

Aufgabe

Use Case:

Ein Kunde möchte eine Software für einen Autoverleih haben. Er führt einen Kundenstamm und eine Liste mit Autos, die zum Verleih bereitstehen.

Anforderungen:

Die Software soll:

- Kundenverwaltung: Als Benutzer möchte ich Kunden verwalten können, damit ich die Kundendaten hinzufügen, bearbeiten, anzeigen und löschen kann (CRUD).
- Autos verwalten: Als Benutzer möchte ich Autos verwalten können, damit ich die Fahrzeugdaten hinzufügen, bearbeiten, anzeigen und löschen kann (CRUD).
- Vermietungsstatus verwalten: Als Benutzer möchte ich den aktuellen Vermietungsstatus verwalten, damit ich sehen kann, welches Auto aktuell vermietet ist und an welchen Kunden.
- Verleihinformationen einsehen: Als Benutzer möchte ich herausfinden, welcher Kunde gerade welches Auto gemietet hat, wie viele Autos derzeit vermietet wurden und wie viele Kilometer jedes Auto gefahren ist, damit ich die Auslastung und Nutzung der Fahrzeuge nachvollziehen kann.

Aufgabestellung

Schreiben Sie eine Client/Server Applikation, mit der diese Anforderungen erfüllt werden. Wählen Sie sinnvolle und fachgerechte Technologien/Programmiersprachen/Frameworks für die Umsetzung aus.

Das Lastenheft ist nicht komplett. Entscheiden Sie im Sinne des Kunden bei den nicht definierten Punkten, um ein MVP der Applikation präsentieren zu können. Seien Sie kreativ 😊.

Mögliche Herausforderungen:

Achten Sie darauf:

- dass sie bei der Erstellung der Datenbank diese normalisieren und so strukturieren, dass diese Sinn macht.
- Denken Sie auch an die Möglichkeit, dass der Kunde den Autoverleih irgendwann hoch skalieren möchte und ggf. auch größere Queries abgefragt werden können.
- Überlegen Sie sich ein sinnvolles GUI-Konzept.
- Das Ziel ist es innerhalb von 4-5 Stunden die Software so fertig wie möglich zu stellen, Sie müssen allerdings nicht komplett fertig werden (MVP).

Deployment

Schreiben Sie anschließend einen kurzen Bericht (max. 1 Seite) in englischer Sprache, in dem Sie die Software erklären und warum Sie bestimmte Techniken / Datenbanken / Programmiersprachen / Frameworks / Patterns etc. benutzt haben und argumentieren Sie, warum Sie sich nicht anders entschieden haben, bzw. begründen Sie Ihre Überlegungen. Schicken Sie uns den Link zum versionierten Quellcode in GitHub etwa einen Tag vor dem technischen Gespräch. Das Readme des Repositorys sollte eine Beschreibung beinhalten, wie das Projekt gebuildet und gestartet werden kann.