Dokumentation CarRent

# Einführung und Ziele

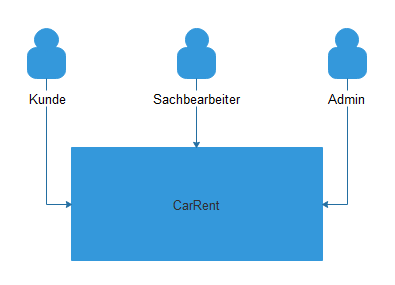
Das Ziel dieses Projektes ist ein Autovermietungssystem. Es wird eine Applikation mit Web-API erstellt, welche eine Einbindung einer WebApp ermöglich. Die Daten warden in einer relationalen Datenbank gespeichert.

# Randbedingungen & Kontextabgrenzung

Aufgrund des Zeitdruckes sowie fehlenden Wissens wird auf die Implementierung der WebApp verzichtet. Die Abgabe erfolgt via Github, es wird keine Infrastruktur aufgesetzt.

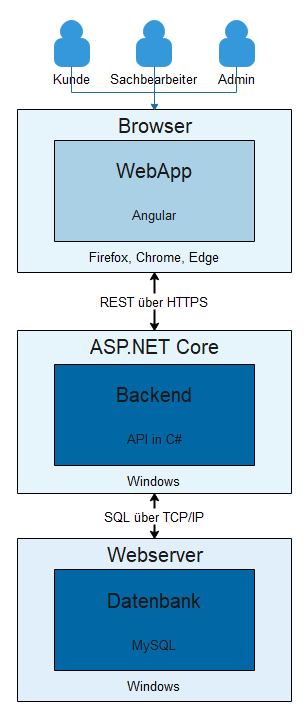
## Context

Das Context-Diagram zeigt die beteiligten User am System.



## Containers

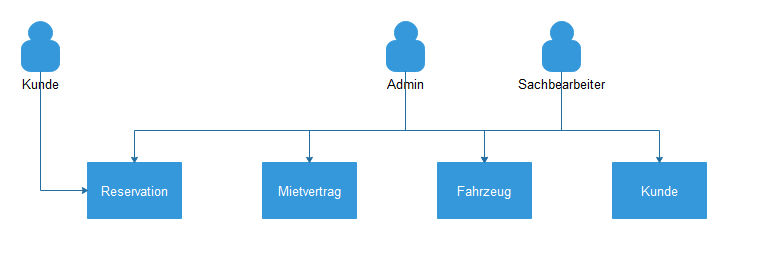
Folgender Aufbau wurde für die Umsetzung gewählt:



Dabei werden zum aktuellen Zeitpunkt nur die dunkelblauen Komponenten umgesetzt.

## Components

Die Struktur der Applikation wird in folgende vier Komponenten aufgeteilt und entsprechend von den Benutzergruppen verwendet:



# Use-Cases

## UC01: Fahrzeug reservieren

Der Kunde wählt eine Fahrzeugklasse sowie die Mietdauer und das Startdatum aus. Ihm wird der kalkulierte Gesamtpreis angezeigt und er schliesst nach Eingabe seiner Angaben (Adresse, Kreditkartenangaben, Geburtsdatum) die Reservation ab. Er erhält dafür eine Reservationsnummer. Das Fahrzeug wird als reserviert markiert. Ein Depot wird auf der Kreditkarte reserviert.

## UC02: Fahrzeug abholen

Der Sachbearbeiter macht bei Abholung des Fahrzeugs die Reservation anhand der Reservationsnummer ausfindig und wandelt diese in einen Mietvertrag um.

## UC03: Fahrzeug zurückgeben

Der Sachbearbeiter prüft das Fahrzeug auf Schäden und erfasst diese im System. Er nimmt das Fahrzeug zurück und schliesst den Vertrag ab. Dadurch wird das Fahrzeug wieder zur Reservation freigegeben.

## UC04: Fahrzeug verwalten

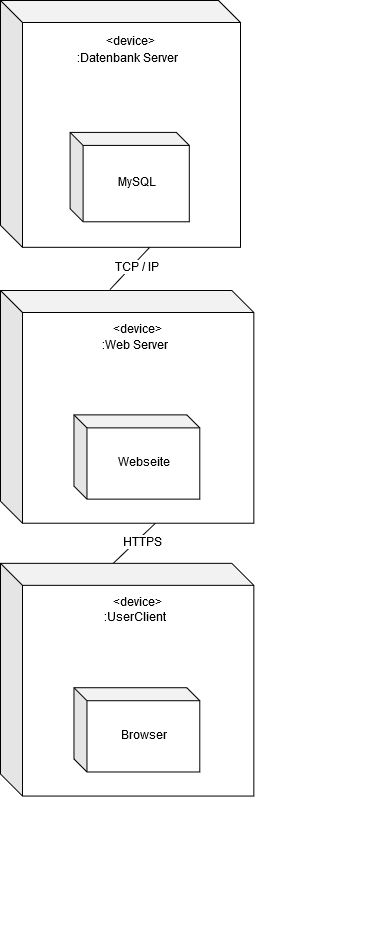
Der Sachbearbeiter sucht und verwaltet die Eigenschaften Klasse, Marke und Typ eines Fahrzeuges.

## UC05: Kunde verwalten

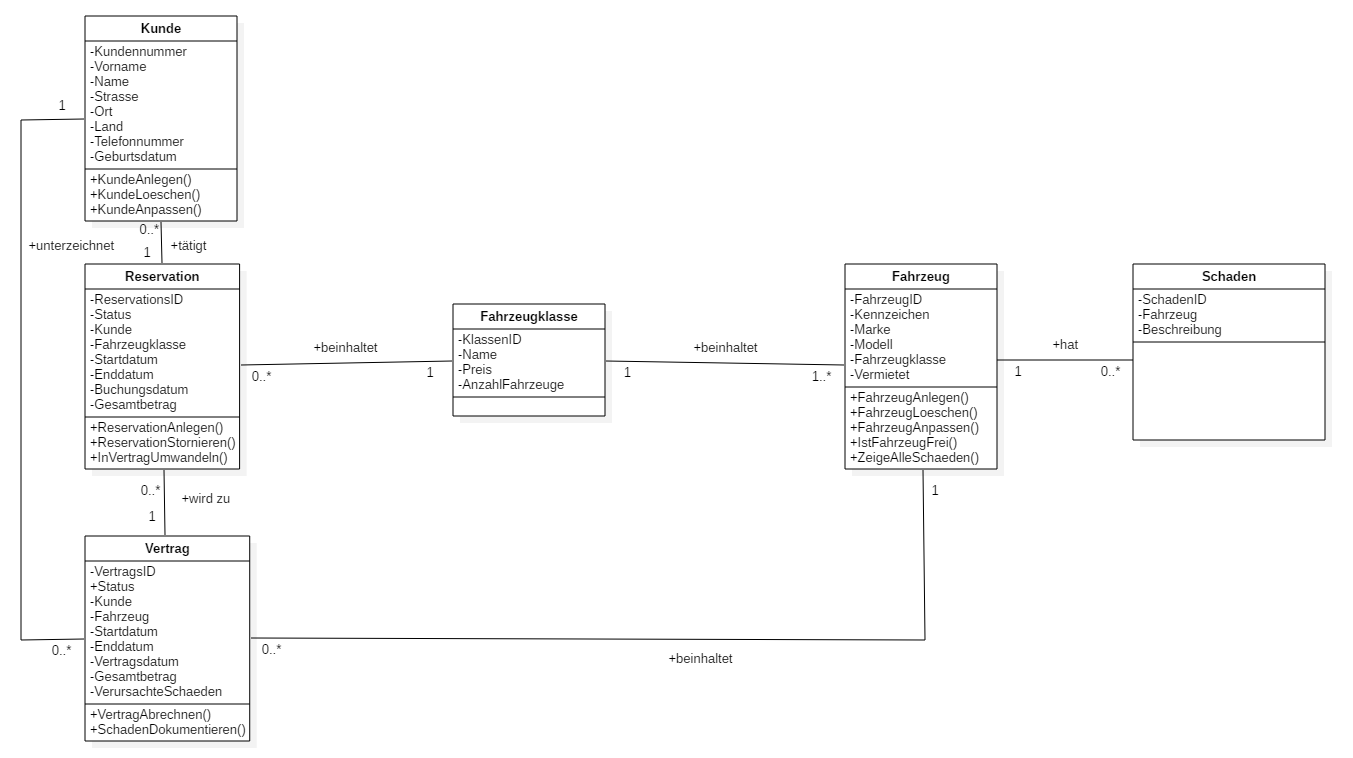
Der Sachbearbeiter sucht den Kunden mittels Namen oder Kundennummer und verwaltet dessen Angaben wie Namen, Adresse und Kundennummer.

# Bausteinsicht

## Deployment View



## Logical View



## Use Case View

