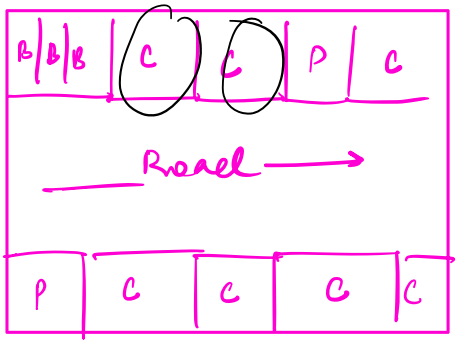
Algorithm
Problem
Challenge

Agenda

Design a { Parking Lot } → Amazon, Microsoft



Im

Rippling?

Key value store

Transaction Comm Rollb

Return PORT

Npoints

get / boolean

{ Entities focus ?
↓
Management of parking lot

id
↓
occupi

get( )

free(id) →

{ users

{ → Command Line / API design +

DB ?

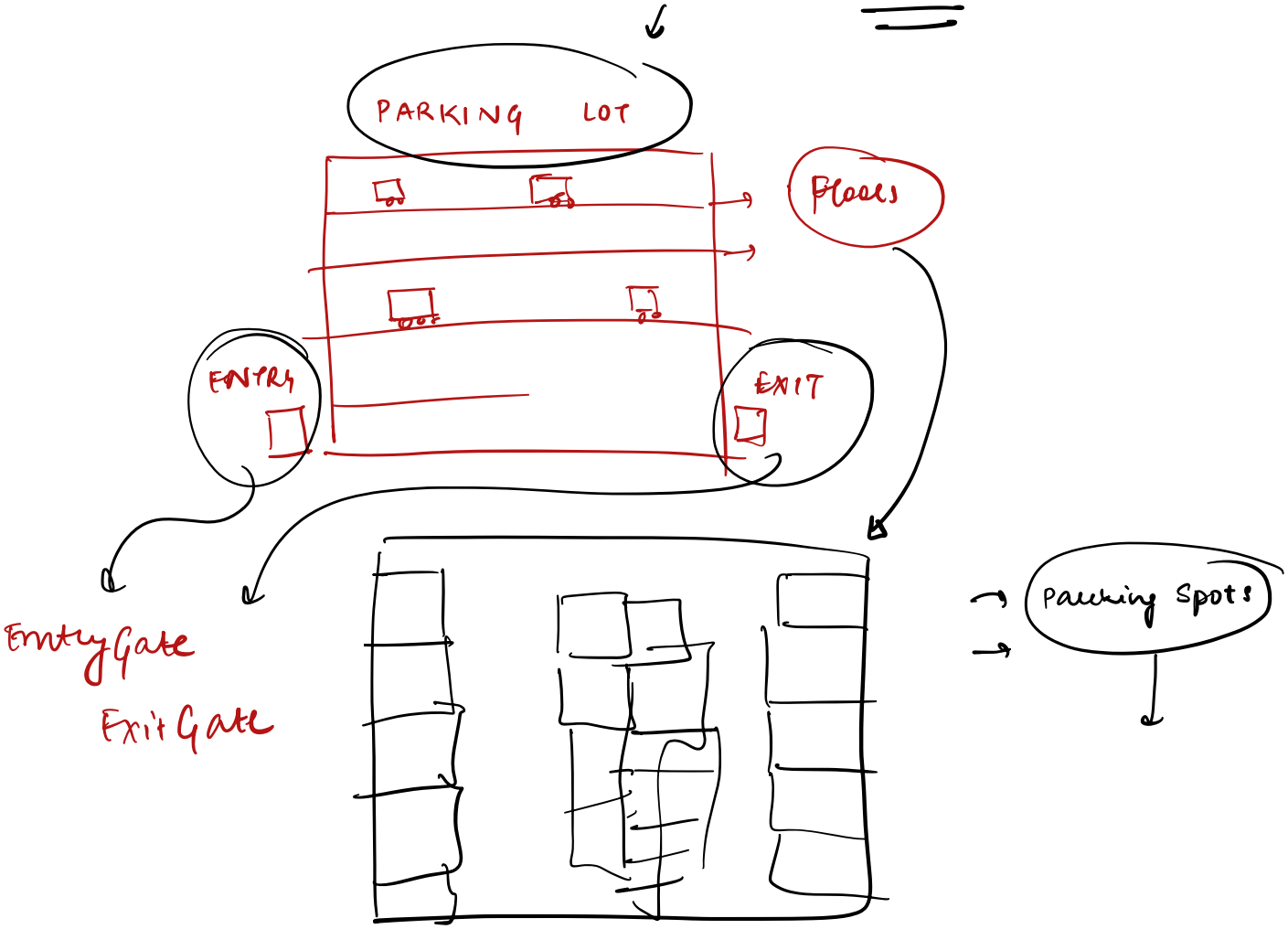
Gather Requirements :

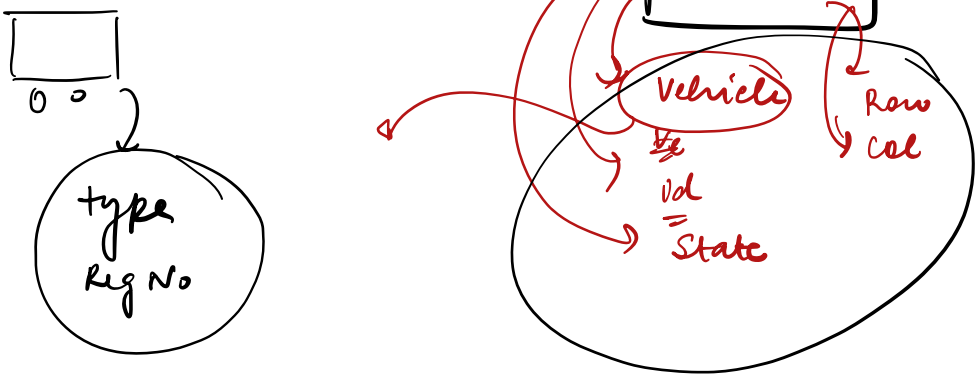
1) Size? Floors (Multi level or single level)
   ↳ Capacity ?
2) Entry and Exit gates ? (Multiple)
3) Entry gate => Ticket is generate and a slot is allocated with time noted
   => Should check availability
4) Parking floors → Many parking spots
   - Compact, SUV, Sedan → CAR
   - Motorcycle → BIKE
   - Bus / Truck → BUS
5) Exit Gates => charged accorded pay the amount
   automatically slot is freed

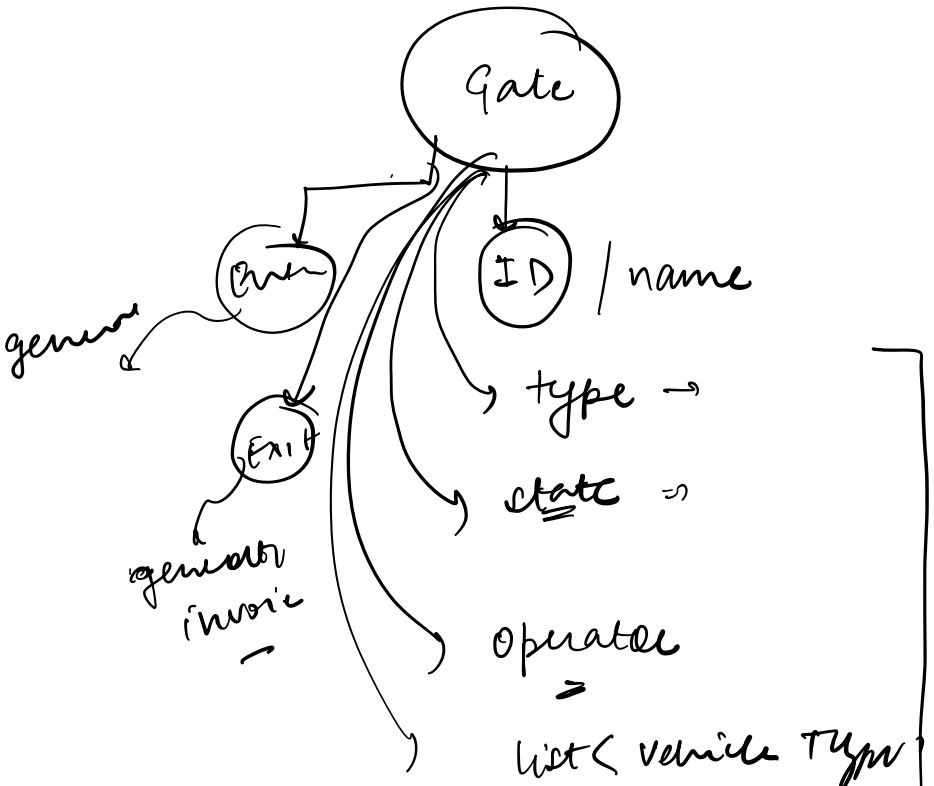
Admin => CRUD
- Add / Edit floors
- spots
- Ticket Pricing

Operator =>
- confirming payment
- => Generate Ticked (no. time)

TOP DOWN


PARKING LOT
Floors
ENTRY
EXIT
EntryGate
ExitGate
Parking Spots

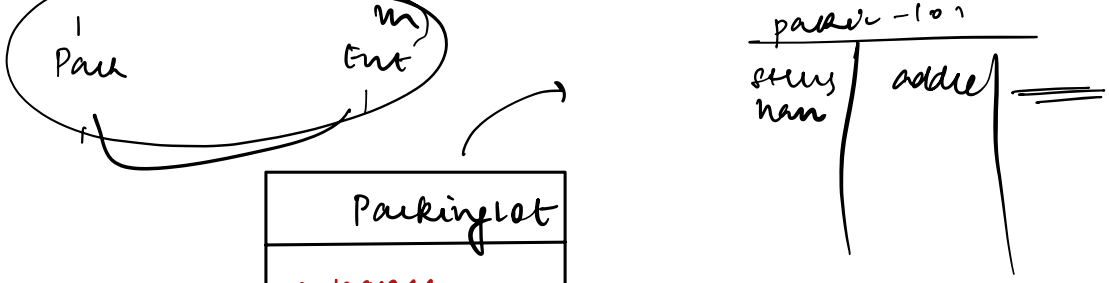

Vehicle
Row
Col
Id
State
type
RegNo

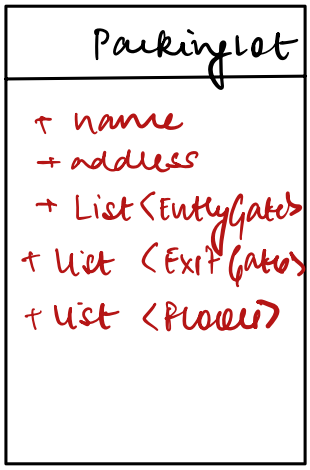


Ticket
I
- id
- vehicle
- QR code
- time entry
- entry gate
- status
- parking spot
- ~~parking floor~~

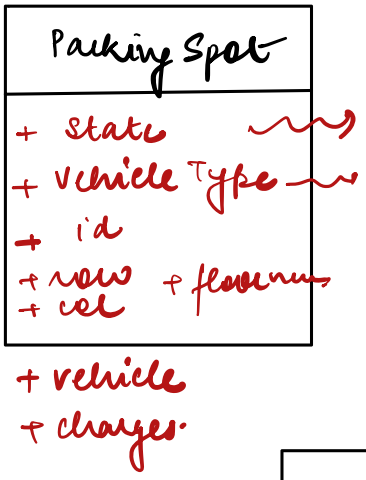
Bill
=
- amount
- exit time
- ~~entry ti~~
- details
- id
- payment type
- ~~vehicle~~
- ~~ticket id~~
- ~~slot~~

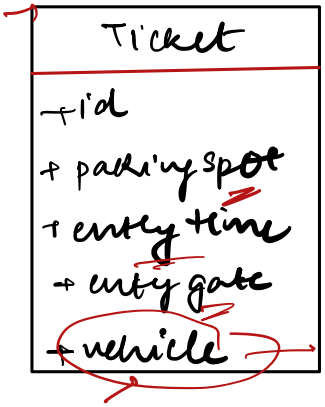
ticket


1
Park
m
Ent

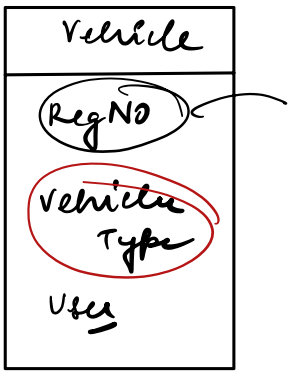

ParkingLot
+ name
+ address
+ List<EntryGate>
+ List<ExitGate>
+ List<Floor>

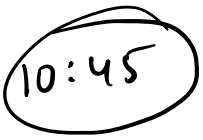

Parking Floor
+ floor number
+ List<ParkingSpot>
+ state
+ rows
+ cols

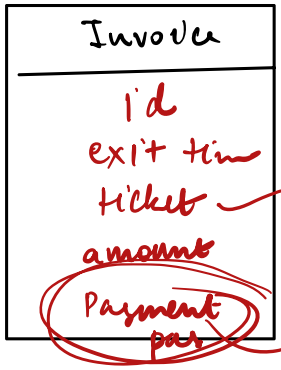

Parking Spot
+ state
+ vehicle Type
+ id
+ row
+ col
+ floornum
+ vehicle
+ charges

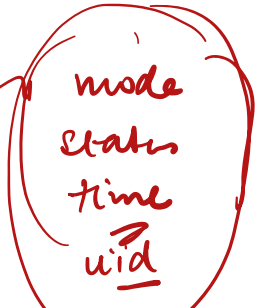

Ticket
+ id
+ parkingspot
+ entry time
+ entry gate
+ vehicle

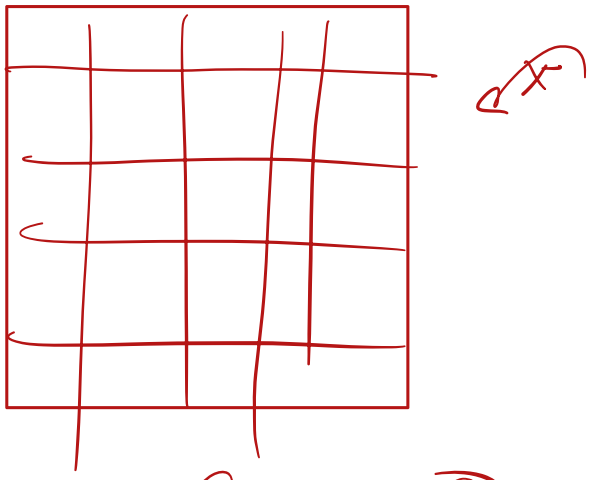

Gate
id
type
attendant

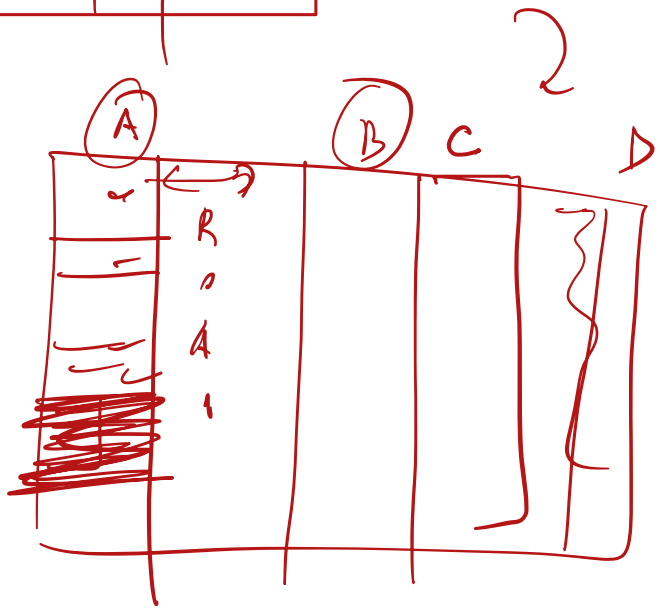

Vehicle
RegNO
vehicle Type
User


10:45


Invoice
id
exit time
ticket
amount
Payment pay


mode
status
time
uid




A
B
C
D
R
A
1