Interne Mitfahrgelegenheit

Moritz Basel, Samuel Metzler, Dominik Weissenseel

30. Oktober 2017

Inhaltsverzeichnis

1 Einführung

1.1 Zweck des Dokuments

In diesem Dokument wird der Umfang und die Funktionalität des Systems ("die Software") Ïnterne Mitfahrgelegenheitßpezifiziert, sowie deren Architektur umrissen.

1.2 Zweck der Software

Die Software stellt ein System zur automatischen Verteilung von Mitfahrern auf Autos zur Verfügung. Zu festgesetzten Terminen stellen Teilnehmer Autos zur Verfügung oder teilen ihren Wunsch mitgenommen zu werden mit.

1.3 Zielgruppe

Die Software richtet sich an mittlere bis große Organisationen, die oft oder regelmäßig Fahrten von einem definiertem Startort zu verschiedenen Zielorten unternimmt und dabei auf private Autos der Teilnehmer setzt.

2 Beschreibung

2.1 Funktionen

Die Software stellt alle Funktionen eines webbasierten Login-Dienstes sowie die zugehörige administrative Funktionalität bereit.

2.1.1 Login-Funktionalitäten

1. Signup - Erstellen von Nutzern
2. Login - Login eines Nutzers

2.1.2 Administrative Funktionalitäten

1. Delete User - Löschen von Nutzern
2. Diagnostics - Einsehen von Daten zu allen Nutzern und Appointments

2.1.3 Appointments

Appointments stellen das Herzstück der Software dar. Nutzer bzw. Administratoren erstellen Appointments, die von allen Nutzern eingesehen werden können. Nach der Erstellung existiert ein Zeitrahmen, in dem andere Nutzer sich als MUSS-Fahrer, KANN-Fahrer oder MIT-Fahrer eintragen. In den ersten beiden Fällen geschieht dies zusammen mit einer Kapazitätsangabe. Am Ende der Frist werden alle Teilnehmer nach logischen Regeln auf die angemeldeten Autos verteilt oder es wird ein Fehlerereignis generiert, wenn zuwenig Plätze vorhanden sind. Nach der Fahrt validieren alle aktiven Fahrer ihre Fahrt und das Appointment wird archiviert.

2.2 MUSS-Kriterien

1. Klares, per TLS gesichertes System zum Einloggen, zur Übertragung der Daten und zur Persistenz derselben
2. Fähigkeit des Nutzers, Appointments zu erstellen, zu löschen und alle aktiven anzuzeigen
3. Anmelden zu einem bestehendem Appointment um mitgenommen zu werden, garantiert mit dem eigenen Auto zu fahren oder es vom System entscheiden zu lassen
4. Anzeigen der vorläufigen Fahrerverteilung zu einem Appointment
5. Klare Meldung zu unvollständigen Appointments mit Aufforderung der Lösung des Problems durch Menschen (z.B. via Email)

2.3 KANN-Kriterien

2.4 Abgrenzungskriterien

2.5 Nichtfunktionale Anforderungen

1. Sämtliche API-Calls finden ausschließlich verschlüsselt via HTTPS/TLS statt.
2. Passwörter werden nicht direkt übertragen; nur Hashes werden vom Backend empfangen
3. Die Antwortzeit des Systems darf für keine Anfrage zwei (2) Sekunden übersteigen und sollte meistens unter 0.5 Sekunden liegen.

3 User Stories

4 Architektur

Die Software wird zur Wahrung der Extensionalität strikt zwischen Backend(Server) und Frontend(Client) getrennt. Diese Praxis ist allgemein bekannt und anerkannt und erfordert eine Spezifikation einer Schnittstelle (API).

4.1 REST-Schnittstelle

Die Spezifikation der REST-API befindet sich in einem separatem Dokument. Datenaustausch findet via JSON statt.

4.2 Datenbankschema

Das Datenbankschema für die Software ist via ein E-R Diagram definiert.

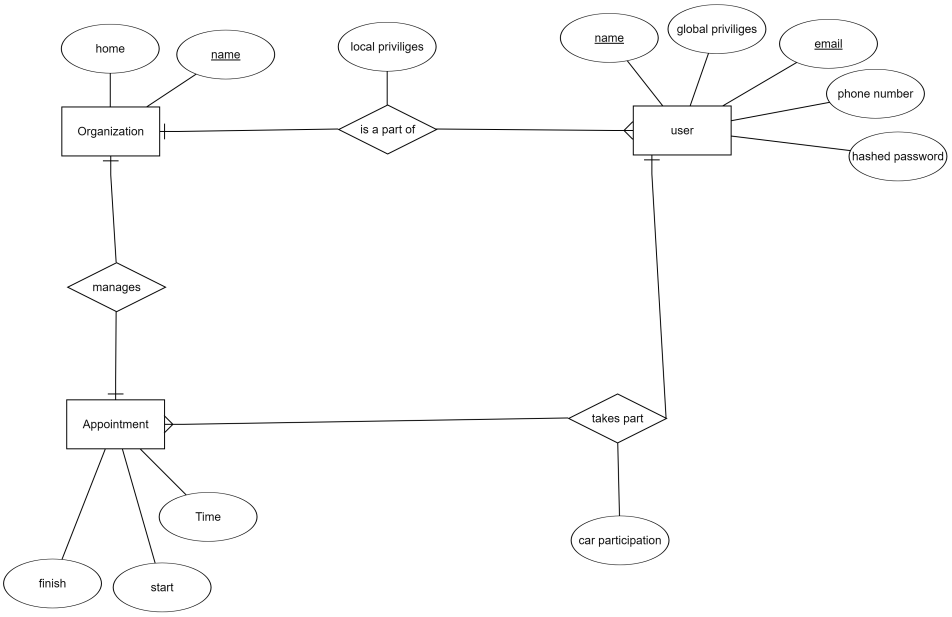

Abbildung 1: ER-Diagram für Interne Mitfahrgelegenheit