(14) LLD of bookmyshow

(Movie booking System)

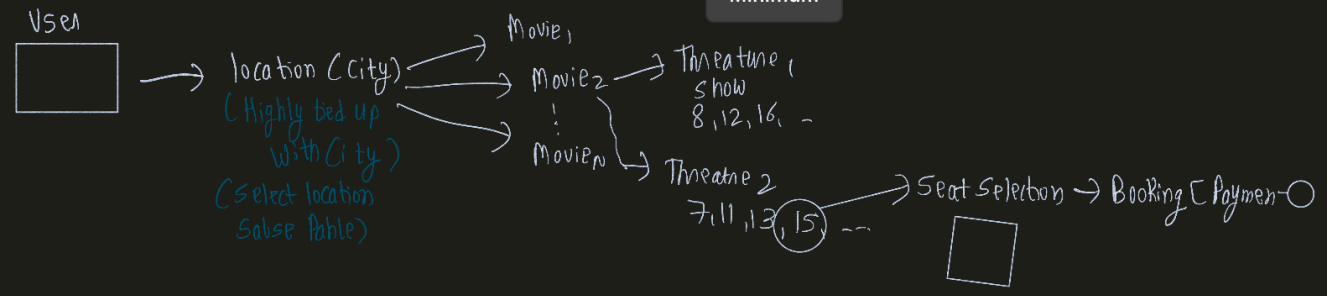
Design + Concurrency Management + Code

Lek ticket do logo ko na mile

Rough flow

Up to Down (as Movie is up, not at bottom) Approach

Minimum



Objects

* User
* Movie
* City
* Theatre
* Screen (hall)
* shows
* Seats
* Booking
* Payment

Theatre Controller (7)

Map<City, List<Theatre>>

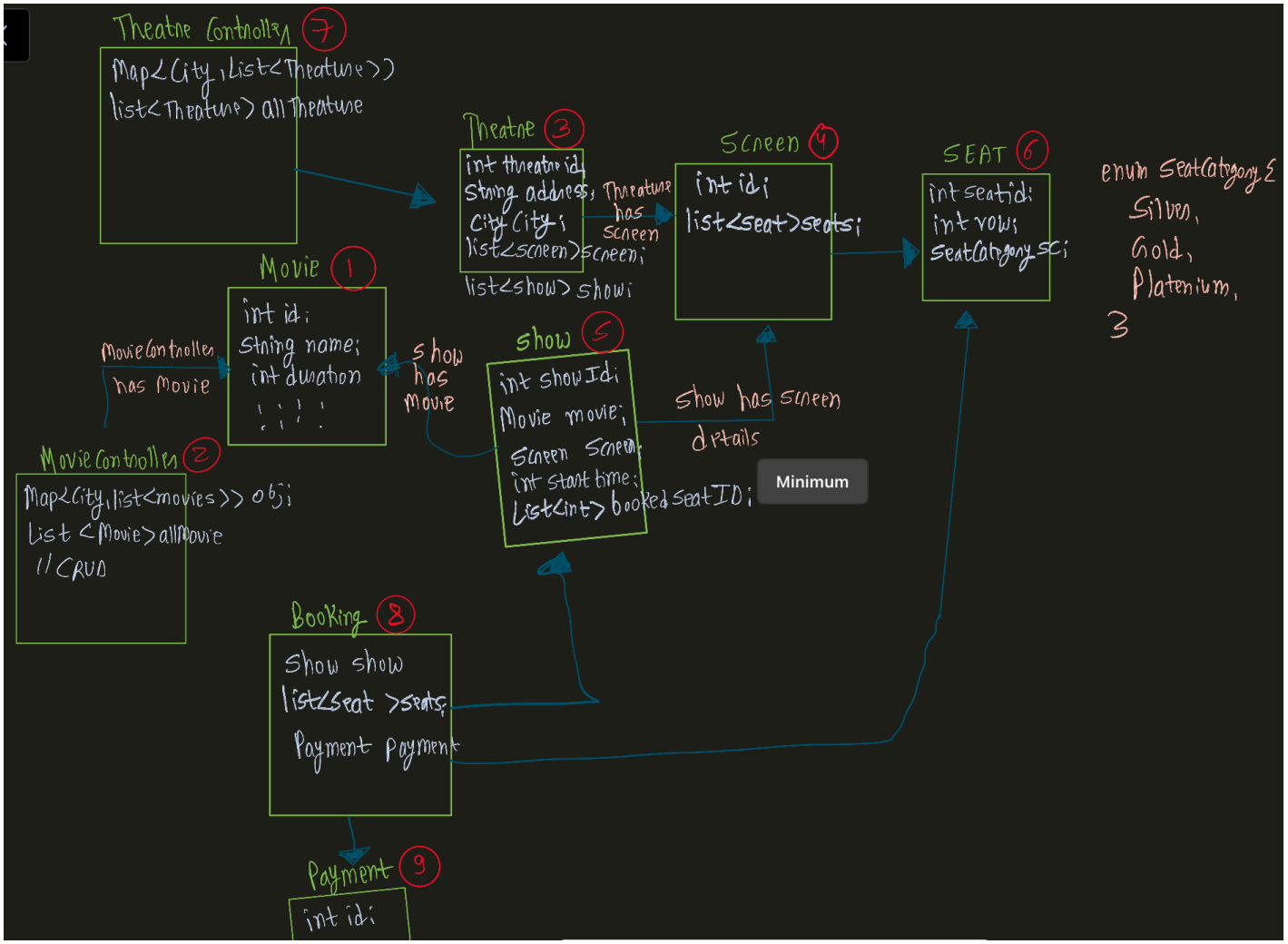
list<Theatre> all Theatre

Theatre (3)

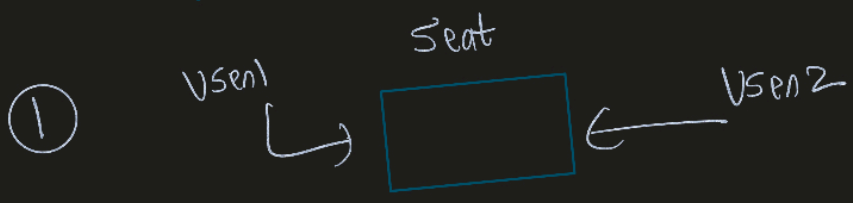
int theatre id

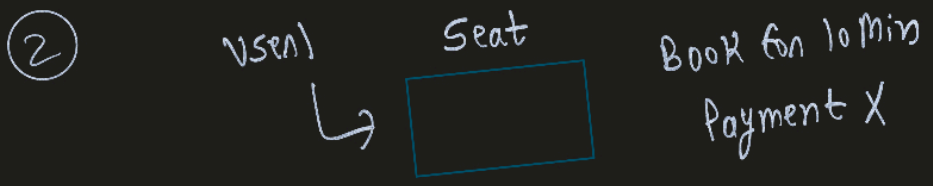
String address, Theatre has

City city;


Theatre Controller 7
Map<City, List<Theatre>>
list<Theatre> allTheatre
Theatre 3
int theatre id;
String address;
City City;
list<screen> screen;
Theatre has screen
list<show> show;
Screen 4
int id;
list<seat> seats;
SEAT 6
int seat id;
int row;
seatCategory sc;
enum SeatCategory {
Silver,
Gold,
Platenium,
3
Movie 1
int id;
String name;
int duration
MovieController has Movie
show has movie
show 5
int showId;
Movie movie;
Screen screen;
int start time;
List<int> booked SeatID;
show has screen details
Minimum
MovieController 2
Map<City, list<movies>> obj;
List <Movie> allMovie
// CRUD
Booking 8
Show show
list<Seat> seats;
Payment payment
Payment 9
int id;

Currency handling





2 ways to handle it Concurrency

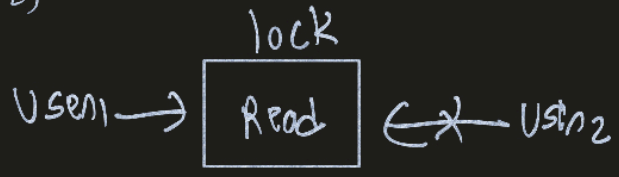
Optimistic (better)

U_1 → Read → Update → V2 (Version2)

Usen2 → Read

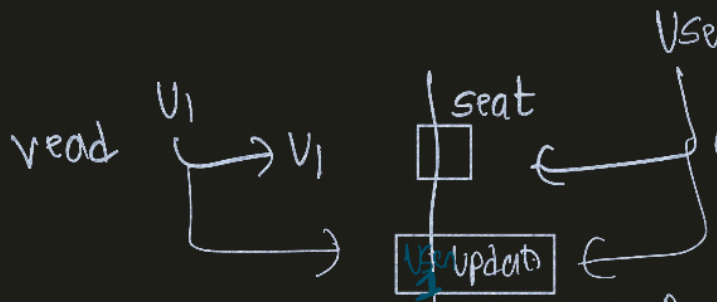
* it allows Multiple users to read, at the time of update, check if still same Version then update Version, other Don't lock it.

Pessistimic



* Jab Usen1 read kar raha ho tabhi lock kardo after updating give to Usen2
* U_1 & U_2 Can't read together

* har row ka Version maintain karenge



(ek hi andar ja payega, as in synchronized block)

User1 will check if
User1 Version == Row Version
(V1)

Minimum

So, book kar dega
& Update Version1 to V2
& release lock
& Now User2 will go inside

Now, for User2, V1 ≠ V2
failed booking, retry

* Redis use to give lock timer
expiry time we can give
to release lock,
for example 10 min