LLD - Book My Show

1. BMS Design
2. Role based access control
3. ORM - JPA
4. Passwords
   - interfaces

ISO 9001
SOC2
GDPR

Book my show

Questions

1) Event bookings

- Movie

2. Login users can book
3. No limit on the no. of shows
4. Admins → update/edit movie
5. 10 seats max
6. A user can edit a booking and cancel the booking

7. customer, admin

8. Types of seating
   - GOLD
   - DIAMOND
   - PLATINUM

9. 1hr cutoff
   - not be allowed to book, edit or cancel

10 City

↳ Cinema

↳ Halls

11 Payments

- UPI
- Card
- Net banking

12 Search, filter & sort

movie

name

Location

Cinema

Language

Genre

Rating

(1) Seat type

② Day of the week

③ Time of the day

④ Cinema

⑤ Movie

Pricing

---

① Problem statement — 5 min

② Clarification — 5 — 5-8 ✓ prioritise

+5 = 10 min

(3) Requirements → 5min

20 min

1:30 -2

→ Cinema

BMS
- cities

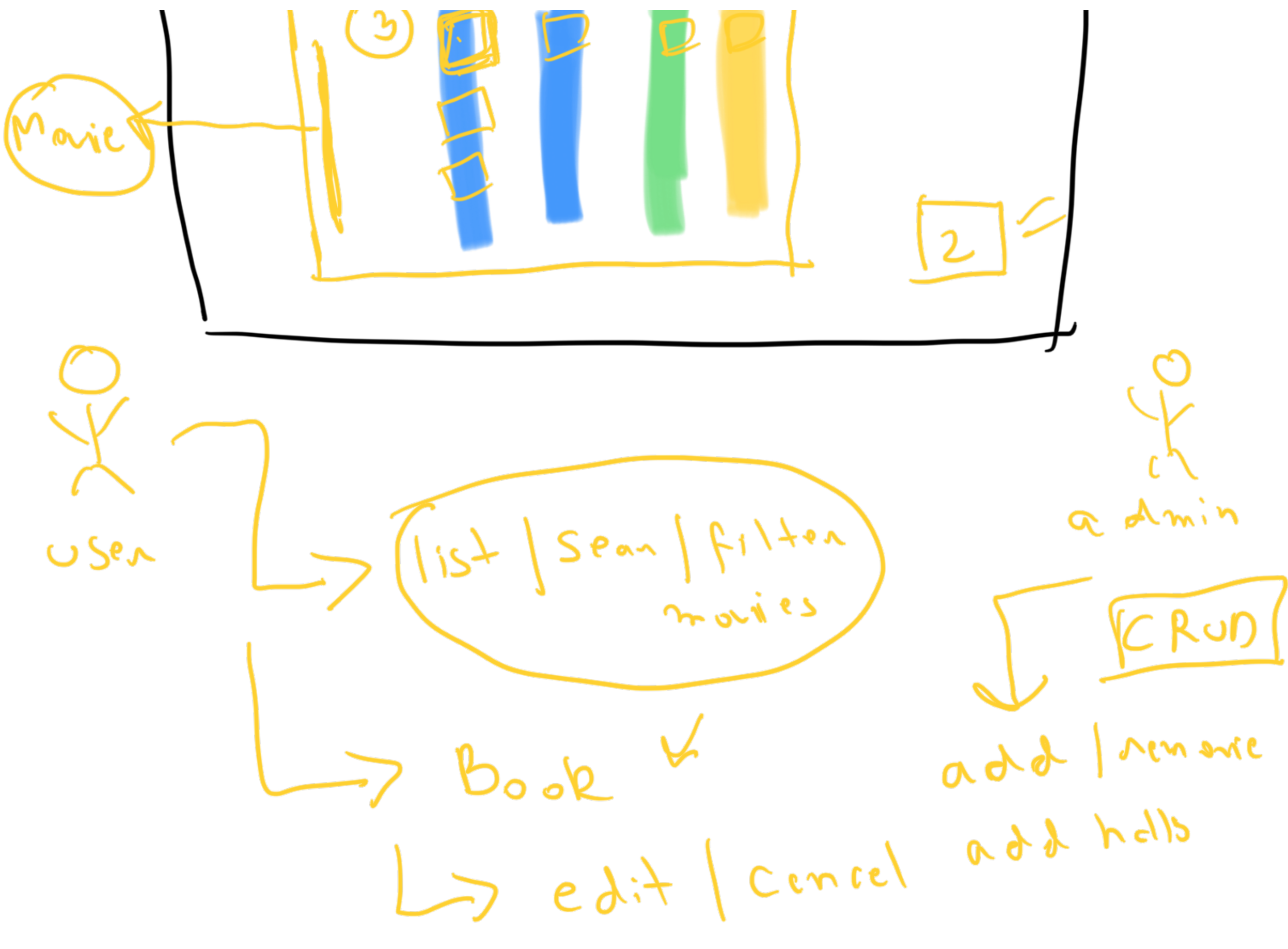
City
Cinema

1

metadata

Cinema

1


Movie
3
2
user
list | seen | filter
movies
Book
edit | cancel
admin
CRUD
add | remove
add halls

booking

CRUD

Ticket booking

Class Diagram

Hall

- hall no
- shows[]

m:1

Show

- start
- end
- Movie
- Cinema Hall
- Seats

HallSeat

- location
- seatType

Show

Show Seat

Hall Seat
status

AVBL

BOOKED

Single Source of truth

User

Admin

book

book

Super Admin

EDITOR

Red Only

Roles

ADD_A_MOVIE

roles().includes

if (user . ge [illegible] (ADD)

Deciding if a user has access to a resource with a role

RBAC - Authorization

Authentication

login

vs

login does a user have access to a resource

Password

Oauth token

Basic Auth

Bearer Auth

Token

Resources

→ JPA

→ Password-

Fly

User

- username
- password
- roles
- email

book

1:

Admin

- user
- designation
- phone
- .

editMovie

→ Role — User

RBAC

interface

Booking

1. coupon

Booking

Showcat

Cardinality

1:m

Concellation

m:n

→ Delete data (CRUD PE)

Deactivate | Tombstoning | soft-delete

booking Show Seat

m:n

-1
-2

6:14 - 6:20
10:45 - 10:50

BookMyShow

↳ Design & JPA intro

↳ Concurrency

Normalized

Data application

↳ consistency

↳ application

A

B
A

1
2
A
b
A
B
A
B

Service B
Service

Consistency
X

write heavy

pass
plaintext
X
Secure
Hashing

Salt encoded password

1. plaintext → unreadable

   123 → #AbCy2

2. hash → 123

   Salt provided by + hash => 123

JPA

→ ORM

Application

Db

1

SQL query →

Mongo q-ry →

② writing & maintaining queries is a pain

③ optimizing native

Abstraction

Save

Find

M no.

— native queries

ORM

JPA

Hibernate