Agenda:-

Design Book My show.

(Movie ticket booking app).

⇒ Concurrency.

Steps in any Machine coding round.

1) Overview.
2) Requirement gathering.
3) Clarify requirements
4) Class diagram.
5) Schema Design.
6) Code.

Overview.

↳ know / Don't know the System.

⇒ We are aligned with the interviewer.

2) Requirement Gathering.

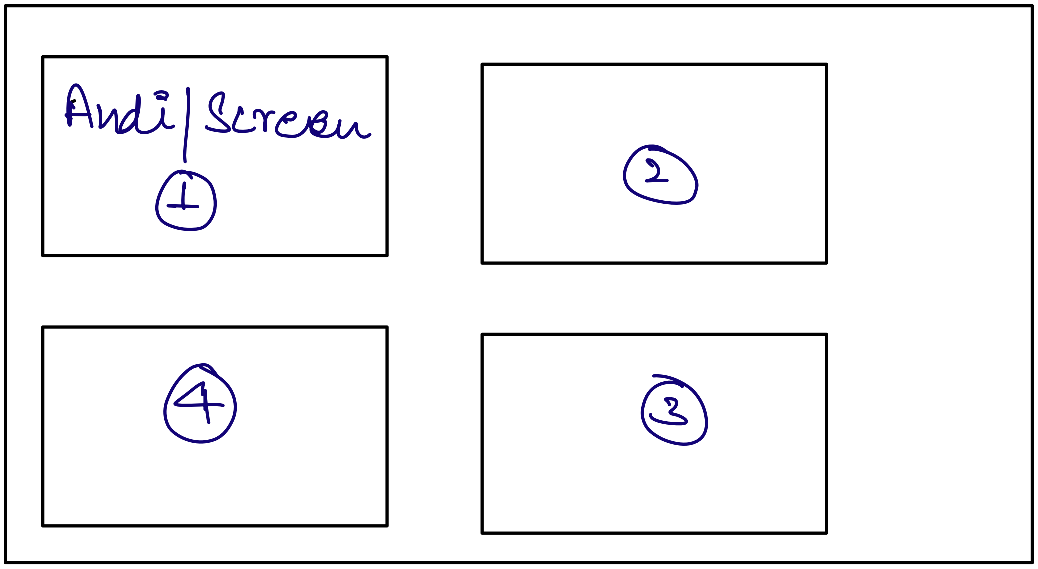
1) Users should be able to book tickets.

2) BMS has multiple regions (city). User should be able to select their city.

3) User can select the movie they want to watch.

4) User should be able to select the seats as per their preference.

5) Each region (city) can have multiple theatres.



(Theatre)
PVR Ambience Mall.

6) Each theatre can have multiple screens/ Audi. Inside a theatre multiple movies can run at same time.

Avengers.	Avengers.
7:00 PM	10:00 PM
PVR	PVR
Audi-2	Audi-2

7) We book a seat for a particular show.

8) A Show is defined as a particular movie running in a particular theatre at some specified date & time.

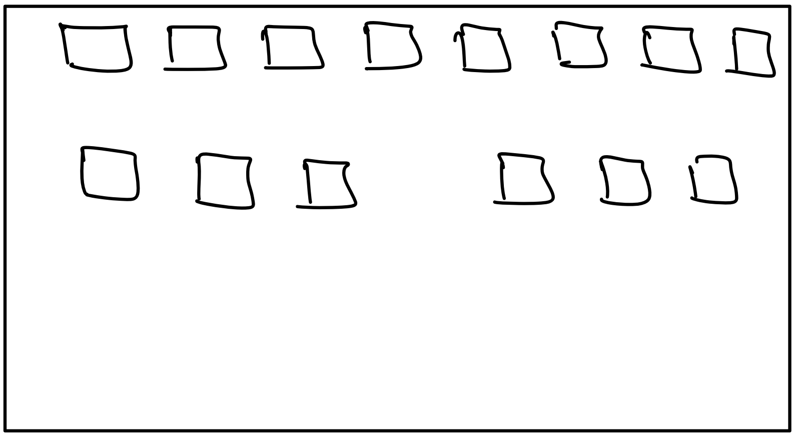
9) How many seats a user can book in one Ticket?
↳ 10.

10) No ② people should be allowed to book the same seats for a particular show.
↳ Concurrency issue.

Avengers.	Avengers.
10:00 AM	7:00PM
↓	↓
200Rs.	300 Rs.

Weekdays | Weekends

2D | 3D | IMAX.



⇒ Price : Seattype + Theatre + time + day + Movie + features.

Show.

[2D | 3D | IMAX | . . .]

11) Price is a funⁿ of SeatType & Show.

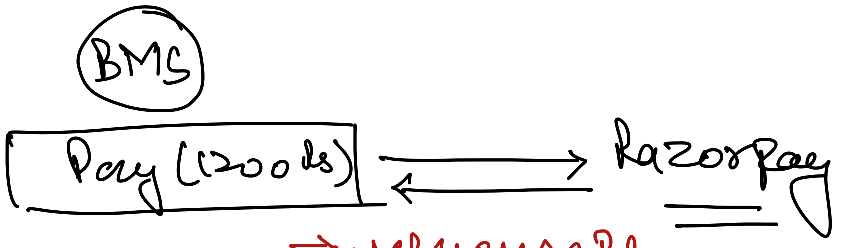
Show_id	Seattype	Price
123	GOLD	300
123	SILVER	200
124	GOLD	400

12) Store the details about Movie.
(name, cast, languages, genre, ...)

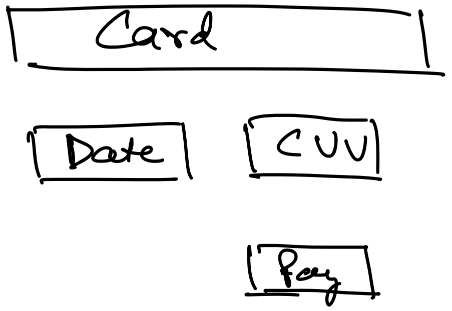
13) Payment. ⇒ will be outsourced to some 3rd party (Razorpay / PayU).

⇒ Wallet ✗

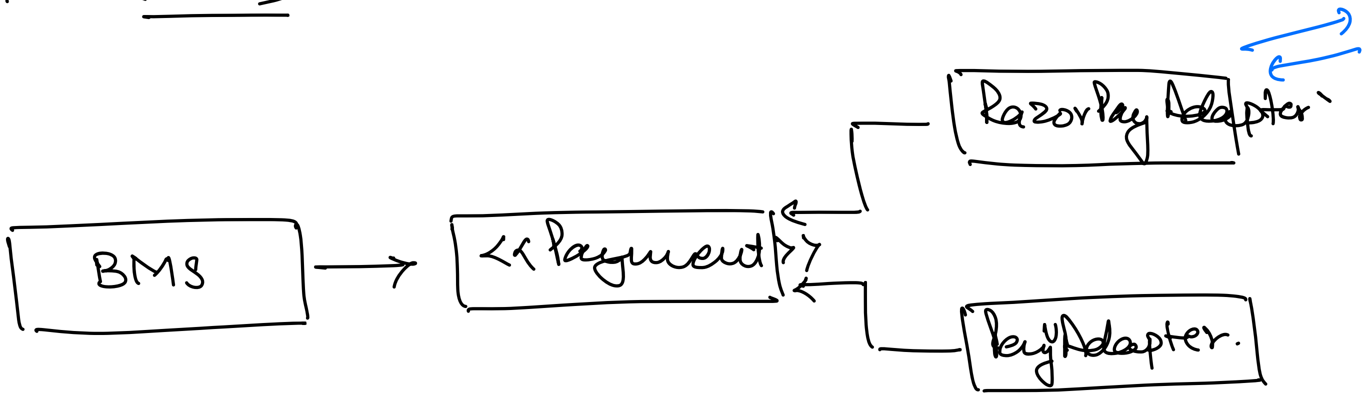
⇒ Only online payments.
⇓
All the payment modes are accepted.
(UPI, CC, DC, Netbanking).



→ referenceId
→ Payment status.

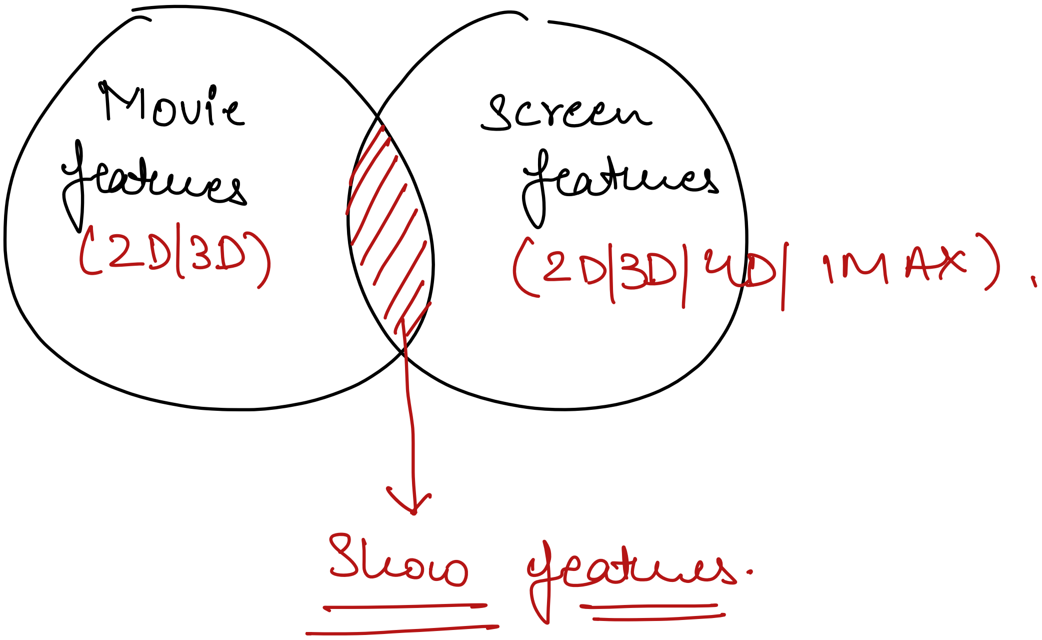


⇒ Adapter.



14) Features. → 2D | 3D | 4D | IMAX | Dolby

Movie Screen

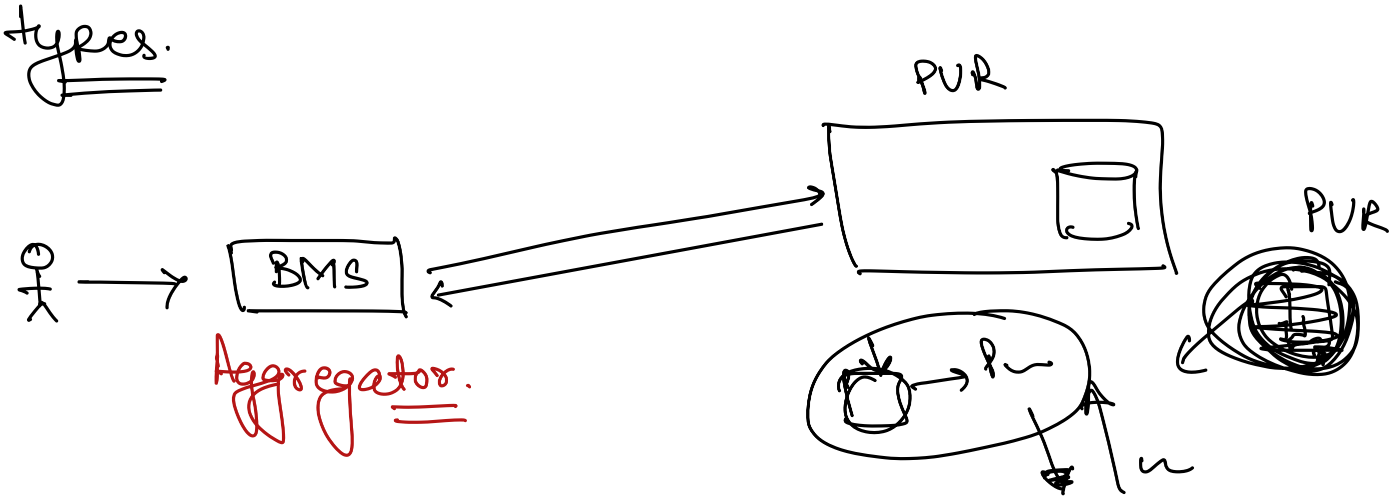


15) For every movie we'll store multiple languages.

⇒ Food / Beverages.

16) Each theatre will have multiple seat types.

#

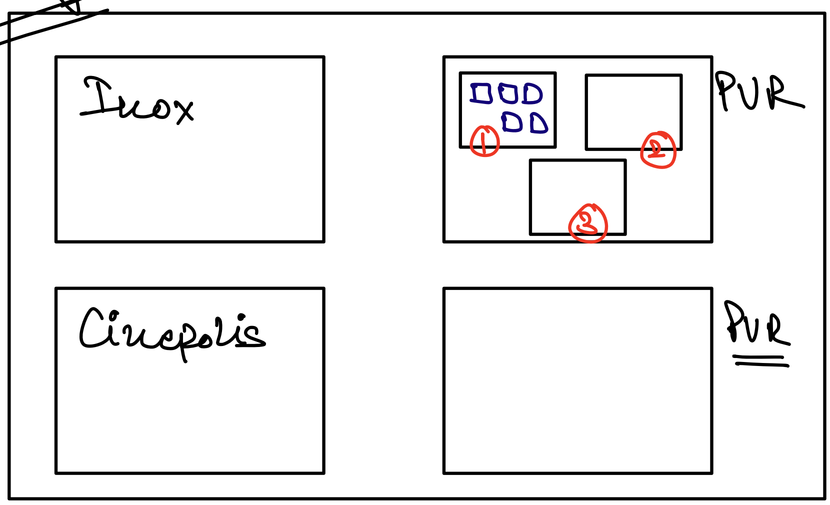


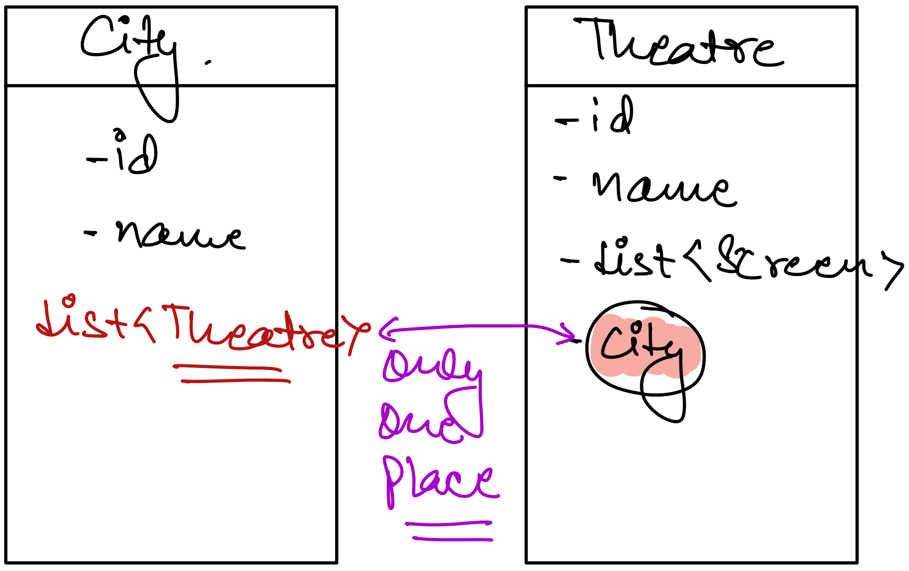
⇒ We are going to maintain DB.

Class Diagram.

- Nouns.
- Visualisation.
  - Physical structure.
  - User journey.

City/Region.





access Pattern

Screen

- id
- name
- List<Seat>
- List<feature>

feature

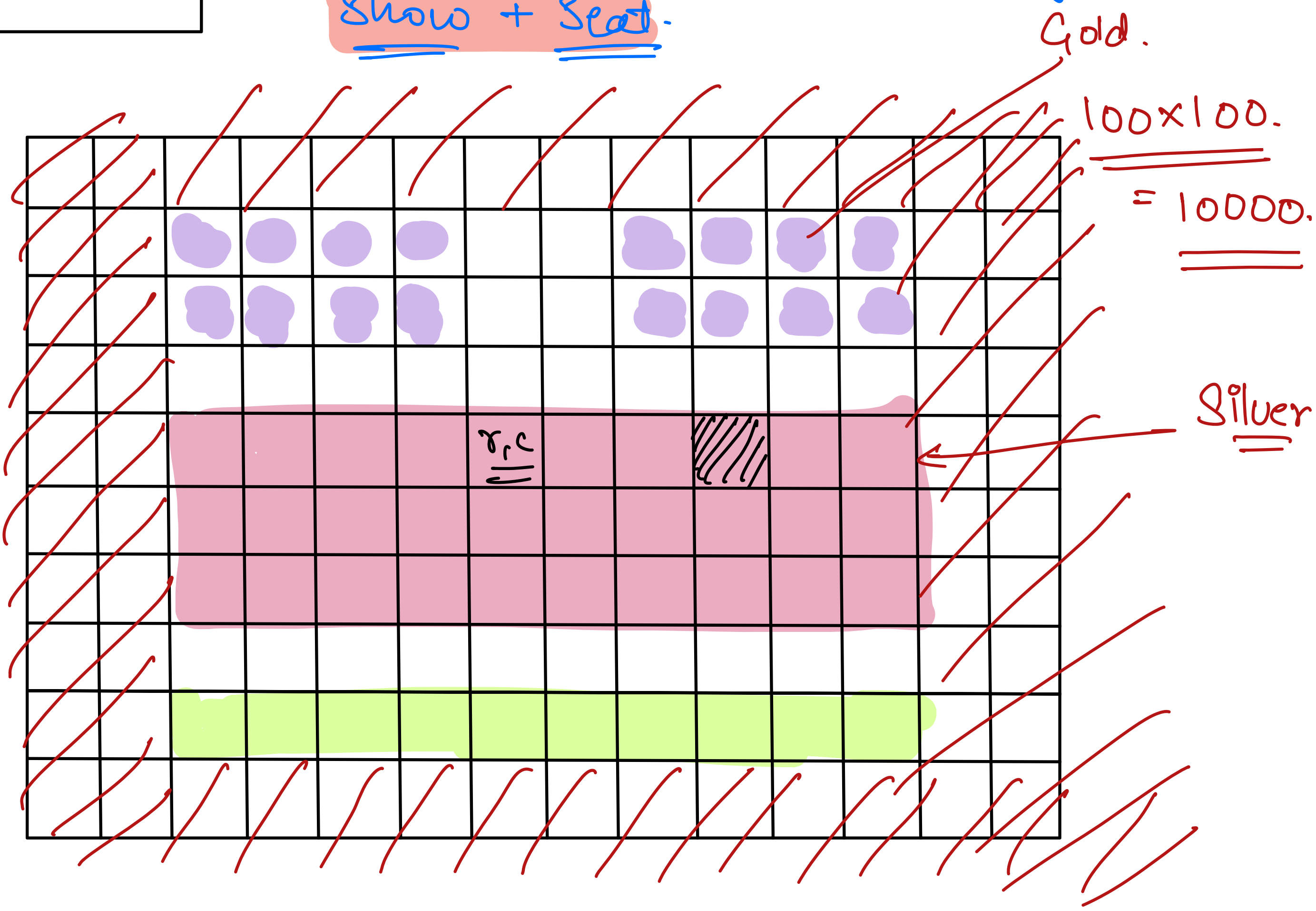
2D,
3D,
IMAX,
DOLBY,

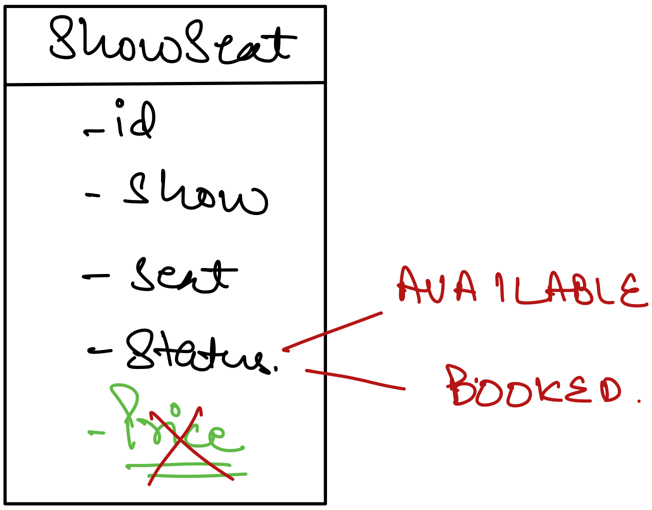
Seat

- id
- name
- row
- col
- seatType
- ~~Status~~

GOLD
SILVER
PLATINUM

Status is an attribute of Show + Seat.





Show_Seat

id	show	seat	Price

→ 100

⇒ for a show, price of all the seats of ① type will have same price.

Show + Seat Type.

Show_seatype

show_id	seat_type	Price
123	GOLD	500
123	SILVER	300

Show

- id
- Movie
- Start Time
- endTime
- Screen
- features
- List~~<Seat~~>

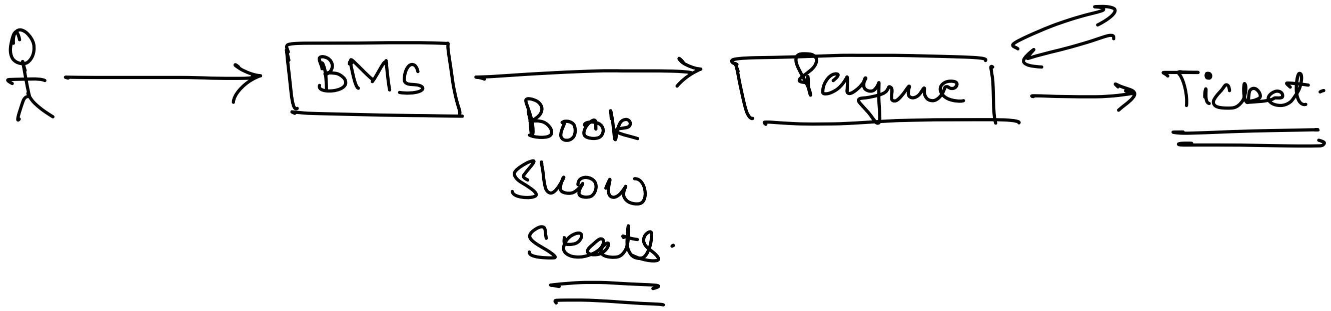
ShowSeatType

id

Show

SeatType

Price



Ticket

- id
- Status — CONFIRMED. / PENDING.
- List<ShowSeat>
- User
- amount
- List<Payment> ⇒ Retries, refunds..

Payment

- id
- reference id
- amount
- Status — SUCCESS / FAILURE.
- Mode.

——*——

→ Try to Do schema Design