1. Good Evening

2. Lecture begins at 9:05 pm

3. Topic → Design of Parking Lot
   & Remaining TTT discussion

Agenda

1. Designing of Parking Lot
2. Remaining TTT – Undo, Order!

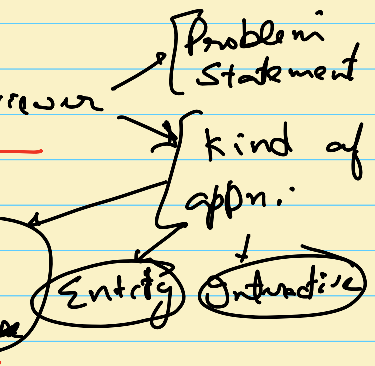
Design Parking Lot

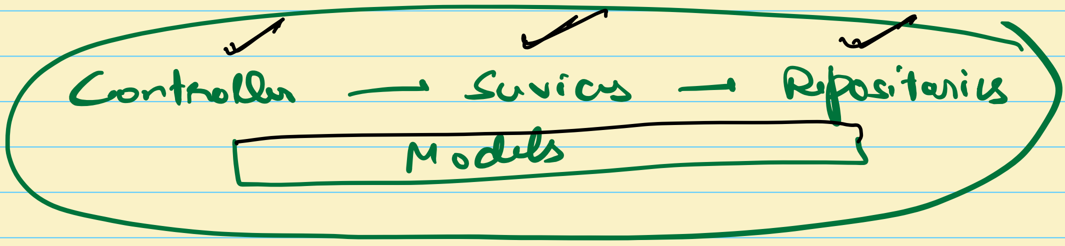
- Pen → Entity
- TTT → Interactive App
- Parking lot → MVC [Not real Db, No framework]
- BookMyShow → MVC + Db + SpringBoot
- Splitwise

→ 2.5 = 30 min + 1.5 hr + 30 min
coding.

→ Design a Parking Lot

1. Overview [Align with interviewer



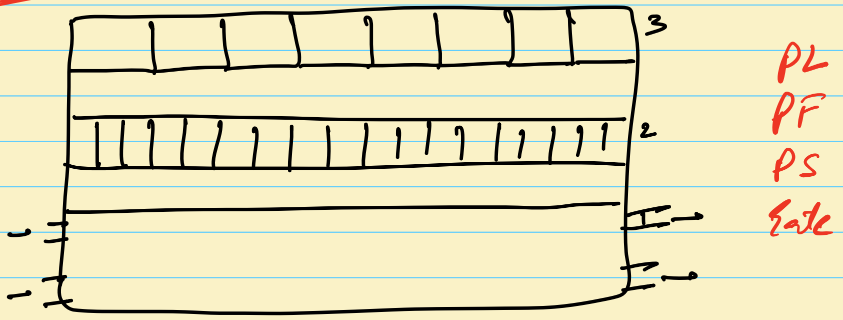


2. Requirement Gathering

1. Draw a sketch [Top- Down]

2. Imagine userflows.

Strategy 2.1. Draw a sketch

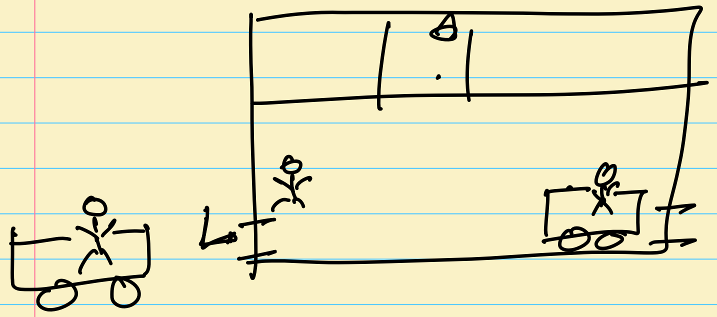


1. A Parking Lot may have multiple gates
2. Gates can be of two types [→ entry, → exit]

3. Parking lot have multiple floors.

4. A floor in Parking Lot has many Parking slots.
5. Parking slot support different type of vehicles.

Strategy 2.2 → Imagine user flows



6. [Get a ticket/~~token~~ at the entry gate [allocation of slot]]
→ vehicle type

7. A bill is generated at exit gate according to duration, vehicle type, timing

8. Payment has to be made
- online
- cash.

3. Clarifying Requirements
- → Edge cases
- → Behaviours

Thinking of changing parts.

2 min

1. Calculation of Bill / Fare
- → Multiple algos possible.
- → 1. Duration → Vehicle Type.
- → 2. 1st hour = 10 / hr.
  2nd hour onwards = 5 / h

2. Allocation of parking slot
1. First person, nearest slot
2. " , farthest slot.
3. Random.
4. Premium.

⊛ Concurrency concerns = BookMyshow.

S-8 K 1.5 hour.

RG

1. Overview, 2. RG via sketch, 3. RG via user flow,

4. [Behaviours & Edge Cases] → Changing Algo [Strategies]

* Class Diagram
  1. Check Nouns
  2. Visualise
     - Sketch
     - User flow

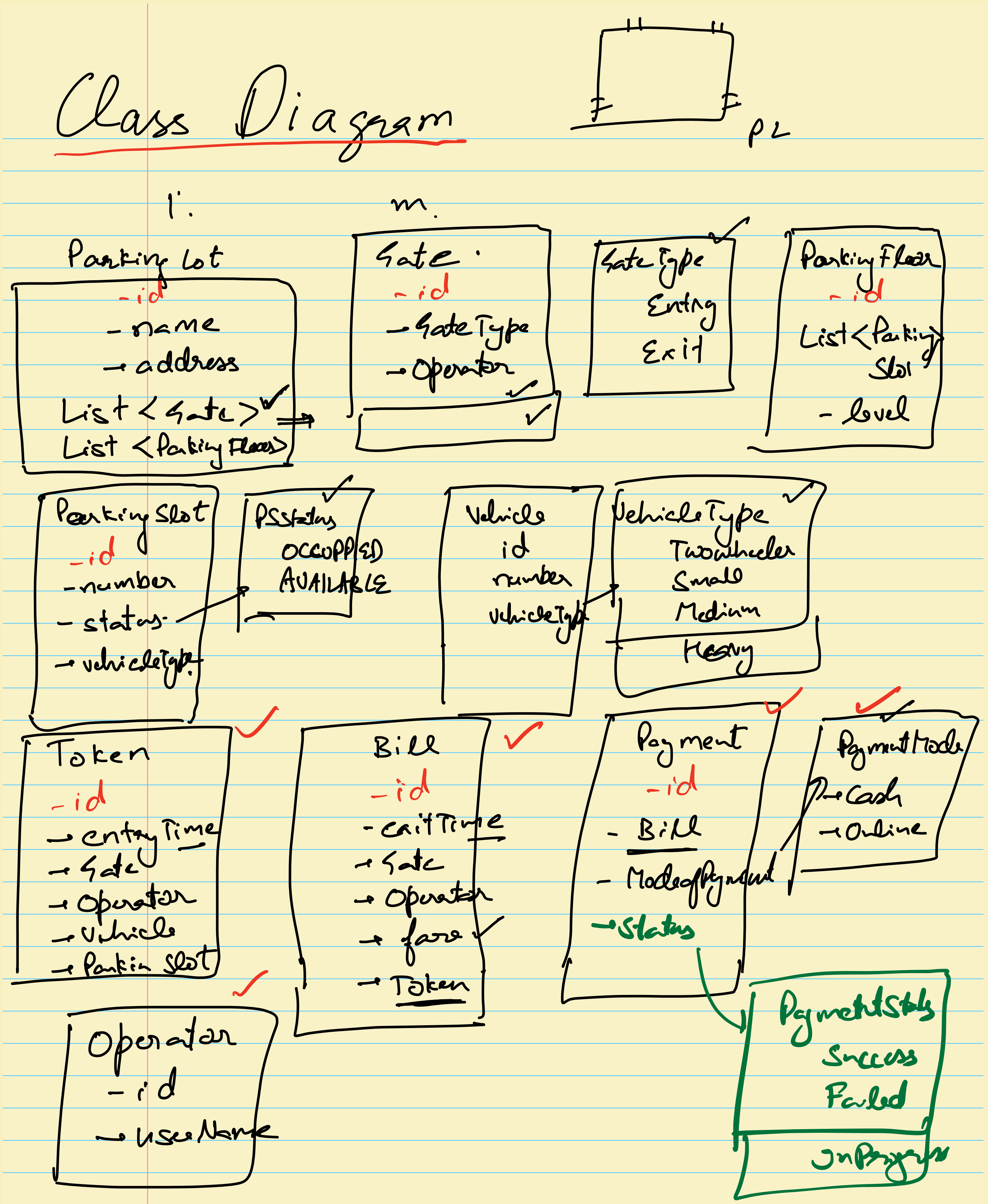
Use case Diagram

Coding
1. Models.
2. Req. by Req.

Break → 9:58 – 10:05

↳ Make class diagram.

Class Diagram



→ << Slot Allocation Strategy >>
allotslot(Gate, VehicleType)

→ << Calculate Fare Strategy >>
calculateFare (entryTime, exitTime, VehicleType)

Schema Design

1. Make a table for all classes & enums in models
2. Make columns for all primitive data members, & strings & date times
3. Identify cardinality & make relations → $\underbrace{1:1}$, $\underbrace{1:m}_{\text{FK on m side}}$, $\underbrace{m:m}_{\text{Separate Table}}$

→ Remaining (30 min) TTT will be discussed in next class [Undo, O1WS]

Assignment $\rightarrow$ 1. Make models with getters & setters