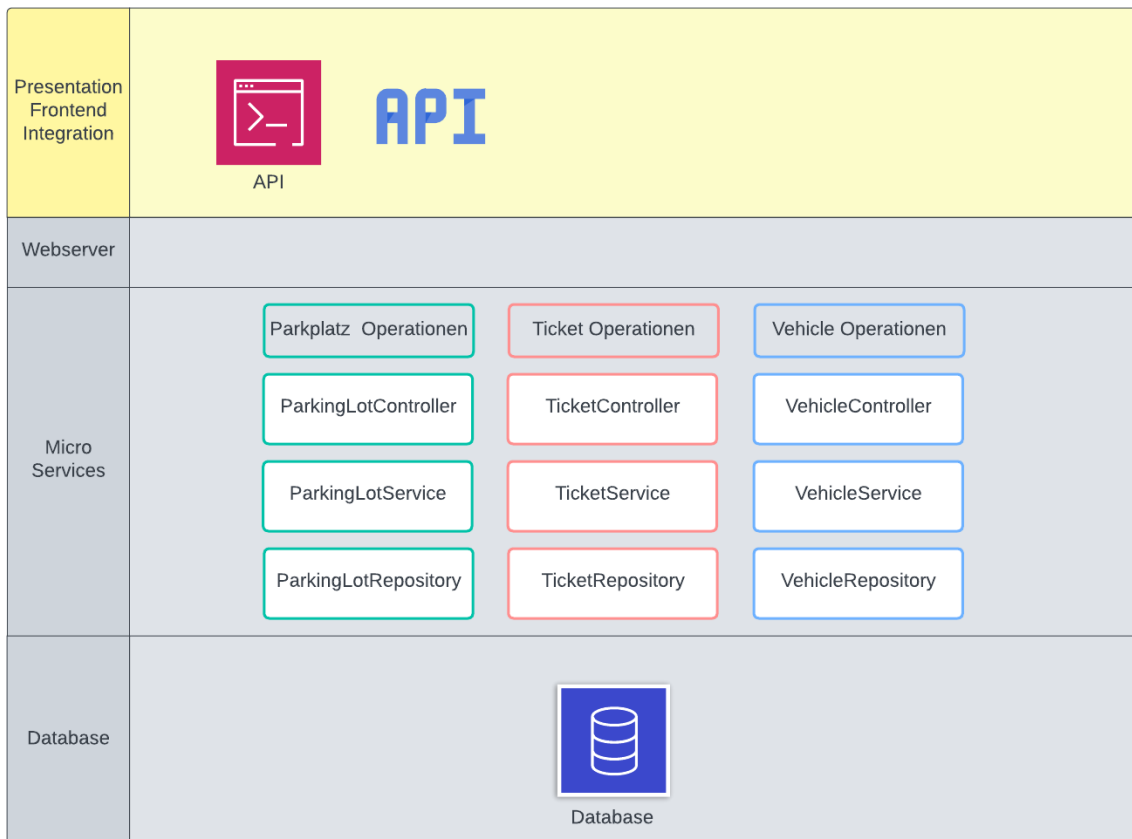


Altay Yavuz Elmas - 200503046
Ahmet Efe Kutbay - 200503019

Architekturdokumentation

Chapter 1: Architekturüberblick



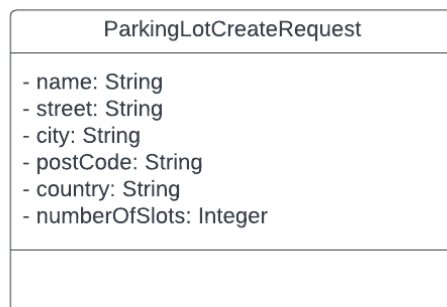
Chapter 2: Beschreibung der Controller Klassen

ParkingLotController

1) createParkingLot

POST <http://localhost:8080/parking-lot/create>

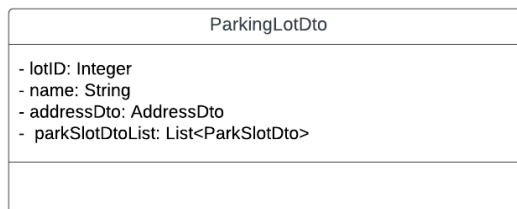
Eine Methode, die Vorgänge wie das Erstellen eines neuen Parkplatzes und das Aktualisieren von Parkinformationen regelt, die für die Parkplatzverwaltung erforderlich sind.



2) getAll

GET <http://localhost:8080/parking-lot/getAll>

Eine Methode, die das Auflisten von DTOs im System ermöglicht.



3) getParkingLotById

GET <http://localhost:8080/parking-lot/{lotID}>

Die Methode, die den DTO des Parkplatzes mit der angeforderten ID auflistet.

ParkingLotDto
<ul style="list-style-type: none">- lotID: Integer- name: String- addressDto: AddressDto- parkSlotDtoList: List<ParkSlotDto>

ParkingLotResponse
<ul style="list-style-type: none">- parkingLotDto: ParkingLotDto- isSuccess: boolean- message:String

AddressDto
<ul style="list-style-type: none">- addressID: Integer- street: String- city: String- postCode: String- country: String- lotID: Integer

4) deleteParkingLot

DELETE <http://localhost:8080/parking-lot/delete/{lotID}>

Die Methode, die den Parkplatz mit der gewünschten ID-Nummer aus dem System löscht.

ParkingLotResponse
<ul style="list-style-type: none">- parkingLotDto: ParkingLotDto- isSuccess: boolean- message:String

ParkingLotDto
<ul style="list-style-type: none">- lotID: Integer- name: String- addressDto: AddressDto- parkSlotDtoList: List<ParkSlotDto>

5) updateParkingLot

POST <http://localhost:8080/parking-lot/update/{lotID}>

Eine Methode, die Vorgänge wie das Aktualisieren von Parkinformationen regelt, die für die Parkplatzverwaltung erforderlich sind.

ParkingLotResponse
- parkingLotDto: ParkingLotDto - isSuccess: boolean - message:String

ParkingLotDto
- lotID: Integer - name: String - addressDto: AddressDto - parkSlotDtoList: List<ParkSlotDto>

TicketController

1) createTicket

POST <http://localhost:8080/ticket/create>

Die Methode, die Tickets mit den notwendigen Informationen für Fahrzeuge erstellt, die auf den Parkplatz fahren.

TicketResponse
- ticketDto: TicketDto - isSuccess: boolean - message: String

TicketDto
- ticketID: Integer - entryDate: Date - exitDate: Date - isValid: boolean - licencePlate: String - slotID: Integer - lotID: Integer

2) parkOut

PATCH <http://localhost:8080/ticket/{ticketID}>

Die Methode, die das Fahrzeug mit der angeforderten Ticket-ID aus dem System entfernt.

TicketResponse
<ul style="list-style-type: none">- ticketDto: TicketDto- isSuccess: boolean- message: String

TicketDto
<ul style="list-style-type: none">- ticketID: Integer- entryDate: Date- exitDate: Date- isValid: boolean- licencePlate: String- slotID: Integer- lotID: Integer

3) getAll

GET <http://localhost:8080/ticket/getAll>

Eine Methode, die DTOs von Tickets im System auflistet.

TicketDto
<ul style="list-style-type: none">- ticketID: Integer- entryDate: Date- exitDate: Date- isValid: boolean- licencePlate: String- slotID: Integer- lotID: Integer

VehicleController

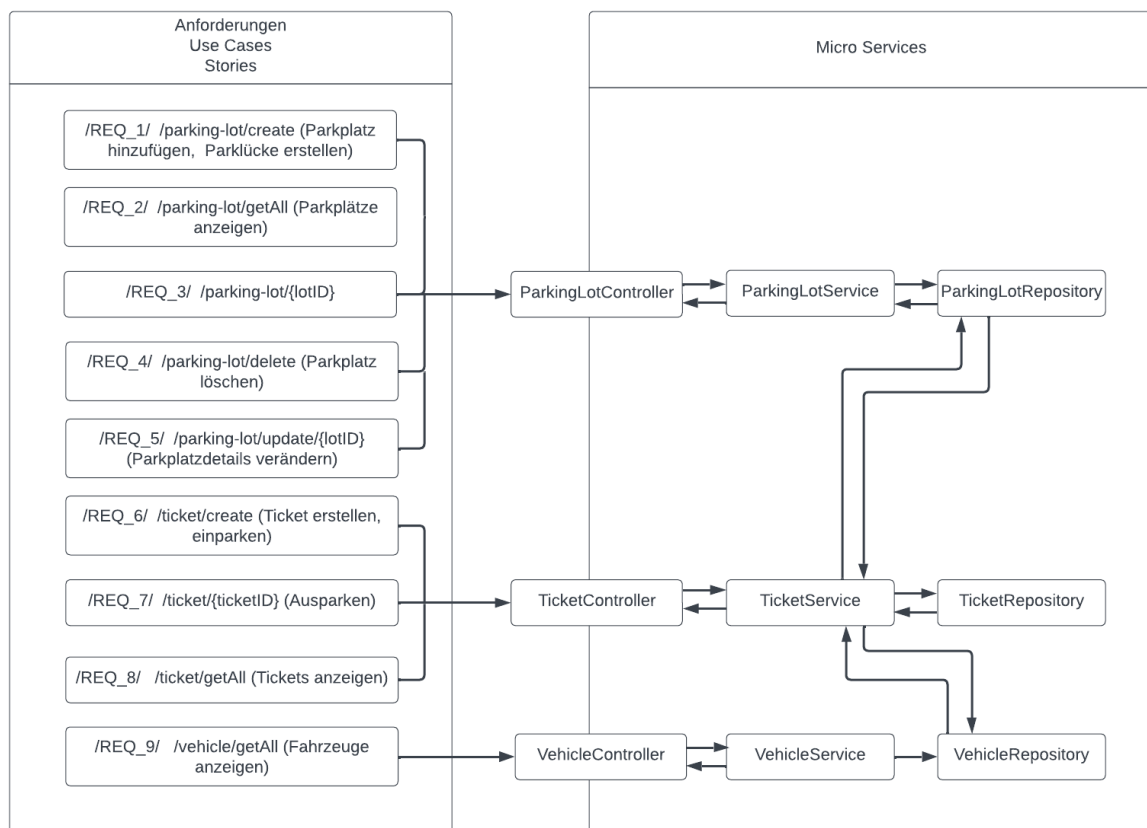
1) getAll

GET <http://localhost:8080/vehicle/getAll>

Die Methode, die auf den Parkplätzen gefundene Fahrzeuge auflistet.

VehicleDto
<ul style="list-style-type: none">- licencePlate: String- vehicleType: VehicleType- ticketID: Integer

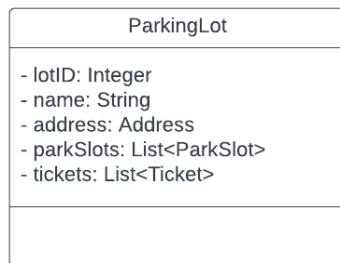
Chapter 3: Rückverfolgbarkeit der Anforderungen



Chapter 4: Beschreibung der DB-Zugriffsschicht(Daten-Modelle)

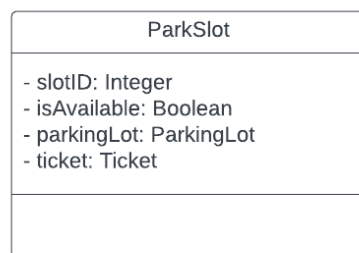
1) ParkingLot

Datenbank-Zugriff für Parkplätze ist über ParkingLotRepository möglich.



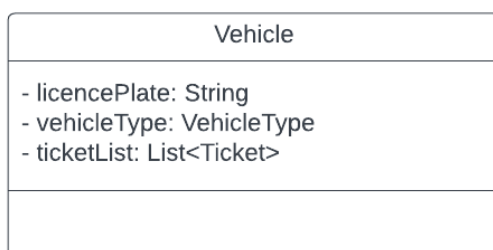
2) ParkSlot

Datenbank-Zugriff für Parklücken ist über ParkingLotRepository möglich.



3) Vehicle

Datenbank-Zugriff für Fahrzeuge ist über VehicleRepository möglich.



4) Address

Datenbank-Zugriff für Adresse ist über ParkingLotRepository möglich.

Address
<ul style="list-style-type: none">- addressID: Integer- street: String- city: String- postCode: String- country: String- parkingLot: parkingLot

5) Ticket

Datenbank-Zugriff für Tickets ist über TicketRepository möglich.

Ticket
<ul style="list-style-type: none">- ticketID: Integer- entryDate: Date- exitDate: Date- isValid: boolean- vehicle: Vehicle- parkSlot: ParkSlot- parkingLot: ParkingLot