Parking lot is an open area designated for parking cars. We will design a parking lot where a certain number of cars can be parked for a certain amount of time. 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 their cars without any human intervention.

**REQUIREMENT**

**FUNCTIONAL REQUIREMENT**

1. The system should be able to check, manage and track about how much parking space available or not.
2. The system should be able to calculate the amount of price according to the time the vehicle is parked.
3. Customers should be able to book the parking slot in advance.
4. The system should be able to keep track of the time a vehicle is parked.
5. System should allow online booking and payment integration.
6. The system should be able to calculate the number of hours a vehicle has been parked.

**NON FUNCTIONAL REQUIREMENTS**

1. Low latency
2. High availability
3. Secure with high performance

**Estimation**

Total Users = 10 million

Daily active users = 1million

Assume we can park 500 vehicles

Estimate storage per parking vehicle = 100bytes

Avg storage for 500 parking vehicle = 100\*500 = 50KB

Monthly avg storage = 50\*30 = 1.5MB

**Database schema or data design model**

User(userId, name, email)

ParkingLot(Id, Name, location, capacity)

ParkingSpace(Id, ParkingLotId, type, status, vehicleType)

Vehicle(Id, licensePlate, model, color, entryTime, exitTime)

**API Design**

**1. Check Parking Availability:**

**GET /api/parking-lot/{parking\_lot\_id}/availability**

**2. Reserve Parking Space:**

**POST /api/parking-lot/{parking\_lot\_id}/reserve**

**3. Enter Parking Lot:**

**POST /api/parking-lot/{parking\_lot\_id}/enter**

**4. Exit Parking Lot:**

**POST /api/parking-lot/{parking\_lot\_id}/exit**

**5. Get User Parking History:**

**GET /api/users/{user\_id}/parking-history**

**6. Manage Parking Lot Information:**

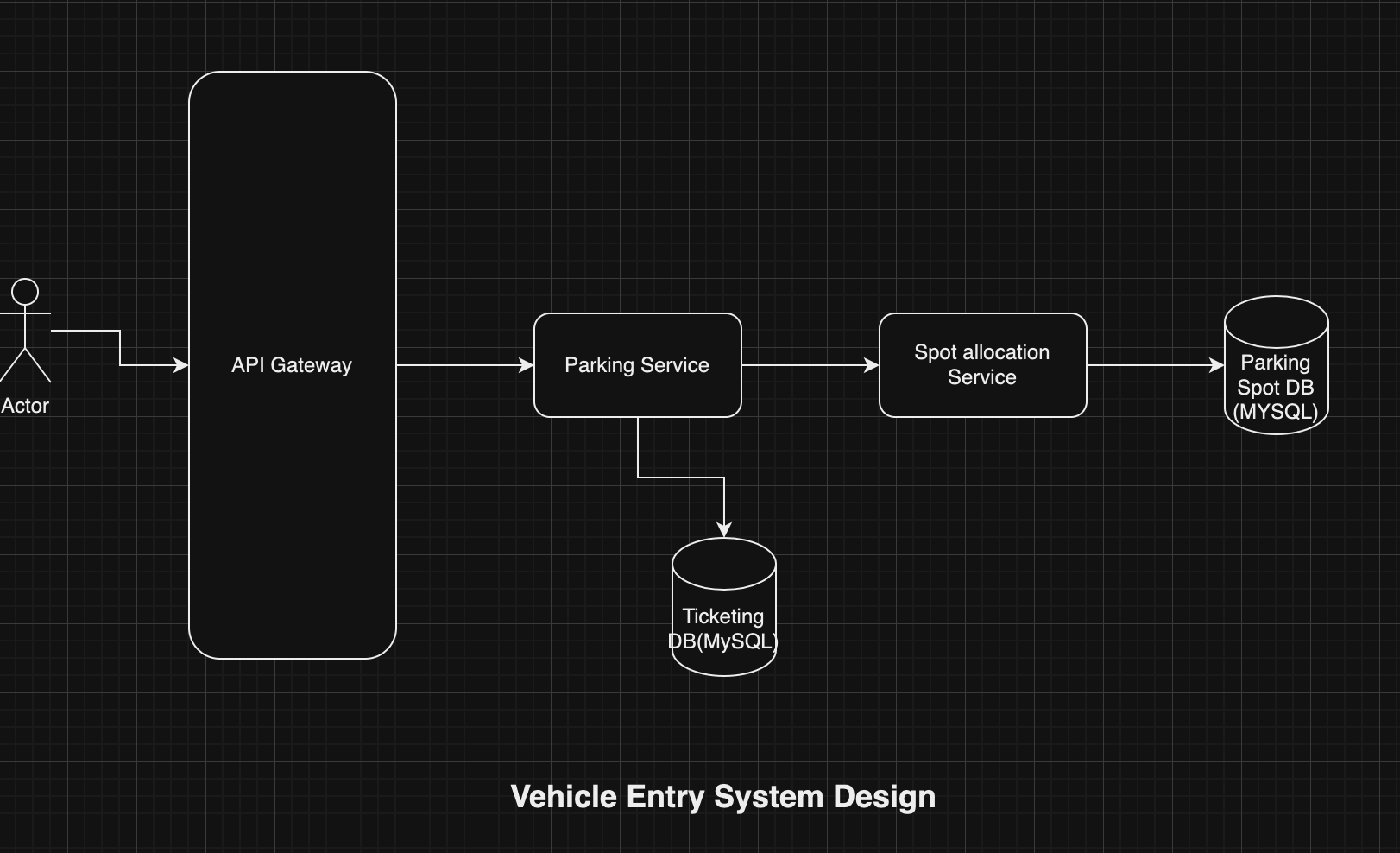
**GET /api/parking-lot/{parking\_lot\_id}**

**High Level Design**

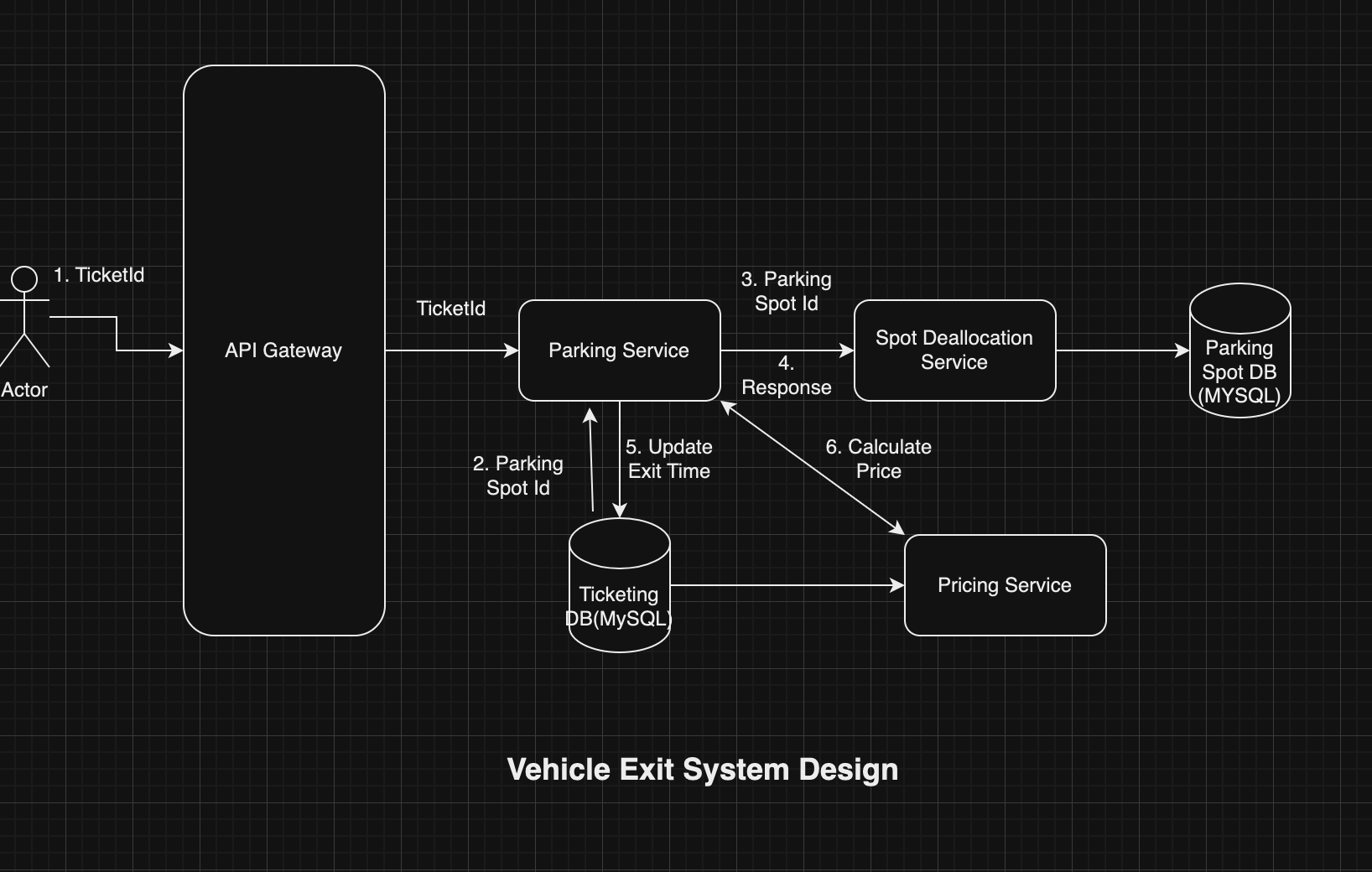
**system primarily manages two functions: vehicle entry and exit**

**Vehicle entry:**

When a user comes into the parking lot to park a vehicle, the request goes to a parking service for entry. The parking service calls the spot allocation service to scan the database and pick and allocate a previously vacant spot for this vehicle. We’ve already covered the strategy that the spot allocation service uses to find and allocate a vacant spot to the vehicle.

****

**Vehicle Exit**

****

**Reference**

**https://medium.com/double-pointer/system-design-interview-parking-lot-system-ff2c58167651**