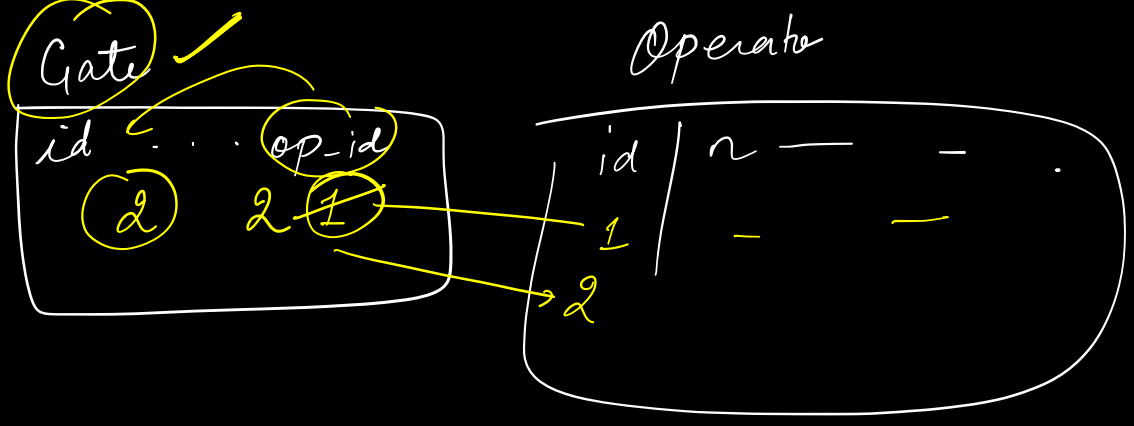
Today's Agenda :-

1) Problem statement
2) Gather requirements
3) Create the class diagram
4) Schema Design.

Shift by 1 forward.

16th April
||
Break
18th April

Yes



Problem Statement → Design Parking Lot.

Overview → Know / Don't Know. ✓

Multi level parking lot.

Persist the data ? → In-memory db.

↳ Disk.

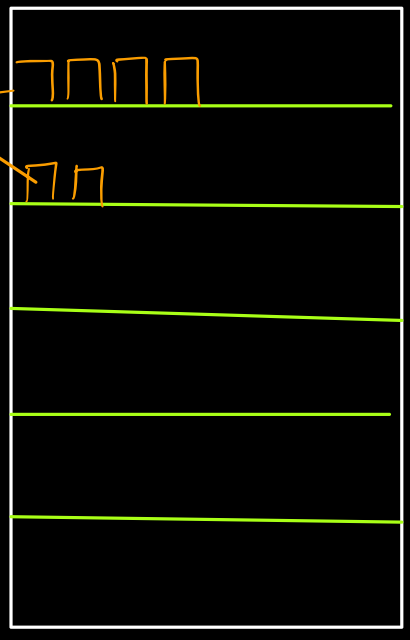
⇩

Storing the data in RAM.

Requirement Gathering :→

↳ core functionalities of the system.

1. Multiple Parking floors.
2. Different parking spots for different types of vehicles.
3. A token is generated at the entry gate.
4. Payment should be made at the time of exit.
5. A parking spot is assigned at the time of entry.
6. Payment algorithm can be dynamic & our system should support multiple payment algorithms.



Parking Lot

↳ {Strategy design} pattern ⇒ Fee Calculation Algo

Festive season k time pr ya different time pr different charges apply kr skte h.. it's out of box requirement which is good for interview health

⑦ Payment can be done in either Cash or Online.

⑧ Online payments will be handled by a 3rd party payment gateways. } ⇒ {Adapter DP}.

Design pattern

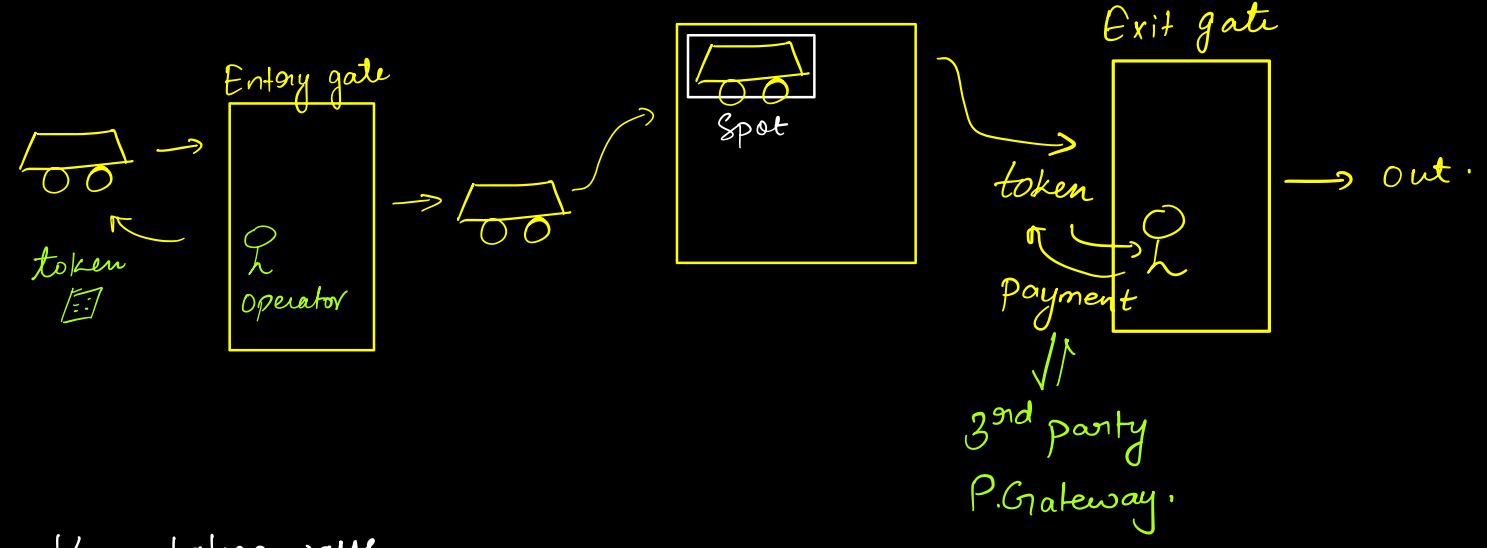
⑨ Multiple entry / exit gates.

⑩ Entry & exit gates will be different.

⑪ Assign a spot to the vehicle. ⇒ SpotAssignmentAlgo

⑫ Only a spot for exact type of vehicle can be assigned.

User-journey :-



Key-takeaways

1) Assign a parking spot to a new vehicle ? } 2Q's.

2) Generate the token

ParkingLotMgr
- id

Class Diagram

- name
- email
:

① **ParkingLot**

- id
- mgr
- List < ParkingFloor >
- List < Gate >
- Status
- SpotAssignmentAlgo
- FeeCalculationAlgo
- capacity

② **ParkingFloor**

- id
- number
- List < ParkingSpot >

③ **ParkingSpot**

- id (PK)
- number
- status
- floor
- VehicleType ← SUV, Sedan, 2-wheeler

EntryGate

gateNo
operator
Status

ExitGate

gateNo
operator
Status

④ **Gate** — Concrete

gateNo
curr. operator
Status
GateType ← ENTRY, EXIT

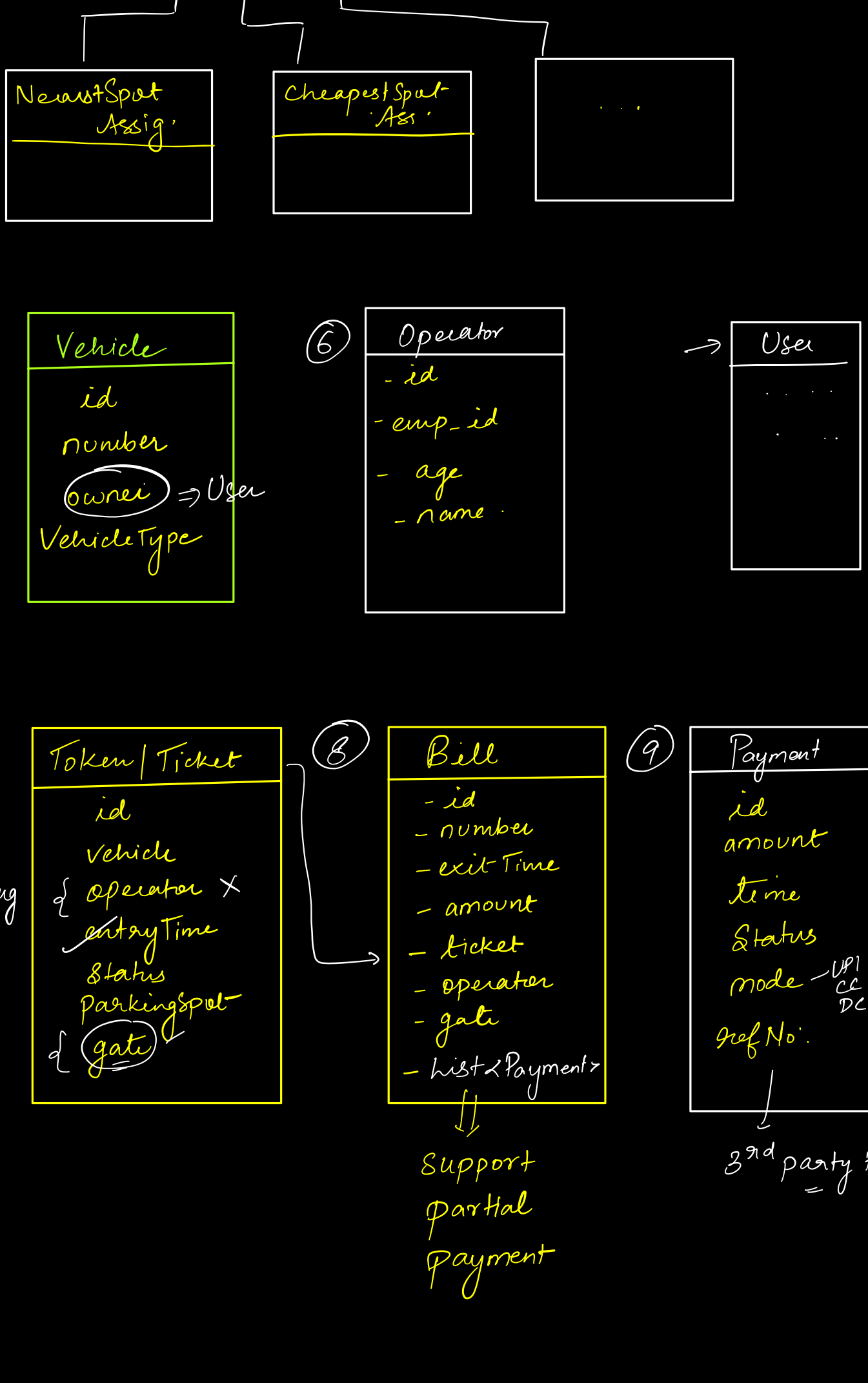
<< SpotAssignmentAlgo >>

assignSpot ( ) ;

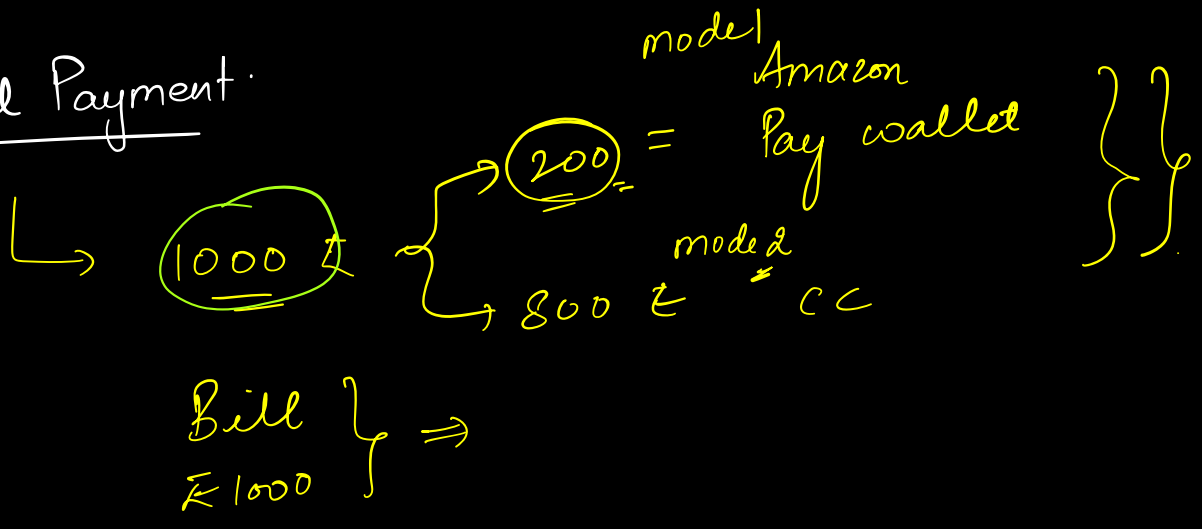
↑ ↑ ↑

Configs

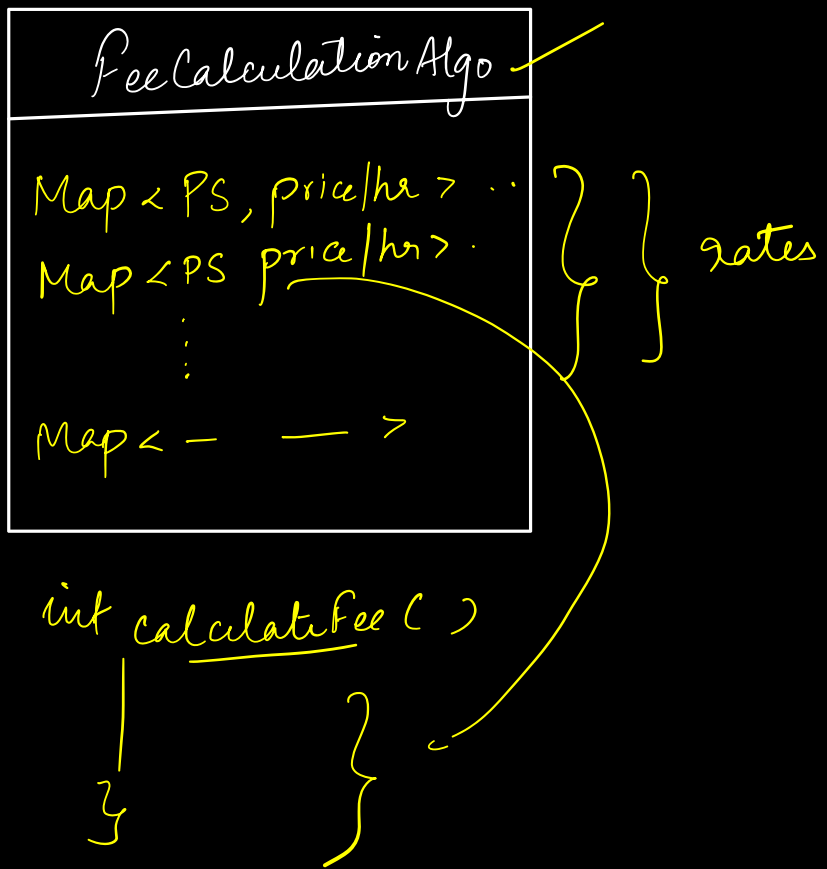
later


NearestSpot Assig.
CheapestSpot Ass.
...
5
Vehicle
id
number
owner ⇒ User
VehicleType
6
Operator
- id
- emp_id
- age
- name
User
7
Token | Ticket
id
Vehicle
{ operator ×
entryTime
Status
ParkingSpot
{ gate
Unang
8
Bill
- id
- number
- exitTime
- amount
- ticket
- operator
- gate
- List<Payment>
Support
Partial
Payment
9
Payment
id
amount
time
Status
mode — UPI CC DC
refNo.
3rd party PG.

Partial Payment.



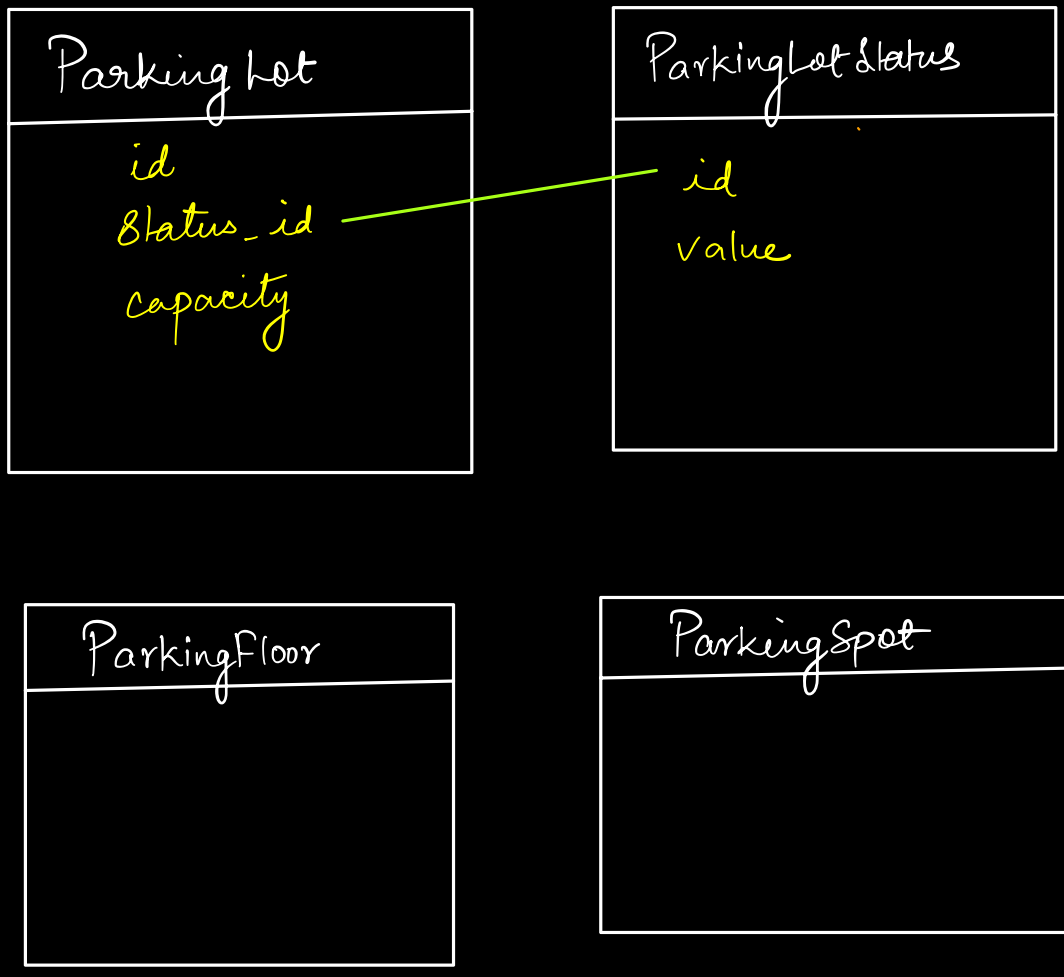
Bill ⇒
₹1000

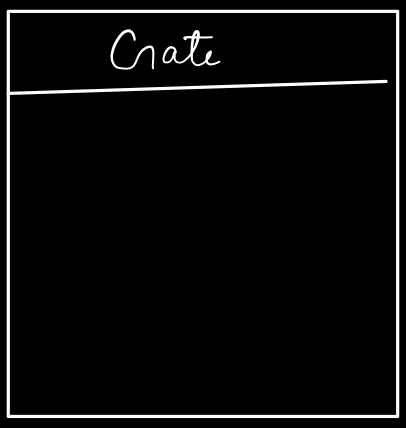


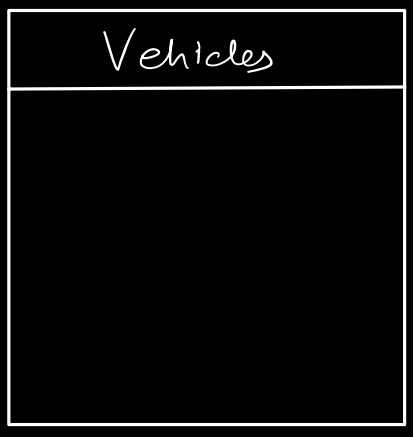
Schema Design

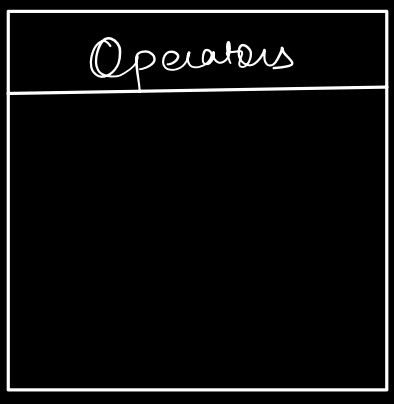
↳ Based on the class diagram, please come up with Schema Design.

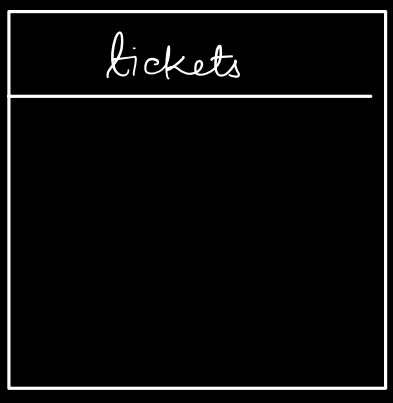
Schema Design of ParkingLot :-

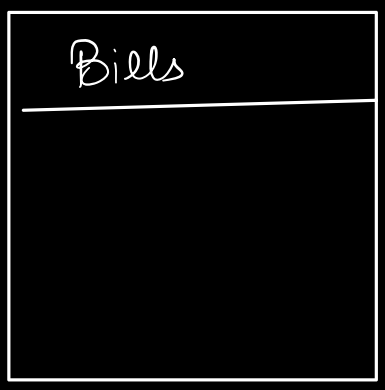












↳ Complete the above Schema Design } 2 days Sat - Sun

① HW 1st priority

↳ 10 - 15 min

(2) Complete the schema design for the assⁿ problem also

Monday

Break for 8 Min

↳ Code models.