch mit freiwilligen Nutzern gemacht werden, um somit die Eingabe von vielen Sample-Daten zu vermeiden.

1. Model View Controller, Entwurfsmuster [↑](#footnote-ref-1)
2. Webanwendung bestehend aus einzelner Root-Datei deren Inhalt dynamisch nachgeladen wird [↑](#footnote-ref-2)
3. Navigation Bar, Navigationsleiste, meist am oberen Teil des Programmes zu finden [↑](#footnote-ref-3)
4. Open Source Alternative zu Google Maps [↑](#footnote-ref-4)
5. Open Source Routing Machine [↑](#footnote-ref-5)