

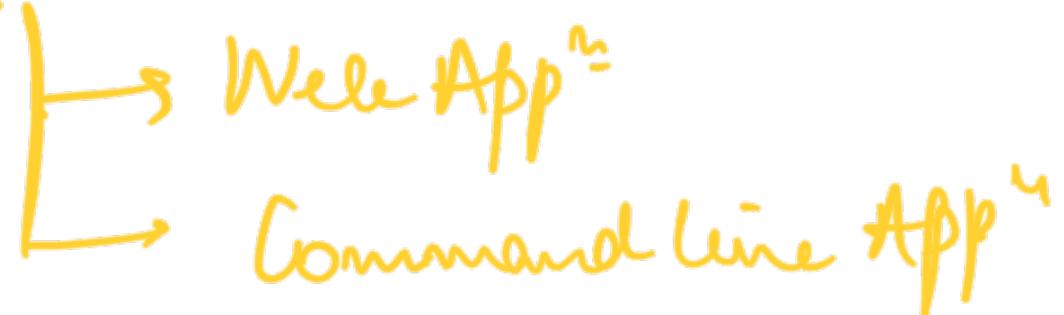
# DESIGN A PARKING LOT

(Will be starting at  
9:10 PM)

- Pen (0)
- TTT (1)
- SBL (2)
- Parking Lot (3)

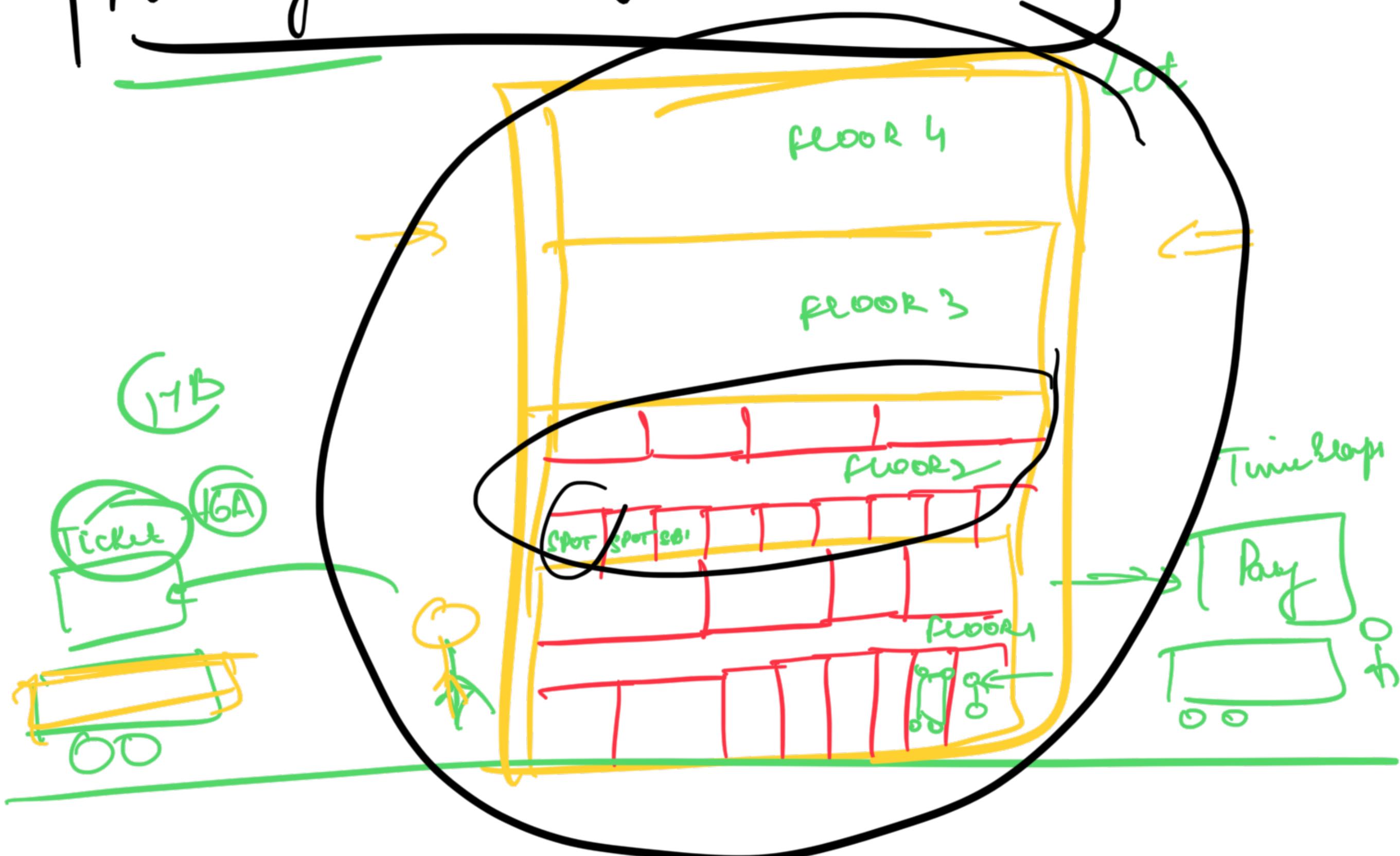


⇒ Designing Real World  
Systems



⇒ Storing data will be required

# Parking lot Management System



1st Hour → 100  
2nd Hour → 200  
every Hour → 300

5 hrs

100 + 200 + 300

1200'

Design a Toll Tax

No. in a Parking lot

no. type

# Design

## ① Overview

②

→ Clarify whether to design ~~existing / management~~

⇒ ③

Get a general idea of behaviour of the system

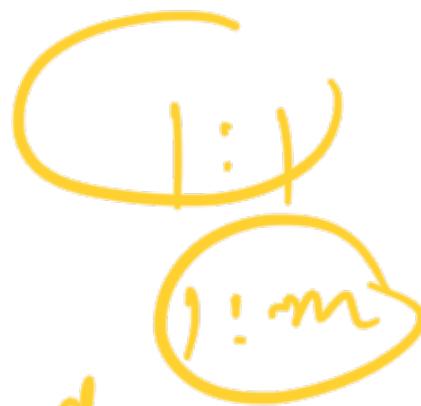
Design Maintenance

Understanding  
of software system

②

## Gathering / Clarifying Requirements

- Parking lot has Multiple floors
- Multiple entry and exit gates
- Each floor can have multiple parking spots
- At entry a ticket is issued and at exit payment has to be made.
  - o ... : calculated based on time



→ The cost is constant

since parking and type of vehicle but there  
can be other ways as well in future

→ Each Parking Spot is for a particular

Vehicle Type

- Small
- Large
- Medium



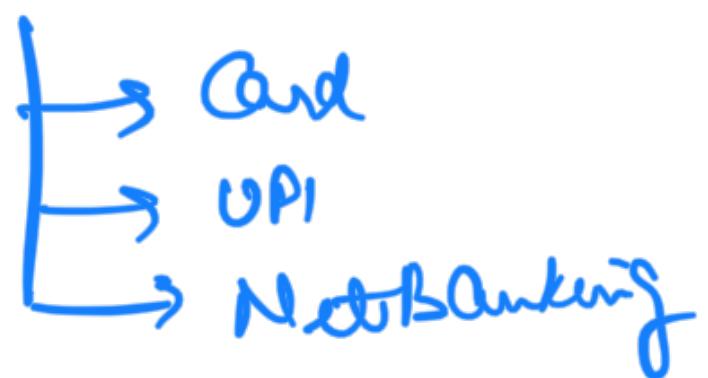
→ When a ticket is issued, a spot is

assigned →

→ A vehicle can only take spot for ~~use~~ <sup>use</sup>

- ⇒ At the ~~entry~~ gate there is a display board that shows count of available spots for each vehicle type.
- ⇒ A ticket can be paid either via cash or

Online -



- - - - - → exit gate. Bill is generated

⇒ A person goes to ~~the~~  
Bill is paid via cash / Dmt

→ A spot is also there for electric vehicles.

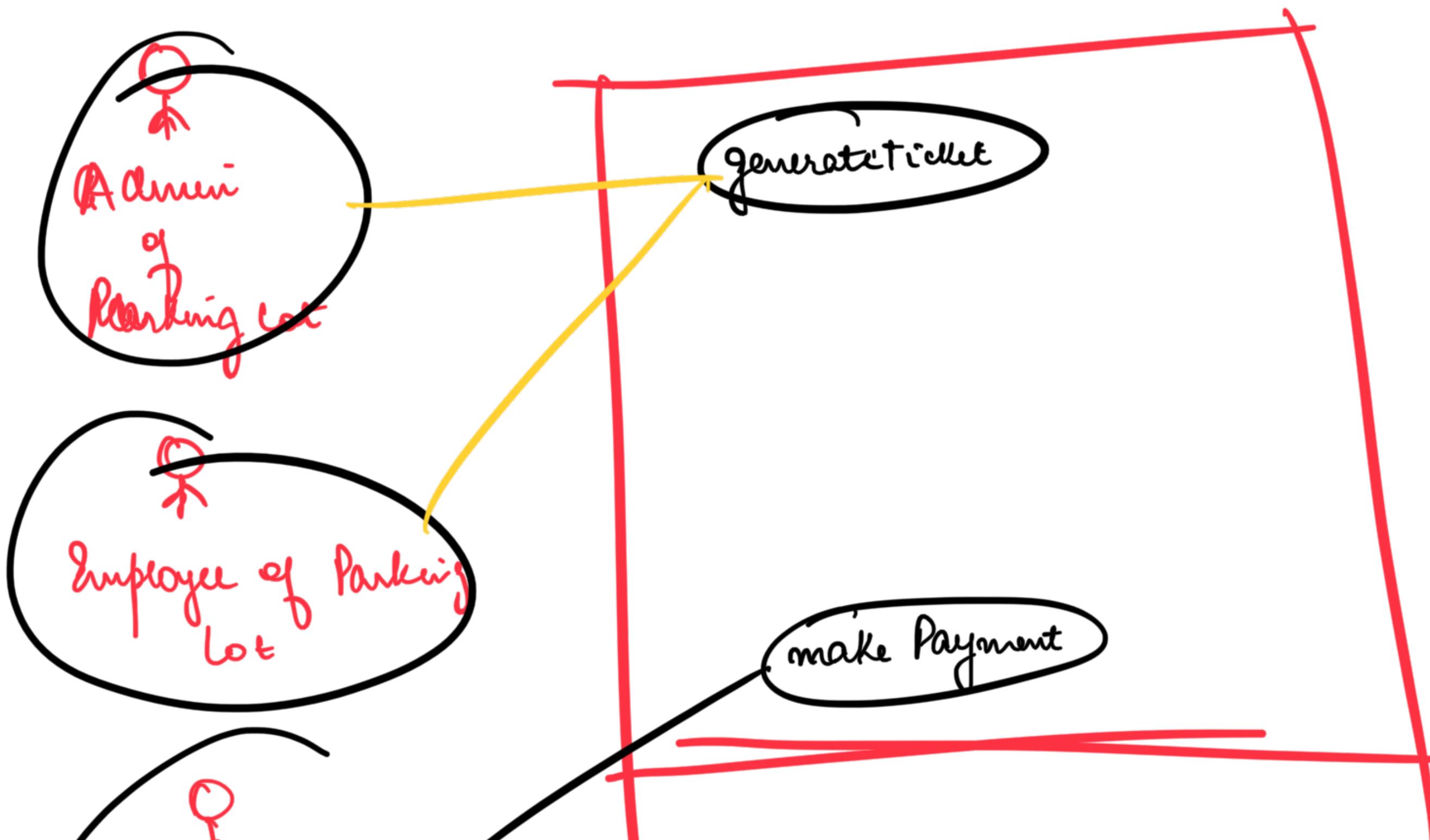
Cost will be calculated based on  
time + amount of electricity used during  
that time

15 Min

7:30

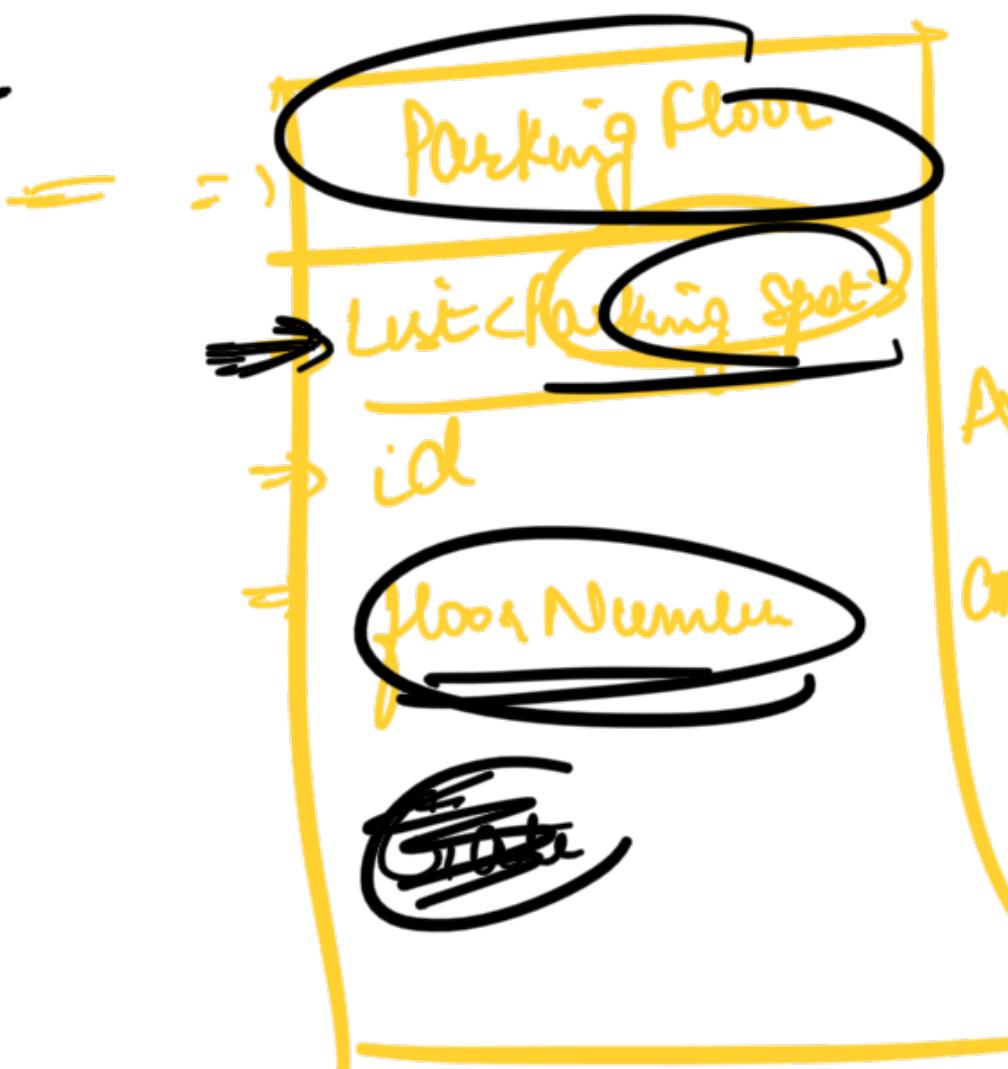
# Use Case Diagram

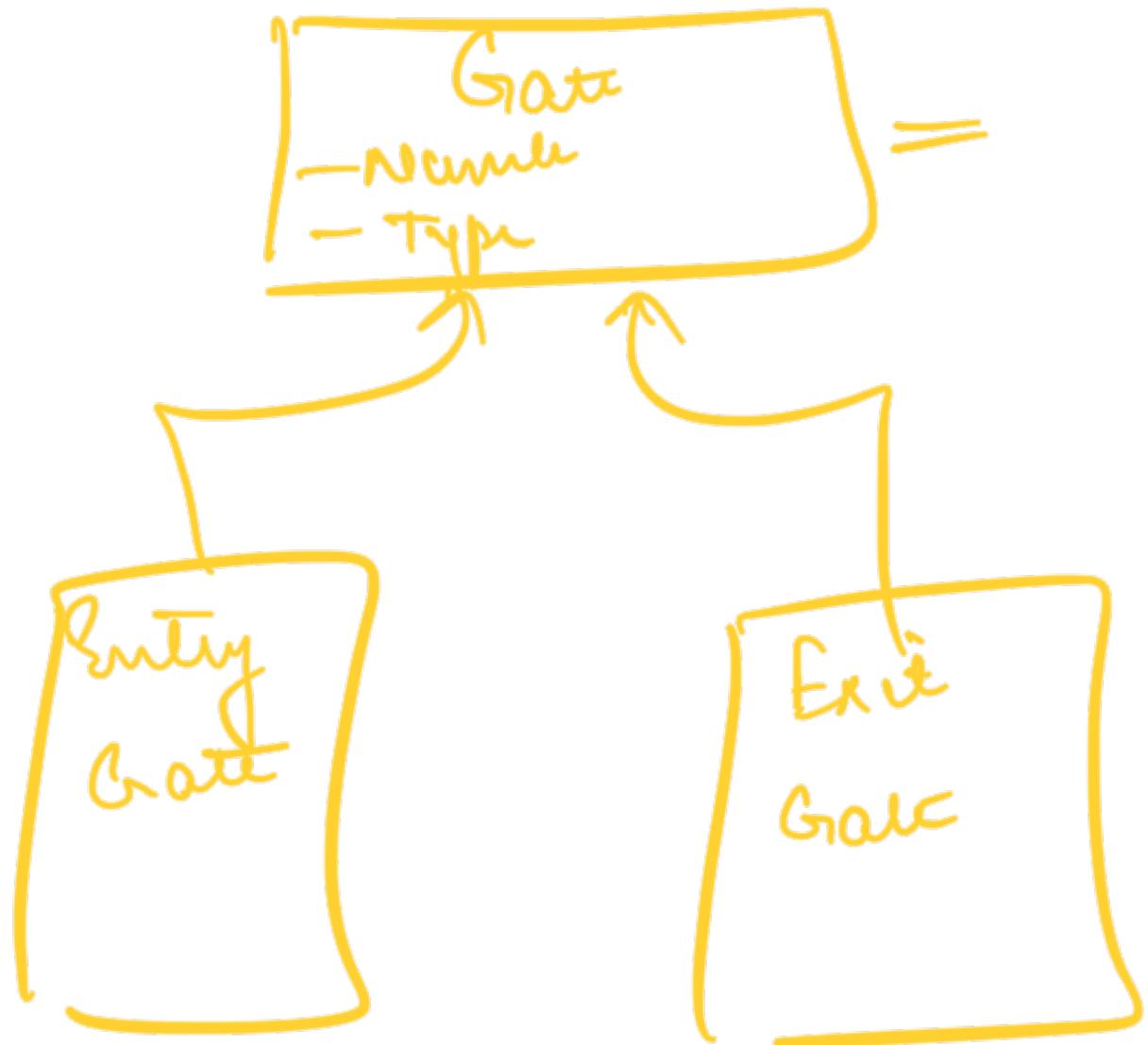
7:45



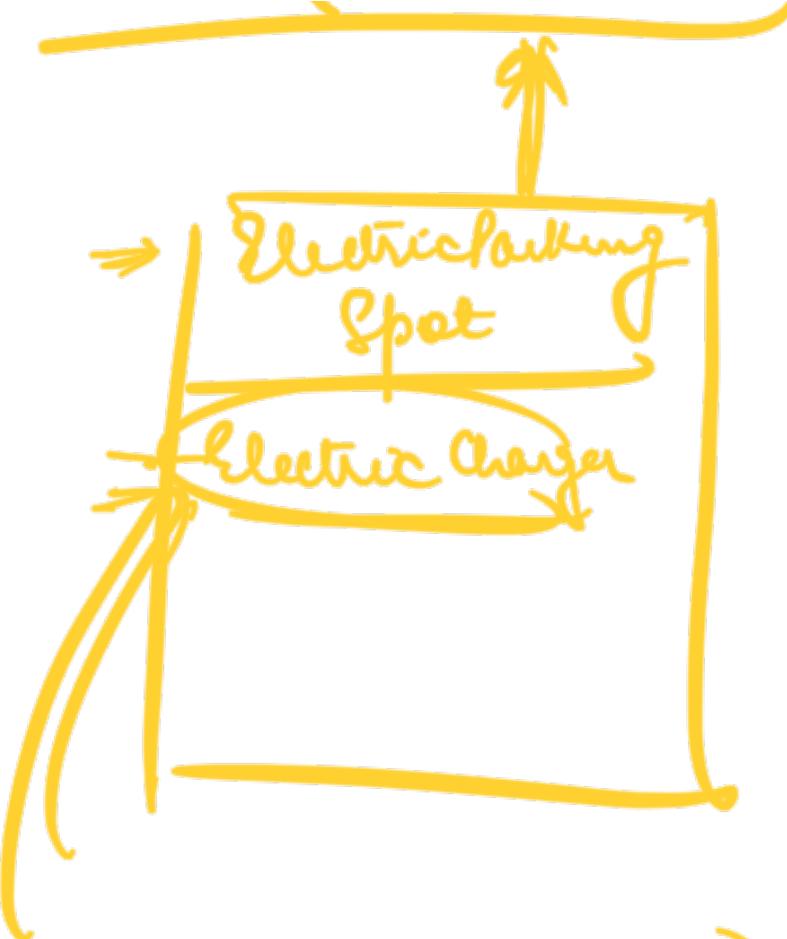
## Normal Users

# Class Diagram



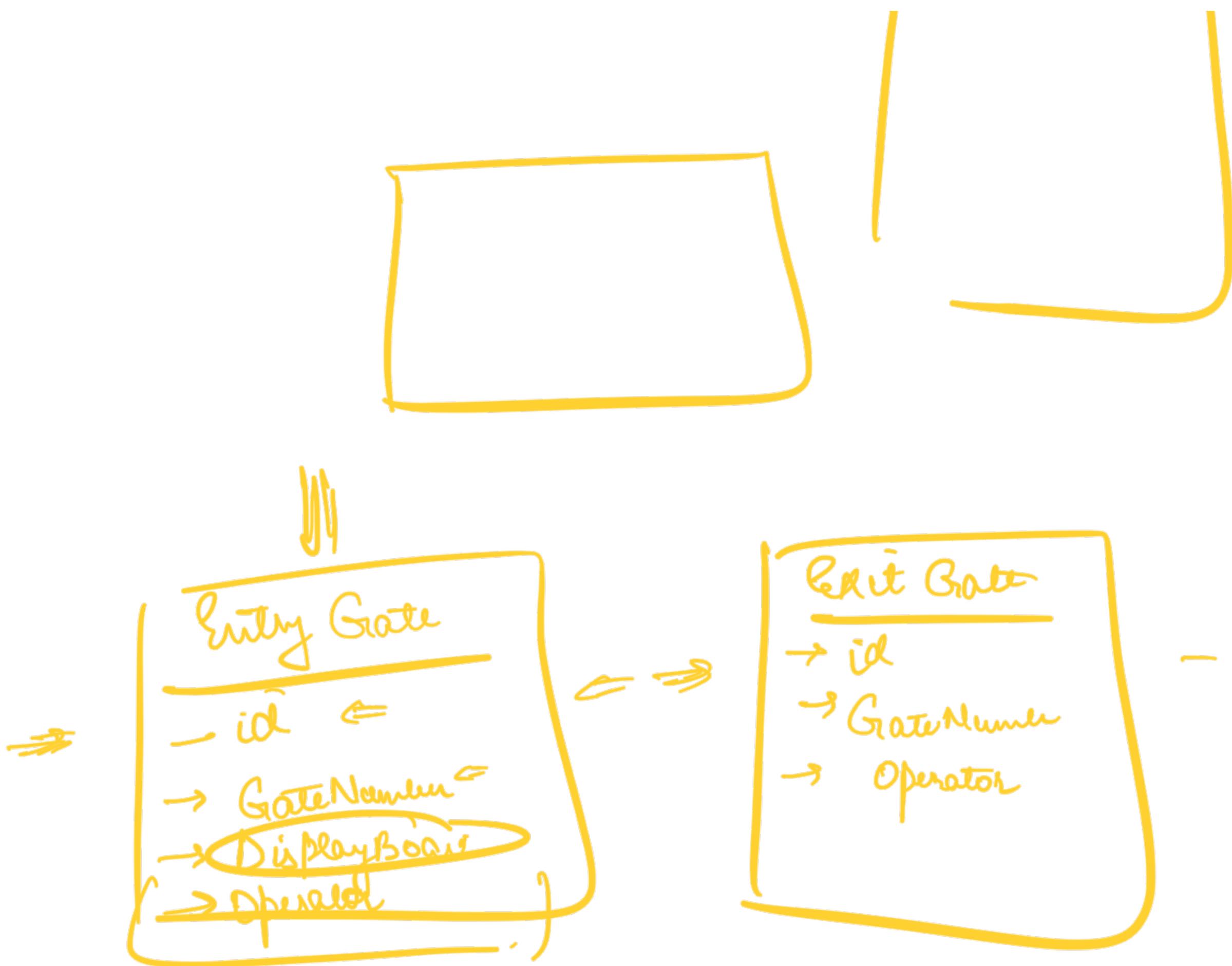


①

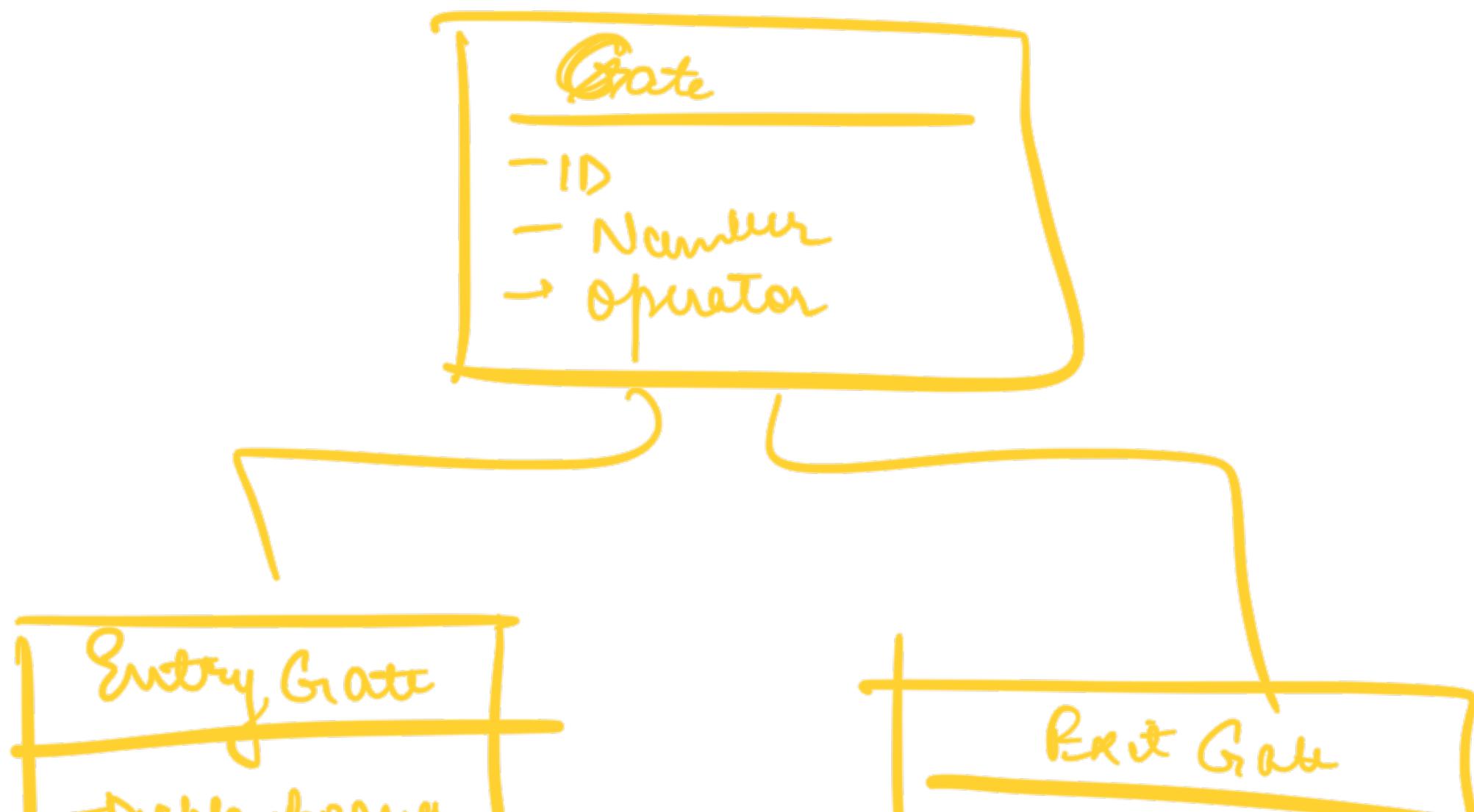


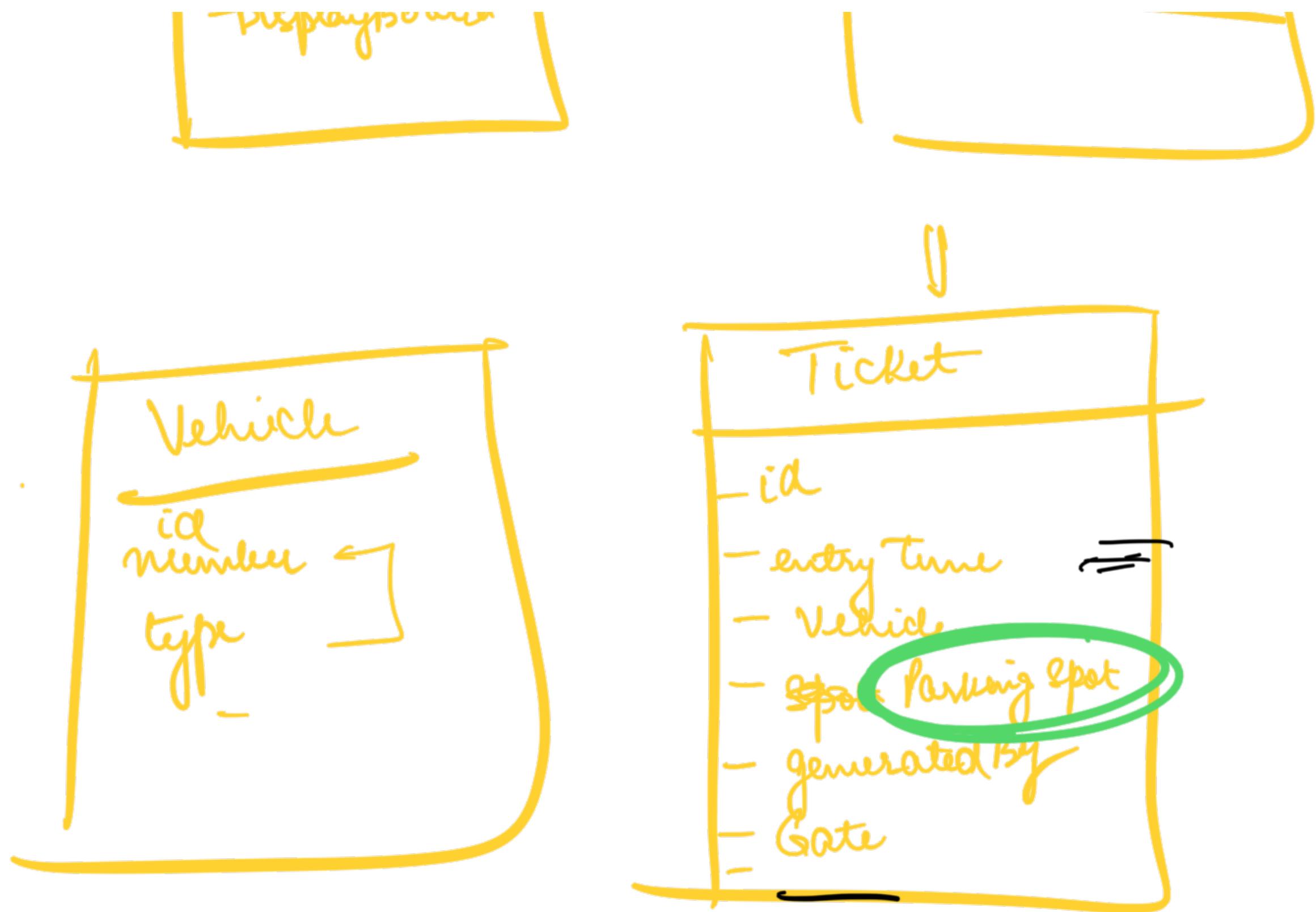
how much electricity  
was used in a time  
from





Whenever we will be building web app<sup>n</sup>,  
Our entities will Only have attributes  
No method other than get / set



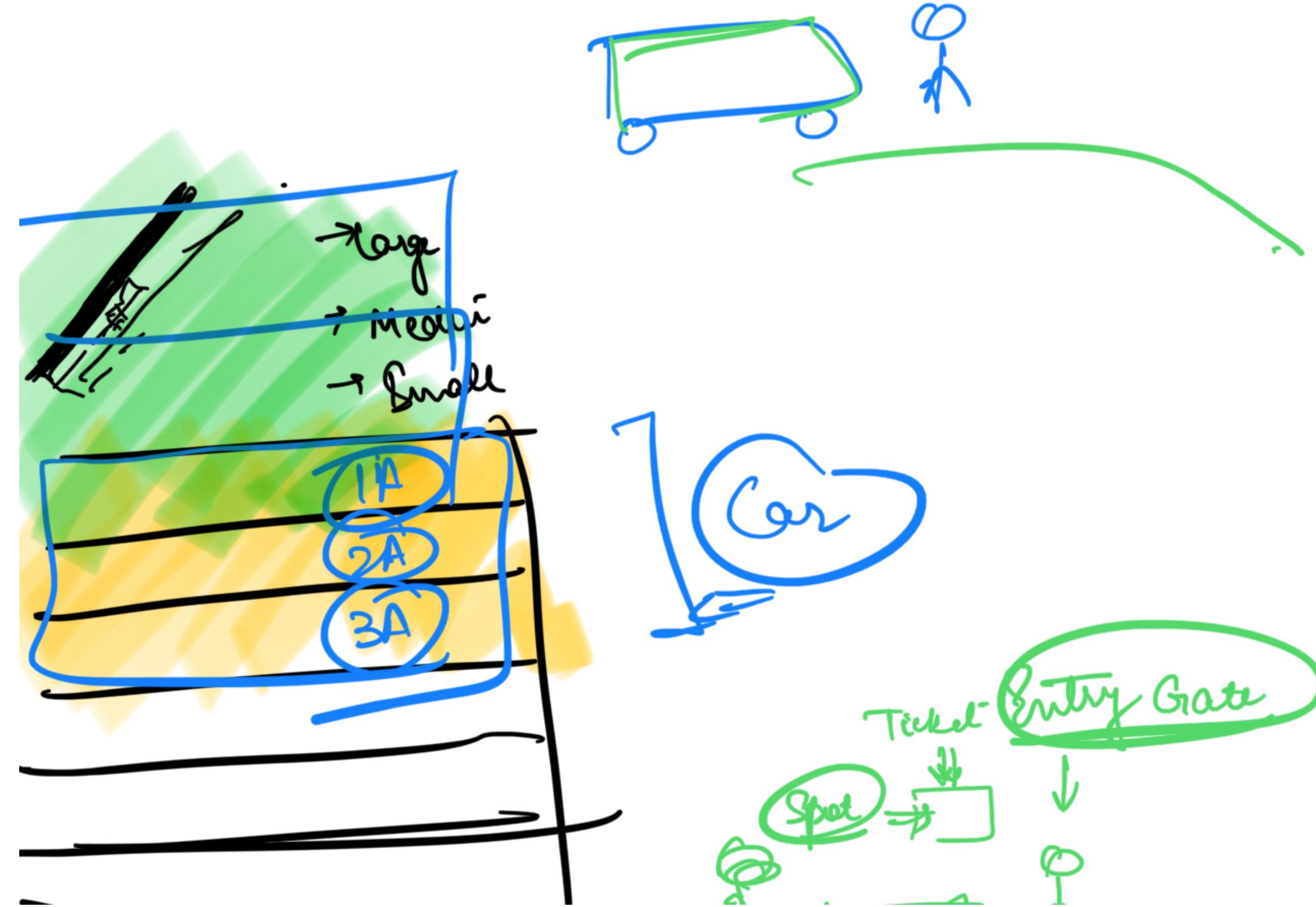


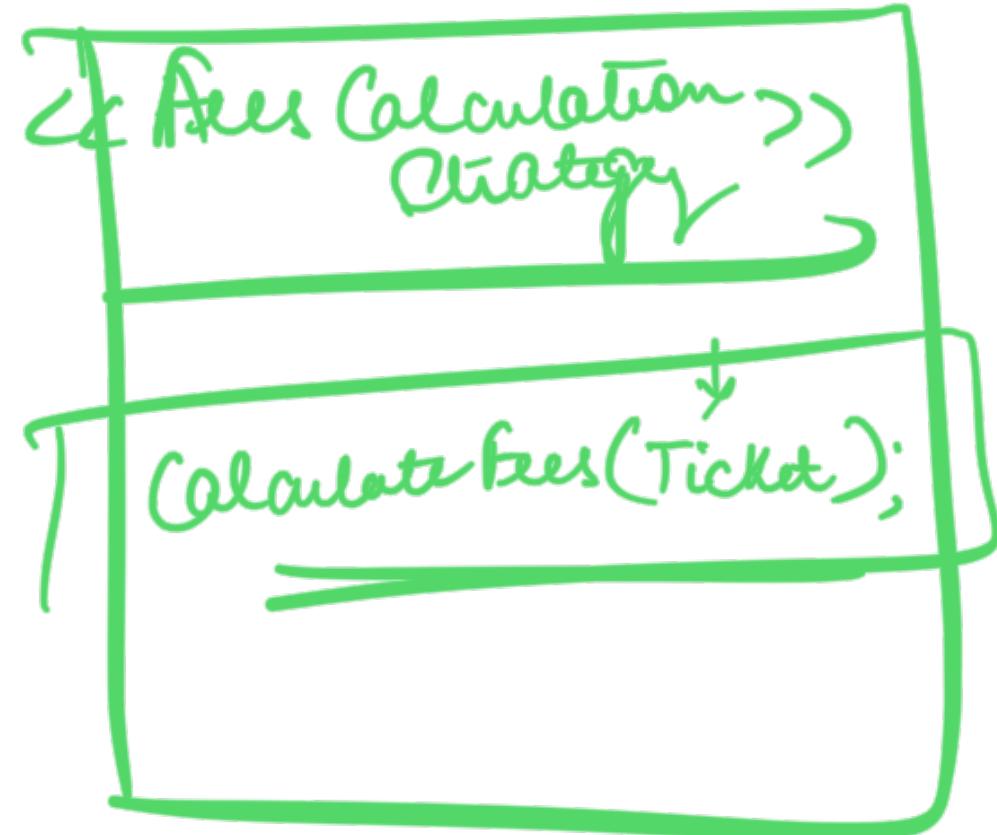
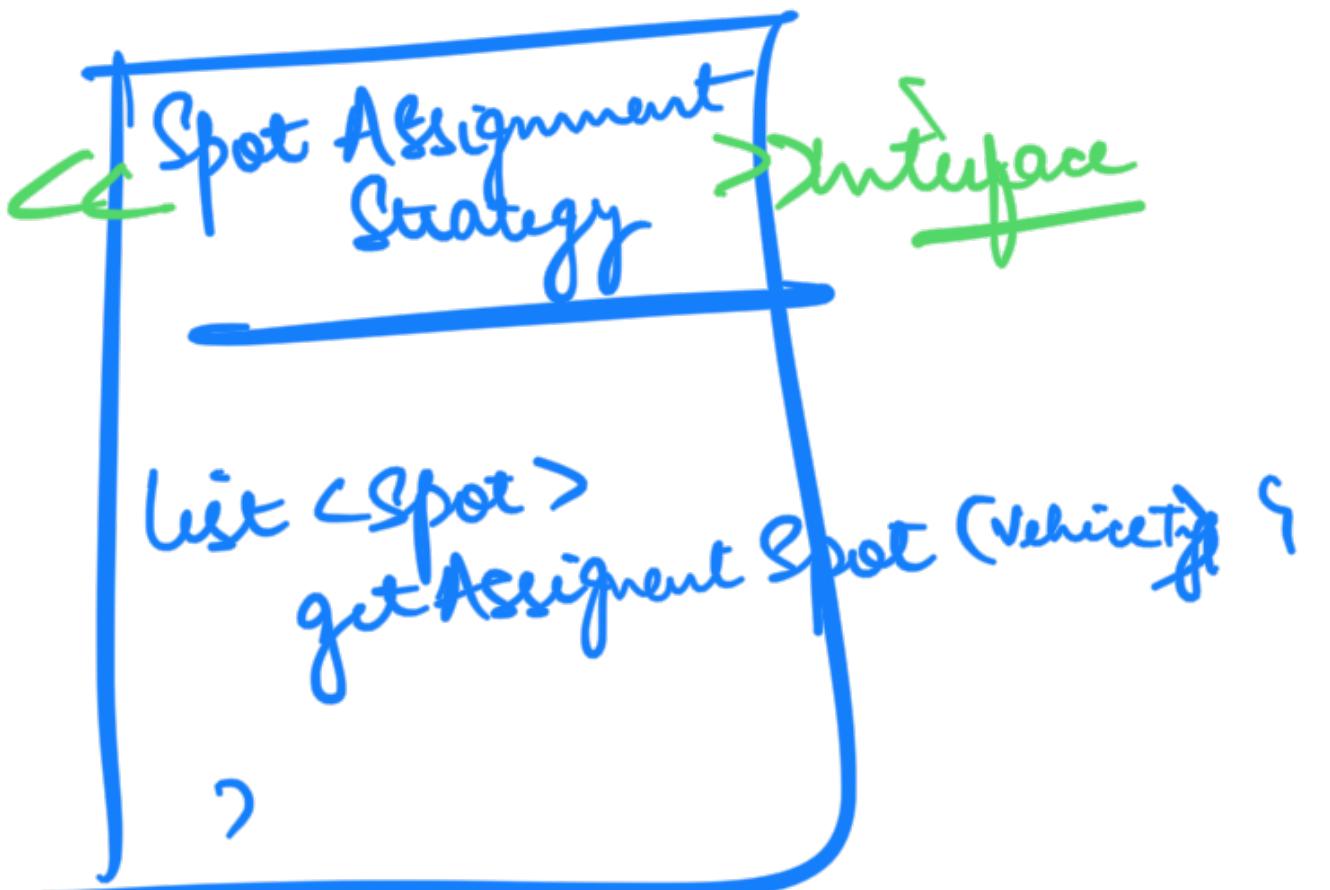


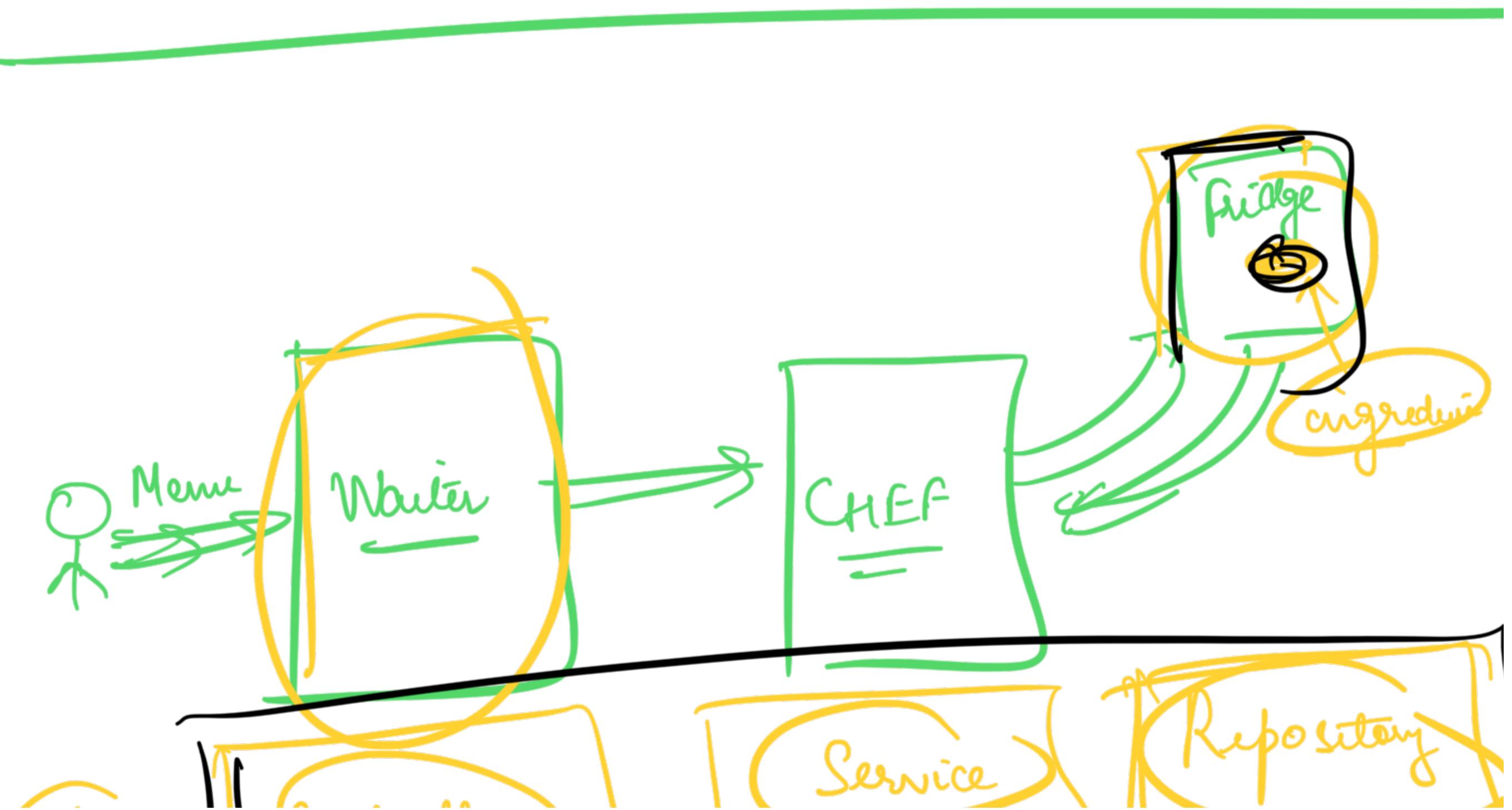
↑ In real app<sup>m</sup>

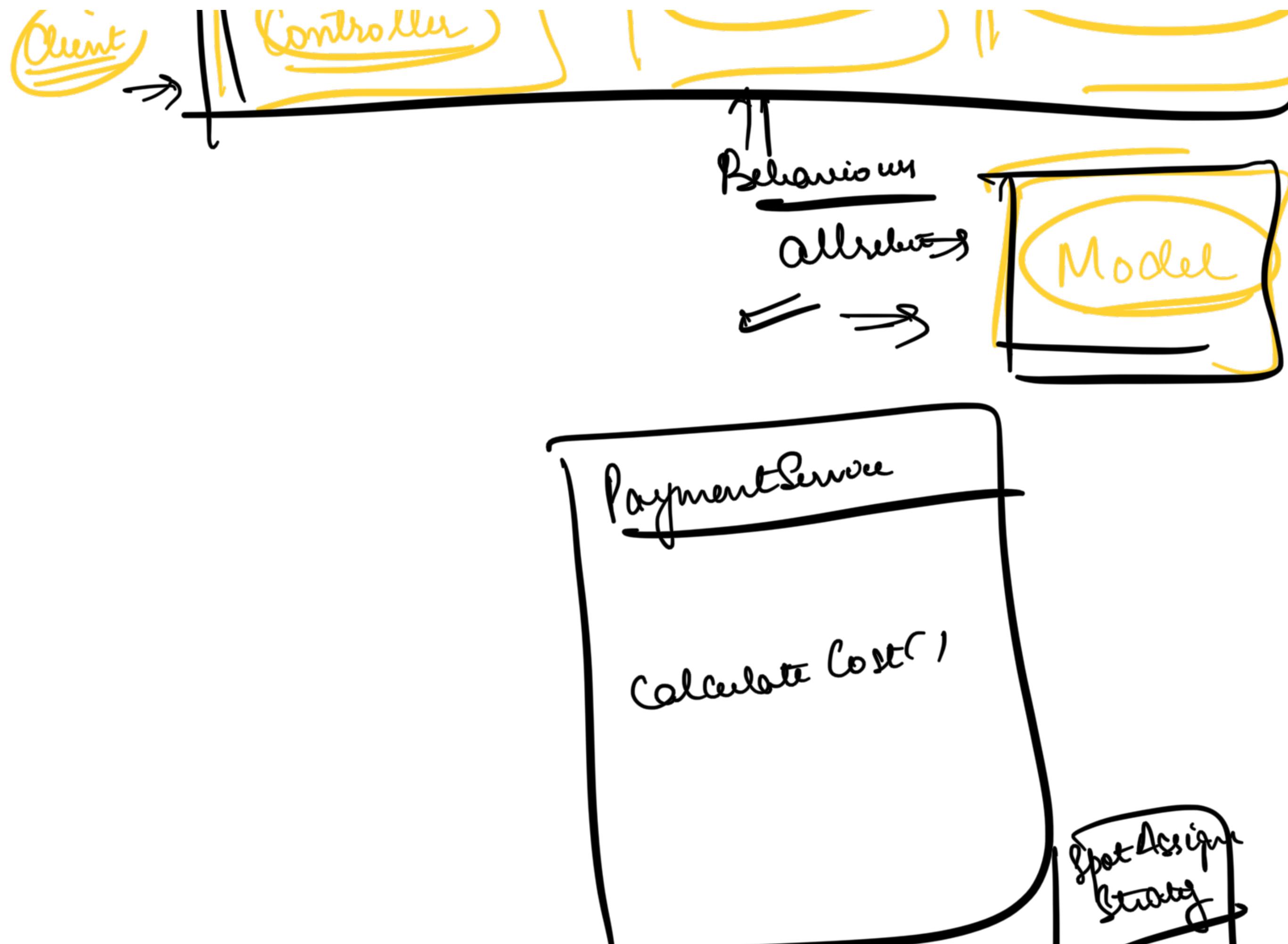
- attributes

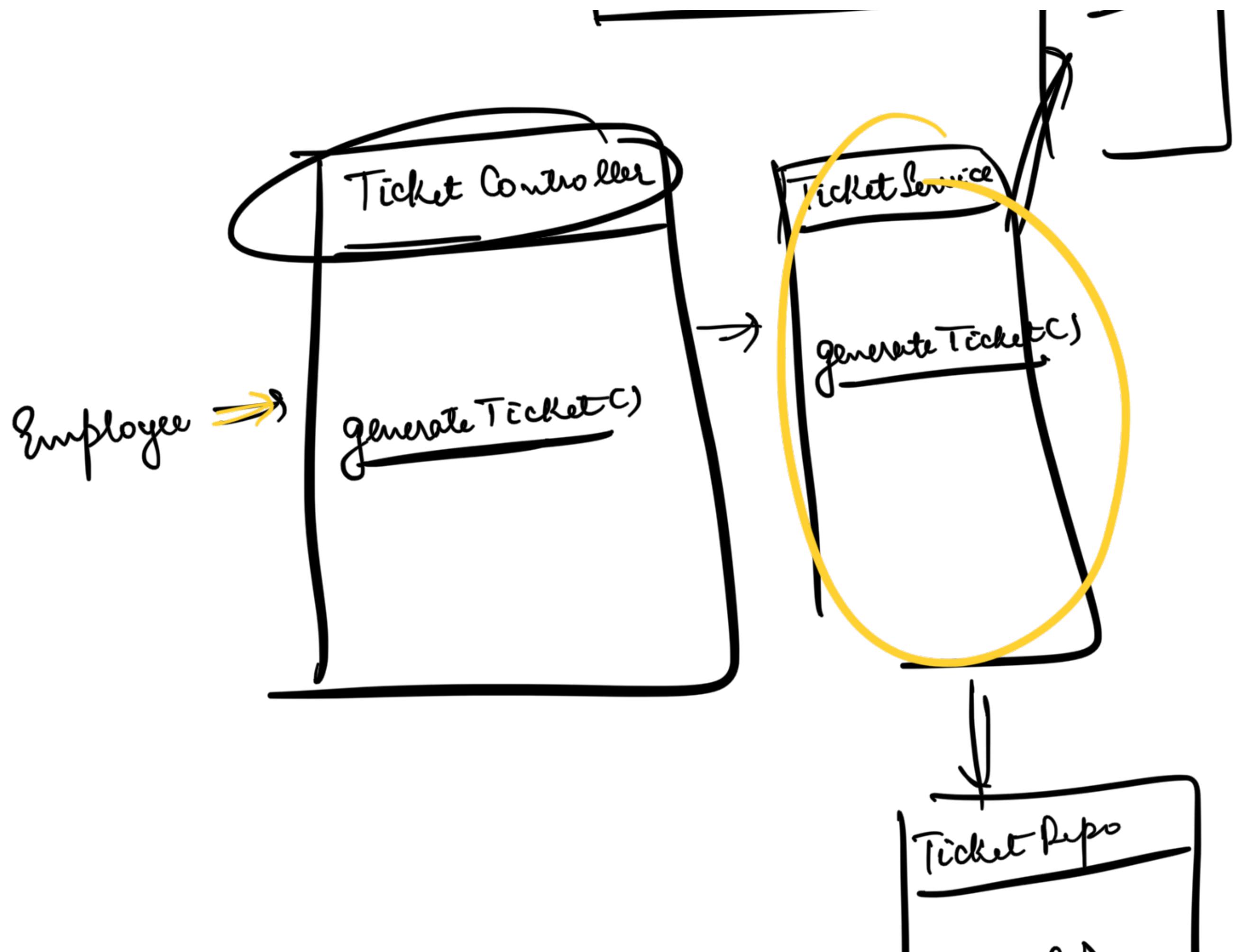
- a) Models will be just arrows
- b) No behaviour in model











Save ()

⇒ Creating a class diagram of parking lot  
Real World App

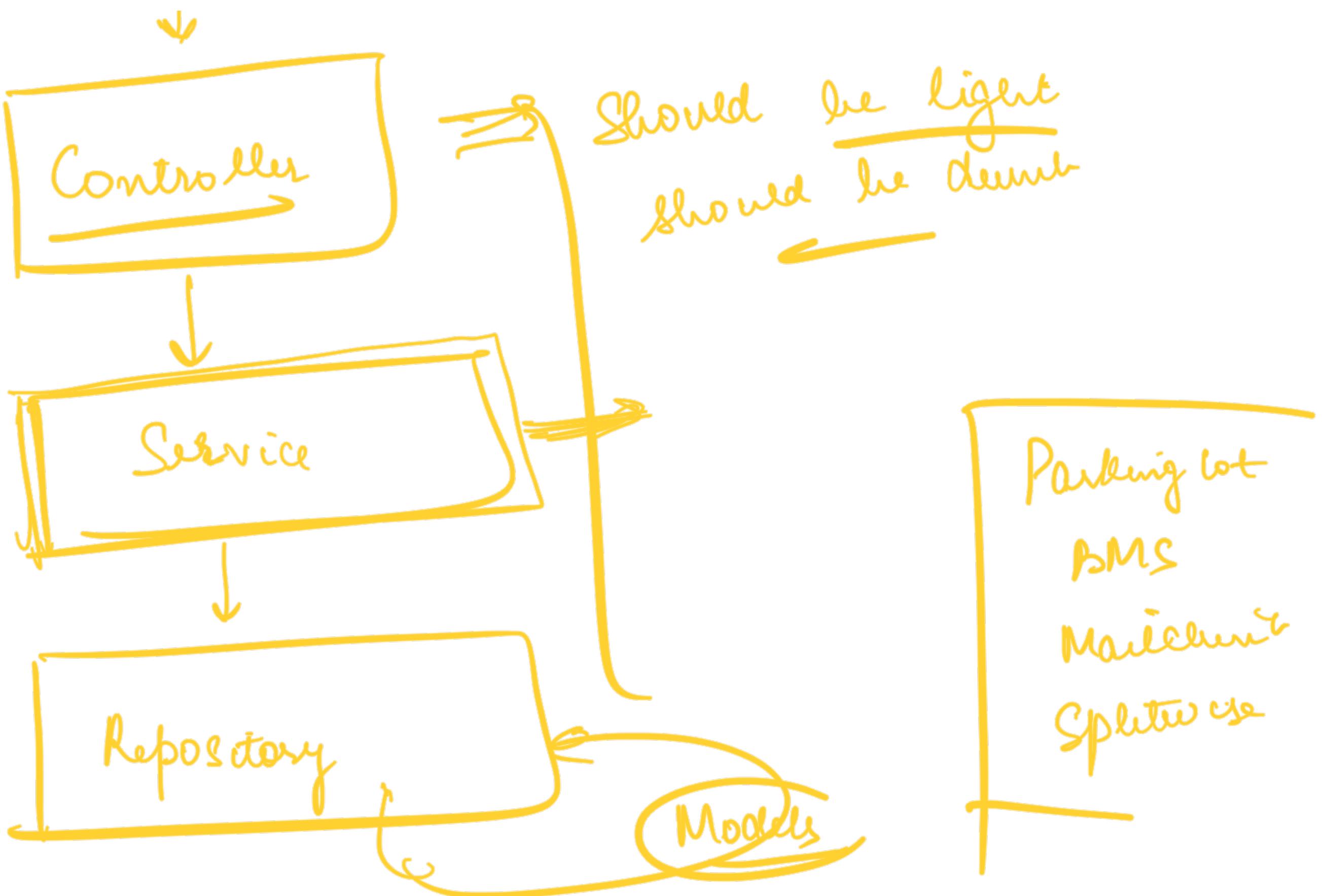
- Models should only have attributes that rep their property structure
- Models will not have any behaviours

PSR ⇒ To store the data

- Codebase is divided into 4 layers

Client

|



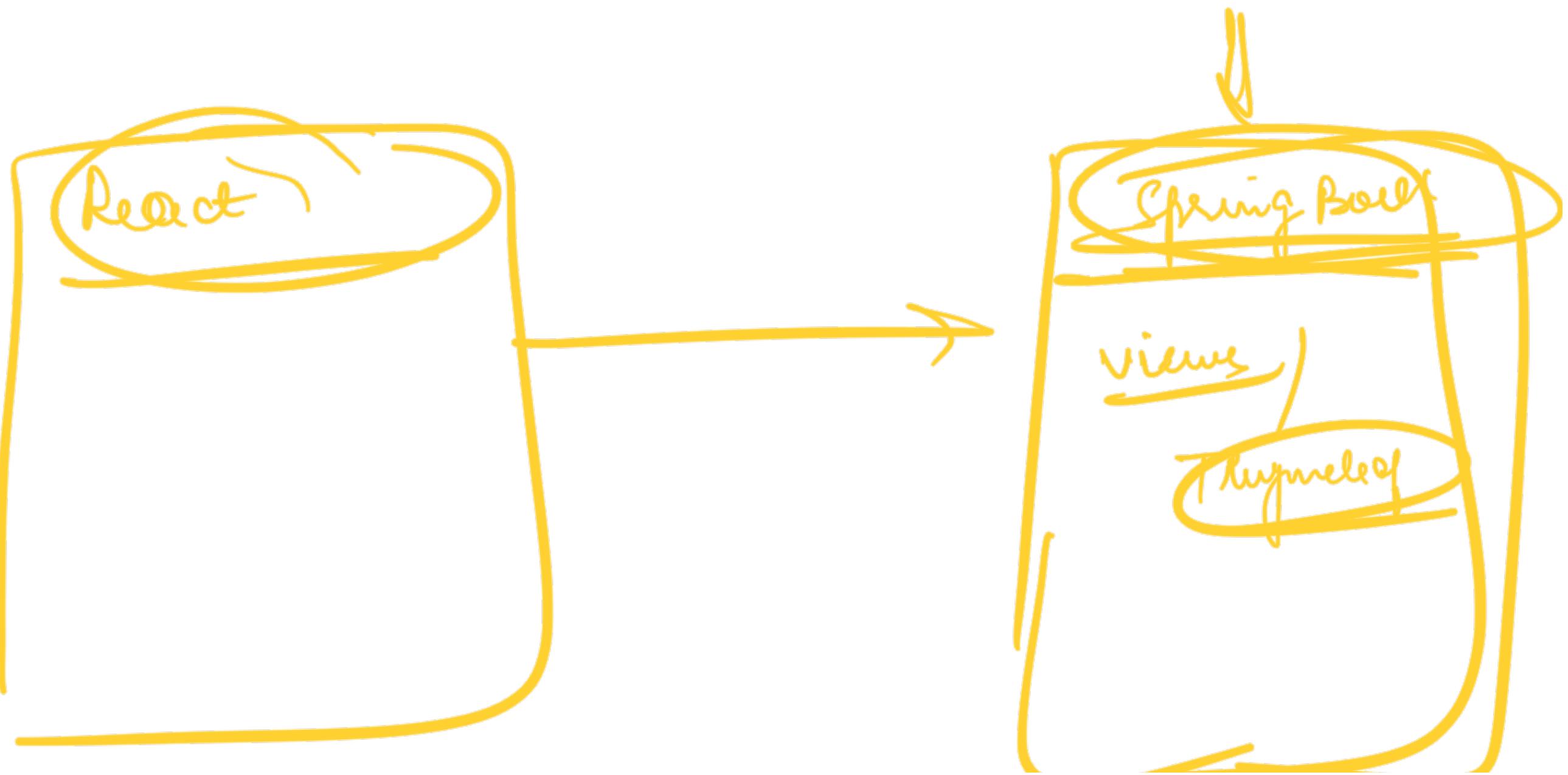
• Run ... hand databases

via lists

Sat: Code Parking lot

admv-video-profile-map

confvideo-profile-map  
Tvideo profile-id /



Ruby on Rails