

Implementation

TravelCompats

Technology
Arts Sciences
TH Köln

Installationsanleitung TravelCompats

Server

1. **NodeJS** downloaden und installieren.
 - <https://nodejs.org/en/download/>
2. **MongoDB** downloaden
 - <https://www.mongodb.com/download-center>
 - a. vom Terminal aus zum Ordner navigieren
 - b. "cd mongodb/bin" eingeben
 - c. mit "./mongod" starten
3. Dann den Server starten
 - a. vom Terminal in das Verzeichnis MS3/Prototyp/Server navigieren
 - b. node app.js eingeben und bestätigen
 - c. Sollten die Module nicht installiert sein, im Terminal "npm install" eingeben
4. Damit der Client betriebsfertig ist und die Testdaten in die Datenbank eingepflegt werden, im Browser folgende Ressourcen ansprechen (ein "Okay" gibt es nicht zurück, beim gestarteten Server bekommt man eine Rückmeldung darüber, dass die Daten eingefügt wurden)
 - a. "<http://localhost:3000/testDaten>"
 - b. "<http://localhost:3000/testRatings>"
 - c. "http://localhost:3000/atmosphere"

Client

1. **Android Studio installieren**
- <https://developer.android.com/studio/index.html>
2. **Projekt Dateien (MS3/Prototyp/Client) in den Projektordner von Android Studio kopieren**
3. **IP-Adresse herausfinden (im Terminal "ifconfig -a" und die eigene IP-Adresse raus suchen (oft bei eth0 oder en0))**
4. **Bei folgenden Dateien die Variable "server_url" anpassen (die herausgefundene IP-Adresse hinterlegen und den Port dran hängen. Bsp.: "private String server_url = "<http://192.168.0.248:3000>";"): Recommend.java, Register.java, Sign.java, Upload.java, Profile.java**
5. **Der Client sollte nun ausgeführt werden können (es könnte in Android Studio oben rechts ein Fenster auftauchen, wo steht "Sync Now", das bitte, wenn gefordert, auch noch machen)**
6. **Angemeldet werden kann man sich mit dem Benutzer "test" und dem Passwort "test". Ein eigener "Account" darf ebenfalls gerne angelegt und benutzt werden.**

Abweichungen vom Konzept

Von Open Street Maps zu Google Places

Ressourcen die zur Ermittlung der Atmosphäre von Orten gesammelt und verarbeitet werden müssen, sollen laut Konzept aus der Bibliothek von Open Street Maps stammen. Zu Beginn war geplant alle Informationen von OSM über REST Methoden abzufragen. Nach einigen Evaluationsschritten während der praktischen Programmierarbeit wurde allerdings leider ersichtlich, dass dies scheinbar nur über schwierige Umwege möglich ist. So konnte man in den Testphasen mit dem OSM-Webservice keine Informationen als JSON-Objekt beziehen. Dies könnte nur umgangen werden, wenn ein OSM Dienst auf dem Client im Hintergrund ausgeführt wird, Informationen an den Server schickt, wo dieser verarbeitet werden und im Anschluss zurück an den Client fließen. Diese Alternative ist allerdings sehr unperformant und wird aus dem Entwicklungsprozess ausgeschlossen.

Als Konsequenz werden nun Daten von der Google Places API bezogen. Diese bietet zwar nicht alle benötigten Informationen und manche sind erst nach Zahlung mit dem Premium Modell erhältlich, dennoch eignet sich dieses Verfahren um mit der Implementation fortzufahren.

Aktuelle Attribute der Atmosphäre

Durch die Profilangaben eines Nutzers kann eine gewünschte Atmosphäre ermittelt werden. Ein Ort verfügt durch seine Attribute bereits über jene. So prüft die Anwendungslogik also ob sich die jeweiligen Atmosphären abgleichen. Passt eine Ort-Atmosphäre zu einer gewünschten Nutzer-Atmosphäre wird dieser Ort mit hoher Wahrscheinlichkeit empfohlen.

Aktuelle Attribute sind:

- **Geruch** eines Ortes.
- **Geschmack** des Essens z.B.
- **Freundlichkeit** der Menschen und des Personals

- **Geeignete Aktivitäten.** Stammt aus Analyse von App Daten
- **Geräusche** (Umgebungs lautstärke) stammen aus Netatmos und eigenen Aufnahmen, um zu prüfen wann Richtwerte überstiegen werden.
- **Gesprochene Sprachen** an einem Ort und eines Landes werden berücksichtigt.
- **Wetter.** Informationen stammen von Open Weather Map.
- **Familientauglichkeit**
- **Sauberkeit** von Straßen und Gebäuden.
- **Behindertenfreundlichkeit/ Barrierefreiheit** (Differenzierbar in verschiedenen Bereichen wie Farbenblindheit und Rollstuhltauglichkeit)
- **Sehenswürdigkeiten.**
- **Textuelle Bewertungen** von anderen Nutzern über gelistete Orte in Google Places werden bei Empfehlungen mit angezeigt. Ein Keyword-Filter wird hierbei eingesetzt, um aus einem Fließtext wertvolle Informationen zu sammeln.

Diese Attribute werden in dezimalen Werten beschrieben - von 0 bis 5. 0 ist dabei ein Platzhalter, falls dieses Attribut gar nicht vertreten ist. 1 steht für den stärksten positiven Einfluss (Schulnoten-System).

```
LocationAtmosphere:{
  Geruch: [0-5],
  Geschmack: [0-5],
  Freundlichkeit: [0-5],
  Aktivitaeten: [(App-Analyse:) Pokemon Go, Fitness, etc.],
  Umgebungslautstärke: [0-5],
  Sprachen: [Deutsch, Englisch, Spanisch, Italienisch, Türkisch, ...]
  Wetter: ["Gebäude","Wetter betroffen"],
  Familientauglichkeit: [0-5],
  Sauberkeit: [0-5],
  Barrierefrei: [(Unterstützende Mittel)],
  Google Bewertung: [0-5],
}
```

Pseudocode Atmosphäre

Daten aus installierten Nutzer-Apps sammeln

Wie Im Konzept beschrieben sollen Informationen von installierten Apps eines Nutzers dazu beitragen die Empfehlungs-Algorithmen zu verfeinern.

Um dies zur Code-Implementierung zu simulieren, werden nun keine Daten aus MyFitnessPal bezogen, sondern aus der Google Fit App. Grund dafür ist der fehlende API-Schlüssel von Seiten MyFitnessPals. Dieser wurde zwar beantragt, doch noch nicht ausgeliefert und wird offensichtlich nicht rechtzeitig eintreffen. Die Funktionalität wurde allerdings vorerst auskommentiert, da das System für den Prototypen kompakter gemacht wurde (siehe Systemänderungen).

Android Versionskontrolle

Im Code ist zu erkennen, dass sich die Versionsabfrage auf Android 6 bezieht. Trotzdem ist die Software und die spezifische Methode auch auf älteren Systemen, z.B. 5.1, funktionstüchtig.

Der Grund für die Abfrage der neuesten Versionsnummer ist lediglich notwendig, weil die Methode und Syntax der aktuellen Android Studio Version danach verlangt, wenn man die Ortungsfunktion nutzen möchte.

Leider fehlten uns während den Tests Geräte mit älteren Android Versionsnummern als 5.0.1. Deshalb können wir nicht gewährleisten, dass die Anwendung unter der Version funktionstüchtig ist.

CompyMarks

Weil sich der Workload auf die Anwendungslogik der Empfehlungssysteme und Atmosphären-Ermittlung konzentriert hat

Systemänderungen

Ursprünglich war ein universal Werkzeug zur Aktivitäts- und Urlaubsplanung geplant. Allerdings sprengte dies den Rahmen der Programmierkenntnisse und der Zeit. Deshalb wurde das System so universell wie möglich entwickelt, aber die Abfrage für die Empfehlung bezieht sich nur noch auf Restaurants im Umkreis von 500 Metern von der TH Köln Mensa (in Gummersbach). Die Entscheidung hatte viele Gründe, einer der wichtigsten Gründe dafür war

allerdings, dass man hätte (un)endlich viele Testdaten für sämtliche Standorte und Etablissements erstellen und speichern müssen. Außerdem müssen die Testdaten ausgewertet und abgewandelt werden (siehe aktuelle Attribute der Atmosphäre). Wir haben nun Daten für folgende Orte hinterlegt: "Thairestaurant Bambusgarten Gummersbach", "Mensa TH Köln", "32 Süd", "Rhodos" und "Dornseifers". Der wichtigste Punkt ist allerdings, dass die Daten dennoch durch das System verarbeitet werden. Werden die Stellen die im Quellcode des Server markiert wurden wieder in den Code integriert (Kommentare müssen dafür gelöscht werden), sollte es möglich sein sämtliche Restaurants zu bewerten.