Parking lot system
The parking lot can have multiple floors where each floor carries multiple slots. Each slot can have a single vehicle parked in it. We can incorporate an automatic ticketing system in the design so customers can park th cars without any human intervention

Functional Requirements

1. The parking lot includes multiple entry and exit points.
2. Multiple parking floors are available.
3. Each floor carries multiple rows of parking slots.
4. The parking lot supports parking of different types of vehicles, including motorcycles, cars, and buses.
5. The parking lot has motorcycle parking spots, compact spots, and large spots.
6. The customer can collect a parking ticket from the entry point and pay the parking fee at any of the exit points

Skip asking difficult requirements to make your interview easier
like what if 2 cars come at the same time

Focus on the main/basic requirements

Class Hierarchy

class Vehicle
attr Car: Type<enum> / inheritance
2. what else the Vehicle can have
give as more as you can
plateNumber

class ParkingSpot
id
type
floorNumber
entryTime
exitTime
isOccupied

class Ticket
parkingSpotId
fee

class ParkingLot
List<ParkingSpot> floor1
List<ParkingSpot> floor2
List<ParkingSpot> floor3
A better design need

Database Structure

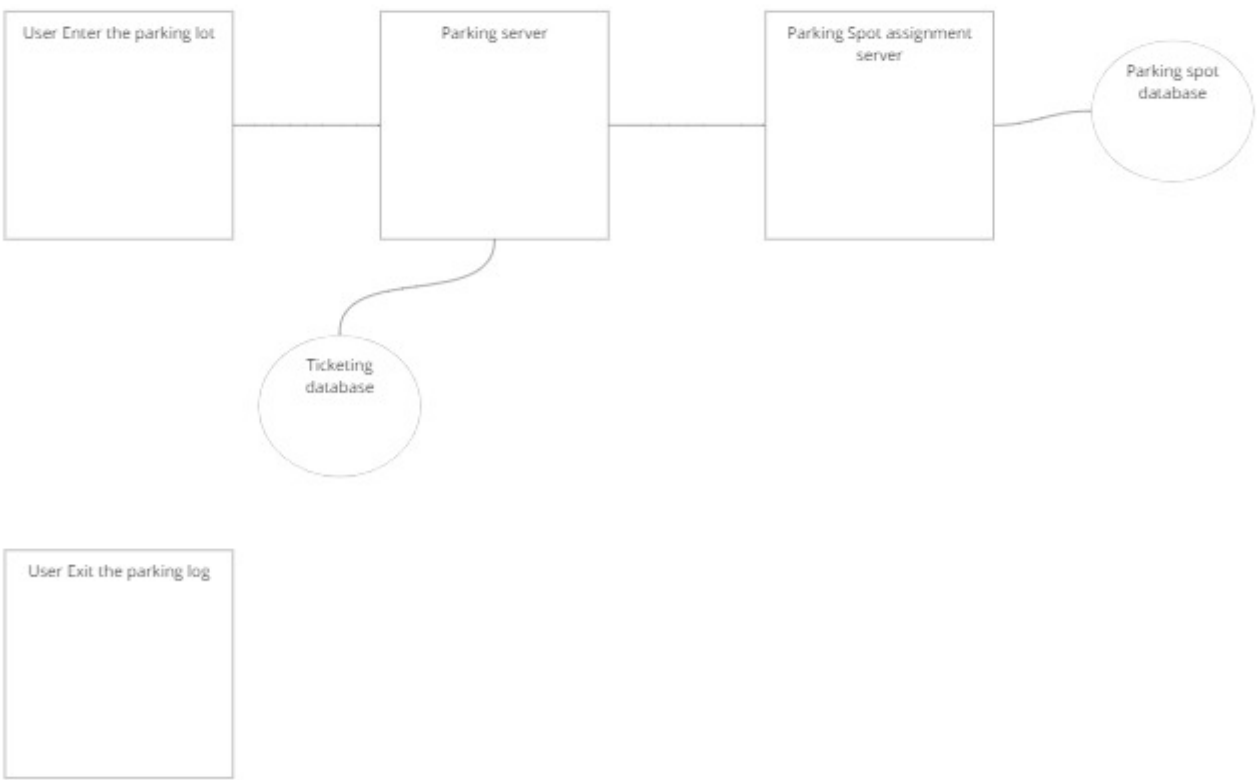
Type of database

Tables

Indexing

Partition split up data to multiple database instances

Workflow Design -> from user entry to user exist



Follow-up questions

1. Always assign the nearest spot based on the user entry location
2. Parking lot container optimization