Proof of Concepts

Matchingsystem

Das Matching findet zwar auf dem Server statt, setzt aber auch unterem Daten vom Client voraus, um ein für den Benutzer zufriedenstellendes Ergebnis zu liefern.
Konkret bedeutet dies, der Matching-Algorithmus (im folgenden “Anwendungslogik” genannt) serverseitigen Stammdaten der Benutzer und die aktuellen clientseitigen Fahrprofile zurückgreift, um relevante Radfahrpartner zu ermitteln.

Die wesentlichen Schritte des Matching sind:

1. Client: Upload des clientseitigen Profils im JSON Format von einer Android App, Server: Aktualisierung des serverseitigen Profils.
2. Server: Ermittlung passender Radfahrpartner durch eine Datenbankabfrage auf die MongoDB Datenbank.
3. Server: Rückgabe der Ergebnisse im JSON Format.
4. Client: Darstellung der Ergebnisse.

Exit: Die Anwendungslogik kann auf alle benötigten Daten (das aktuelle Fahrprofil des Benutzers und die Stammdaten sämtlicher Benutzer) zugreifen. Das Matching liefert anschließend für den Benutzer zufriedenstellende Ergebnisse.
Sprich: Der Benutzer und die vorgeschlagenen Radfahrpartner treiben die selbe Radsportart(en), fahren etwa gleich schnell und leben vorzugsweise in der selben Stadt.

- Das Fahrprofil in der MongoDB Datenbank konnte aktualisiert werden.
- Die Merkmale der Matching-Ergebnisse stimmen mit denen des Benutzers überein:
    - Der gefundene Radfahrpartner treibt dieselbe Radsportart und lebt in einem Radius von 5km vom Wohnort des Benutzers.
    - Die durchschnittliche Fahrgeschwindigkeit eines Radfahrpartners befindet sich innerhalb eines Toleranzbereichs von -+5km/h des Benutzers.

Fail: Das Matching liefert keine und nicht zufriedenstellende Ergebnisse. Beispiel: Es werden nur Benutzer in der näheren Umgebung gefunden, die jedoch nicht die gleiche Radsportart treiben.
Die Ursachen für diese Fail-Situation könnten sein:

a) Die JSON-Daten konnten nicht oder nur fehlerhaft übertragen werden.
b) Der Matching-Algorithmus ist fehlerhaft.

Fallback:

a) Ein anderes Datenformat, welches von Android und Nodej.js unterstützt wird, muss verwendet werden, wie beispielsweise XML.
b) Einen neuen Matching-Algorithmus entwickeln.

Verarbeitung der GPS-Daten

Mit Hilfe der GPS Daten soll die Erstellung eines Fahrerprofils ermöglicht werden.
Dazu könnte im Rahmen eines Prototypen GPS Daten in einem Intervall aufgezeichnet werden und die durchschnittliche Fahrgeschwindigkeit berechnet werden. Außerdem soll mit Hilfe der GPS-Daten der Aufenthaltsort des Nutzers bestimmt werden, um Events in der Umgebung abrufen zu können.

Exit: Das System liefert akkurate Ergebnisse der Standorte. Die berechnete durchschnittliche Geschwindigkeit entspricht einem Referenzwert.
Fail: Der berechnete Wert weicht vom Referenzwert ab.
Fallback: Sollte die Position über GPS nicht ermittelbar sein, kann die Verwendung der Android Geolocation API in Erwägung gezogen werden.

Asynchrone Kommunikation mit GCM

Benachrichtigung von Events in der Umgebung werden mit Hilfe unter Verwendung eines Push Benachrichtungdienstes durchgeführt.

Dafür sind folgende Vorgänge notwendig:

- Server: Nach dem Erhalt und Analyse eines neuen Events, werden die passenden Benutzer ermittelt.
- Server: An die mobilen Endgeräte der passenden Benutzer werden Push-Nachrichten gesendet.

Exit: Der Nutzer wird automatisch benachrichtigt, sobald ein für ihn relevantes Event im System veröffentlicht wurde. Die Relevanz eines Events ergibt sich aus der Radsportart und der Veranstaltungsort. Die Push-Nachricht wurde erfolgreich dem Client zugestellt.
Fail: Die Push Nachricht wurde nicht erfolgreich übermittelt.
Fallback: Als Alternative kommen entweder andere Dienste in Frage oder die Auswahl eines anderen Messaging-Patterns wie zum Beispiel das Publish/Subscribe Prinzip, nach dem das node.js faye arbeitet.

Erstellung der Fahrprofile

Zu den stategischen Zielen gehört das Finden gleichwertiger Radfahrpartner. Um die Gleichwertigkeit zweier Radfahrpartner zu bestimmen, werden die Fahrprofile miteinander verglichen. Dies setzt vorraus, dass die Fahrprofile korrekt und vollständig sind.

Exit: Ein aussagekräftiges Fahrprofil vom Benutzer liegt vor, wenn es seine durchschnittliche Fahrgeschwindigkeit und die Radsportarten beinhaltet. //zeitpunkt und was ? datenformat
Fail: Aufgrund einer fehler- oder lückenhaften Aufzeichnung der GPS-Daten kann kein korrektes Fahrprofil erstellt werden.
Fallback: Aufzeichnungslücken schätzen.