



## Labor App 1 – Car Sharing

*Teilnahme am Labor wird nur für diejenigen registriert, die etwas vorzeigen!*

*Die Laboraufgabe ist in eine Untergruppe zu bearbeiten.*

## App Beschreibung

Die Anwendung soll einem Car-Sharing-Geschäft zur Verwaltung des Lagers dienen.

## Funktionale Anforderungen

### Autorisierung

A1. Es werden mehrere Benutzer zugelassen. Jedoch gibt es folgenden Arten von Benutzer:

- Kunden
- Mitarbeiter
- Admins

A2. Ein Benutzer wird authentifiziert über E-Mail und Passwort. Bevor man die Anwendung startet, die E-Mail und das Passwort werden erstmals überprüft. Wenn die beiden stimmen, dann kann man das App bedienen.

A3. Nach dem App-Start, einem Benutzer wird gezeigt nur die Aktionen den er/sie bedienen kann.

### Mitarbeiter

M1. Mitarbeiter verfügen über folgende Informationen:

M1.1 Name, Vorname, E-Mail, Position, Geburtsdatum, Kürzel, Gehalt (EUR), Bemerkungen.

Beispiel: Mustermann, Max, [max.mustermann@test.de](mailto:max.mustermann@test.de), Verkäufer, 3.7.1996, MM, 2100.

M2. Admins sind Mitarbeiter mit Rechten:

M2.1 andere Mitarbeiter zu verwalten.

M2.2 anderen Mitarbeiter den Gehalt sehen/anpassen.

M3. Mitarbeiter verwalten:

M3.1 Mitarbeiter erstellen.

M3.2 Mitarbeiter ändern.

M3.3 Mitarbeiter aktivieren/deaktivieren.

M3.4 Mitarbeiter löschen.

M3.5 Mitarbeiter Admin-Rechte zuweisen/entfernen.

M3.6 Mitarbeiter Passwort zurücksetzen.

M4. Mitarbeiter abfragen (aufpassen: nur Admins dürfen Gehälter sehen!)

M4.1 Alle Mitarbeiter auflisten, nach Nachnamen und Vornamen sortiert.

M4.2 Mitarbeiter nach Kürzel suchen.

M4.3 Mitarbeiter nach E-Mail suchen.

M4.4 Mitarbeiter nach Namen und Vornamen suchen.

M4.5 Mitarbeiter mit Geburtsdatum innerhalb eines Zeitraums suchen.

M5. Eigener Profil anpassen (Mitarbeiter)

M5.1 Passwort ändern.

M5.2 Bemerkungen ändern.

M6. Validierungen

M6.1 Das Gehalt muss mindestens 100 EUR sein.

M6.2 Das Alter der Mitarbeiter darf nicht negativ oder größer als 80 sein.



## Labor App 1 – Car Sharing

*Teilnahme am Labor wird nur für diejenigen registriert, die etwas vorzeigen!*

*Die Laboraufgabe ist in eine Untergruppe zu bearbeiten.*

M6.3 Die E-Mail und Kürzel müssen jeweils eindeutig sein.

M6.4 Alle Werte sind Pflicht, bis auf Position und Bemerkungen.

### Kundenverwaltung

K1. Jeder Kunde wird durch die folgenden Charakteristika beschrieben:

K1.1 Name, Vorname, E-Mail, Handy, Adresse, Bemerkungen, GdprDeleted(boolean).

K2. Nur Mitarbeiter dürfen:

K2.1 Kunden erstellen.

K2.2 Kunden ändern.

K2.3 Kunden löschen. Ein Kunde darf nicht gelöscht werden, wenn schon einmal ein Auto bestellt hat.

K2.4 Kunde nach GDPR anonymisieren. Folgende Werte werden in den Felder eingetragen:

Name <- "Kunde-{Laufendennummer oder ID}"

Vorname <- „Unbekannt“

E-Mail <- null

Handy <- null

Adresse <- null

Bemerkungen <- null

GdprDeleted <- true

K3. Kunden abfragen (Nur Mitarbeiter dürfen Kunden sehen)

K3.1 Alle Kunden auflisten, nach Nachnamen und Vornamen sortiert.

K3.2 Kunde nach E-Mail suchen.

K3.3 Kunde nach Handy suchen.

K3.4 Kunde nach Namen und Vornamen suchen.

K3.5 Alle Kunden suchen, die ein bestimmtes Auto bestellt haben, absteigend nach Bestelldatum sortiert.

K4. Eigener Profil anpassen (Kunde)

K4.1 Passwort ändern.

K4.2 Bemerkungen ändern.

K4.3 Ein Auto in/von die/der Favoritenliste hinzuzufügen/entfernen.

K4.4 Die Favoritenliste ansehen.

K5. Validierungen

K5.1 Die E-Mail muss eindeutig sein.

K5.2 Die E-Mail muss ein entsprechendes Format haben.

K5.3 Die Handy darf nur + (am Anfang, optional), Zahlen und Leerzeichen beinhalten.

K5.4 Alle Werte sind Pflicht, bis auf Bemerkungen, wenn GdprDeleted ist **false**.

K5.5 Name und Vorname sind Pflicht, wenn GdprDeleted ist **true**.

### Autos-Verwaltung

V1. Jedes Auto wird durch die folgenden Charakteristika beschrieben:

V1.1 Kennzeichen, Modell, Marke, Erstzulassungsjahr, Kilometerstand, Tagespreis, Kraftstoffart (Diesel/ Benzin/ Elektro/ Gas), Getriebe (Automatik/ Schaltgetriebe), Farbe, Bemerkung.

V2. Nur Mitarbeiter dürfen:

V2.1 Autos erstellen.

V2.2 Autos ändern.



## Labor App 1 – Car Sharing

**Teilnahme am Labor wird nur für diejenigen registriert, die etwas vorzeigen!**

**Die Laboraufgabe ist in eine Untergruppe zu bearbeiten.**

V2.3 Autos löschen. Ein Auto darf nicht gelöscht werden, wenn schon einmal bestellt wurde.

V2.3 Autos deaktivieren. Ein Auto darf nicht deaktiviert werden, wenn zum aktuellen Datum noch bestellt ist.

V3. Autos abfragen (alle Benutzer dürfen Autos abfragen)

V3.1 Alle Autos auflisten, die in einen bestimmten Zeitraum zur Verfügung stehen (sind nicht bestellt), aufsteigend nach Tagespreis sortiert. Optional kann man Marke, Modell, Getriebe, und Tagespreis-Spanne angeben.

V3.2 Auto nach Kennzeichen suchen.

V3.3 Autos, die von einem Kunde bestellt wurden, absteigend nach Bestelldatum sortiert.

V4. Validierungen

V4.1 Auto Kennzeichen sind eindeutig und dürfen nicht leer sein.

V4.2 Auto Kennzeichen dürfen nur Buchstaben und Zahlen beinhalten. Kleinbuchstaben werden in Großbuchstaben konvertiert.

V4.2 Erstzulassungsjahr darf nicht kleiner als 2000 sein und nicht in die Zukunft liegen.

V4.3 Kilometerstand und Tagespreis darf nicht weniger als 0 sein.

## Auto Bestellung

B1. Jede Bestellung wird durch die folgenden Charakteristika beschrieben:

B1.1 Bestellungsnummer, Bestelldatum, Status (Reservation/Order/Completed), Auto, Beginn, Ende, Kunde, Mitarbeiter (wer den Vertrag abgeschlossen hat), Gesamtsumme, Bemerkung.

B2. Nur Mitarbeiter dürfen:

B2.1 Bestellungen als Order anlegen, oder Reservierungen als feste Bestellung (Order) umändern.

B2.2 Bestellungen ändern. Folgende Regeln gelten:

B2.2.1 Bestellungen in Status Completed dürfen nicht mehr geändert werden.

B2.2.2 Mitarbeiter dürfen nur eigene Bestellungen ändern, nicht die von deren Kollegen.

B2.2.3 Admins dürfen alle Bestellungen ändern.

B2.3 Bestellungen auf Completed setzen. Die App wird zu diesem Zeitpunkt die Gesamtsumme automatisch berechnen, anhand von Autos Tagespreis und Anzahl der Tage.

B2.4 Bestellungen übernehmen. Ein Mitarbeiter kann sich selbst in die Bestellung oder Reservierung von einem anderen Mitarbeiter sich eintragen.

B2.5 Bestellungen abgeben. Ein Mitarbeiter kann in seine Bestellungen oder Reservierungen einen anderen Mitarbeiter eintragen.

B3. Kunden und Mitarbeiter dürfen:

B3.1 Reservierungen löschen. Mitarbeiter dürfen alle Reservierungen, Kunden nur die eigenen.

B3.2 Reservierungen anlegen. Mitarbeiter müssen der Kunde angeben, für Kunden wird automatisch der Kunde vorgegeben, ohne die Möglichkeit zu ändern.

B3.3 Reservierungen ändern. Mitarbeiter dürfen alle Reservierungen, Kunden nur die eigenen.

B4. Bestellungen abfragen (Mitarbeiter dürfen alle Bestellungen abfragen und Kunden dürfen nur eigene Bestellungen abfragen)

B4.1 Alle Bestellungen auflisten, die in einen bestimmten Zeitraum definiert sind, aufsteigend nach Gesamtsumme sortiert. Optional kann man Auto, Kunde, oder/und Mitarbeiter angeben.

B4.2 Bestellung nach Bestellungsnummer suchen.

B4.3 Abfragen die Gesamtsumme in einem Jahr oder in einem Monat, optional nach Auto, Kunde, oder Mitarbeiter gruppiert.

B5. Validierungen

B5.1 Reservierungen oder Bestellungen dürfen nicht überlappen, d.h. für einen Auto kann an einem bestimmten Tag nur eine Reservierung/Bestellung erfolgen.



## Labor App 1 – Car Sharing

*Teilnahme am Labor wird nur für diejenigen registriert, die etwas vorzeigen!*

*Die Laboraufgabe ist in eine Untergruppe zu bearbeiten.*

- 
- B5.2 Beginn muss gleich oder weniger als Ende sein.
  - B5.3 Ein Kunde darf nicht mehr als 5 Reservierungen haben.
  - B5.4 Die Bestellungsnummer sind eindeutig und werden automatisch vom System belegt.
  - B5.5 Wenn ein Kunde eine Reservierung erstellt, ist der Mitarbeiter Feld nicht belegt.
  - B5.6 Das Bestelldatum wird immer mit dem aktuellen Datum belegt.

## Nicht-funktionale Anforderungen

### Weitere Anforderungen

- W1. Die Interaktion des Benutzers mit dem Programm passiert via **Konsole**. Dem Benutzer werden dort das Menu und Ergebnissen angezeigt, und dort auch die Befehle des Benutzers eingelesen (User Interface, UI).
- W2. Das Programm soll bei Start Beispieldaten von 10 oder mehr Entitäten beinhalten, je Bereich.

### Qualitäts-Anforderungen

- Q1. Ausnahmen (z.B. ungültige Eingaben) sollen sorgfältig behandelt werden.
- Q2. Code sollte durchwegs in Englisch geschrieben sein und **selbsterklärende und zweckgemäße Namen** für Klassen, Methoden und Variablen verwenden.
- Q3. Um sicherzustellen, dass die Komponenten funktionieren, ohne Verwendung der Benutzeroberfläche, sind Unit-Tests erforderlich.

## Abgabekriterien

- K1. Die Lösung wird von einem Team entwickelt (eine Untergruppe) in einen GitHub Repository.
- K2. Jedes Teammitglied soll das Ergebnis einreichen als eigener Commit und den Code erklären können. Es wird nicht akzeptiert, dass Teammitglieder für andere Kollegen Code commiten oder pushen.
- K3. Die Datei **Main.cpp** enthält `main()`
- K4. Es ist nur eine Klasse per Datei erlaubt.

## Empfehlungen

- R1. Vereinbart im Team zuallererst die Schnittstellen der Bereiche; das ermöglicht danach parallele Arbeit an der Implementierung verschiedener Teile.
- R2. Die Anforderungen können „versteckte“ Informationen beinhalten. Z.B. aktivieren/deaktivieren von Entitäten kann dazu führen ein Attribut „Aktiv“ bei der Entität hinzuzufügen. Wenn nicht klar, bitte mit Kollegen oder Professoren abstimmen.
- R3. Die eindeutigen IDs können mithilfe aktueller Zeitstempel generiert werden.



## Labor App 1 – Car Sharing

*Teilnahme am Labor wird nur für diejenigen registriert, die etwas vorzeigen!*

*Die Laboraufgabe ist in eine Untergruppe zu bearbeiten.*

---

R4. Es gibt nicht nur eine einzige Lösung. Alle Lösungen, die die Anforderungen richtig implementieren sind als gültig anerkannt.