

# Rising from Failure

## LLD3 Machine Coding Interview Experience



**Q1. Can you give an intro yourself?**

Ans: I have given my intro

**Q2. Design Car Rental System. List down you the requirement**

## **My Answer**

- 1. Users have the option to register an account and sign in.**
- 2. Users can manage their profiles.**
- 3. Location-based search functionality is available.**
- 4. Results are presented according to stores.**
- 5. Users can select and reserve a vehicle.**
- 6. Payment transactions can be completed.**
- 7. Vehicle submissions are enabled for users.**

### **Q3. Can you add more features on it**

Ans. Sure, we can add more features, but I believe I've included all the usual things. Can you tell me if there's anything else you'd like to add or any special features you're thinking of?

### **Q4. Final question asked by Interviewer from leetcode**

# Requirements

We will focus on the following set of requirements while designing our Car Rental System:

- The system will support the renting of different automobiles like cars, trucks, SUVs, vans, and motorcycles.
- Each vehicle should be added with a unique barcode and other details, including a parking stall number which helps to locate the vehicle.
- The system should be able to retrieve information like which member took a particular vehicle or what vehicles have been rented out by a specific member.
- The system should collect a late-fee for vehicles returned after the due date.
- Members should be able to search the vehicle inventory and reserve any available vehicle.

# Requirements

- The system should be able to send notifications whenever the reservation is approaching the pick-up date, as well as when the vehicle is nearing the due date or has not been returned within the due date.
- The system will be able to read barcodes from vehicles.
- Members should be able to cancel their reservations.
- The system should maintain a vehicle log to track all events related to the vehicles.
- Members can add rental insurance to their reservation.
- Members can rent additional equipment, like navigation, child seat, ski rack, etc.
- Members can add additional services to their reservation, such as roadside assistance, additional driver, wifi, etc.

# My Approach

## ParkingManagmentSystem

-----  
List<user>, List<Parking>, crud operations

## Member

-----  
userId, name, vehicleLicence

## Parking

-----  
parkingId, location, VehicleInventory , List<Reservations>

## VehicleInventory (abstract)

-----  
List<Vehicle> (crude)

## Vehicle

-----  
VehicleType (enum), barcode, vehicleNumber, location,  
pickup date, pickupLocationhourlyRentle, dayRentle

## Reservations

-----  
parkingId, Vehicle, Member, fromDate, toDate, fromTime, toTime,

## AdditonalEquipment (abstract class)

-----  
EquipmentType (Enum), List<Equipment>

## Equipment

-----  
Name

## Services

## Bill

## Payment

**Q7. Can you please list down the design pattern used**

Ans.

**SingleTon** - ParkingManagmentSystem

**Factory Pattern** - Vehicle

**Notifcation** - Overserver Pattern



## **Q9. Start implementing class for following pattern**

Ans. First, I explained to him what a singleton is and why it's needed. A singleton is used to ensure only one instance is created, and there are generally three common ways to achieve it:

- 1. Eager Loading**
- 2. Lazy Loading**
- 3. Double check for multithreading**

```
public class ParkingManagmentSystem {  
    private static ParkingManagmentSystem parkingManagmentSystem = null;  
    private ParkingManagmentSystem() {  
    }  
    public static ParkingManagmentSystem getInstance() {  
        if (parkingManagmentSystem == null) {  
            synchronized (ParkingManagmentSystem.class) {  
                if (parkingManagmentSystem == null) {  
                    parkingManagmentSystem = new ParkingManagmentSystem();  
                }  
            }  
        }  
        return parkingManagmentSystem;  
    }  
}
```

## Feedback given by Saran Balaji C:

Good but missed a few requirements. Classes and properties:  
Covered most points but missed some of them. Implementation:  
Good understanding in OOPS. Some more practice is required on  
Design patterns. Time management needs to be improved.

**Final Result -> Successfully Cleared**

## How do I prepare for my interview:

I spent two days reviewing my notes on lld2 and lld3 to refresh my knowledge.

I also worked on various schema designs, including those for a **car rental system**, **parking lot management**, **clickbuzz**, and a **logger**

@GeekySanjay

# Thank You



## Contact Me

If you have any questions or suggestions regarding the video,  
please feel free to reach out to me

on WhatsApp: **Sanjay Yadav Phone: 8310206130**

<https://www.linkedin.com/in/yadav-sanjay> | <https://www.youtube.com/@GeekySanjay>

