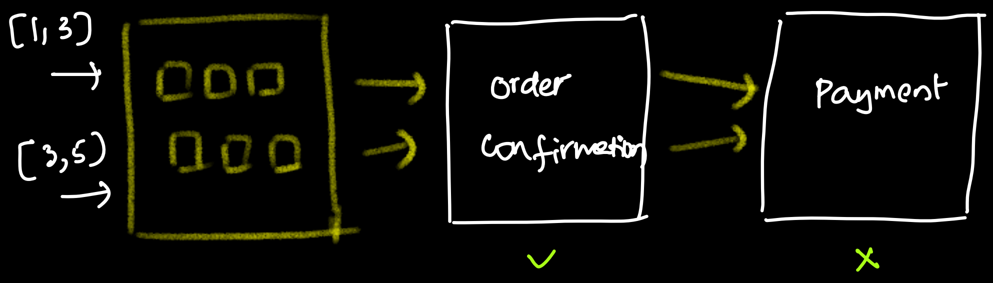
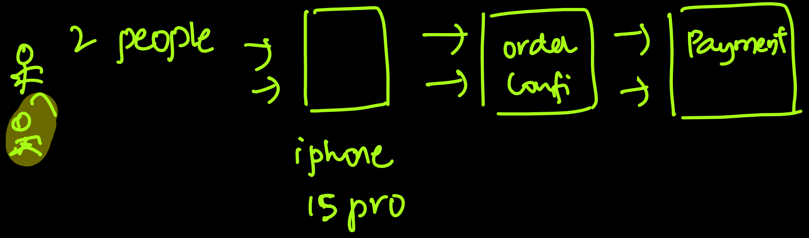
Book a ticket.

```
Booking Controller
    public [redacted] bookMovie( [redacted] input)
        bookingService. bookMovie(input);
```

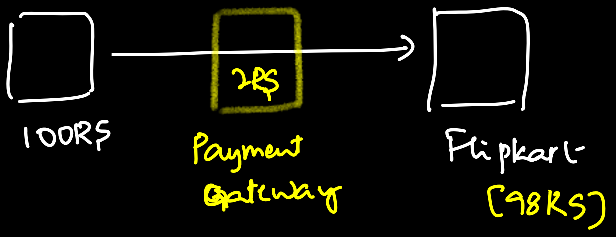
BookMovieRequestDto
- userId
- ~~showId~~
- List<ShowSeat>
  - ↓
  - [show, seat]

BookMovieResponseDto
- bookingId
- amount
- responseStatus. (success/failure)
- responseMessage (errorMessage)



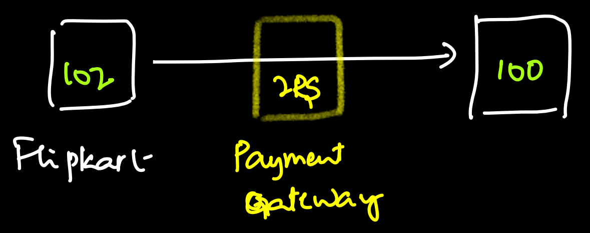


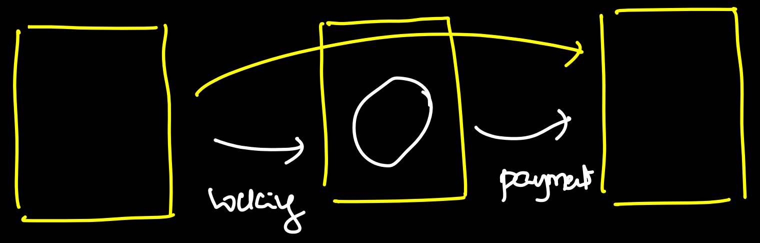
both of them made the payment, but order is confirmed for one person only.



Incurring of loss of 4Rs.

& also bad experience for user.





We need to handle more than one transaction.

DB transactions.

When should transaction start?

user clicks on book button . .

① Long running transactions,

start a transaction that takes a lock on seats we've selected, once the payment is done / after they leave we release the lock.

Avg time for order conf + make payment ≃ 5min

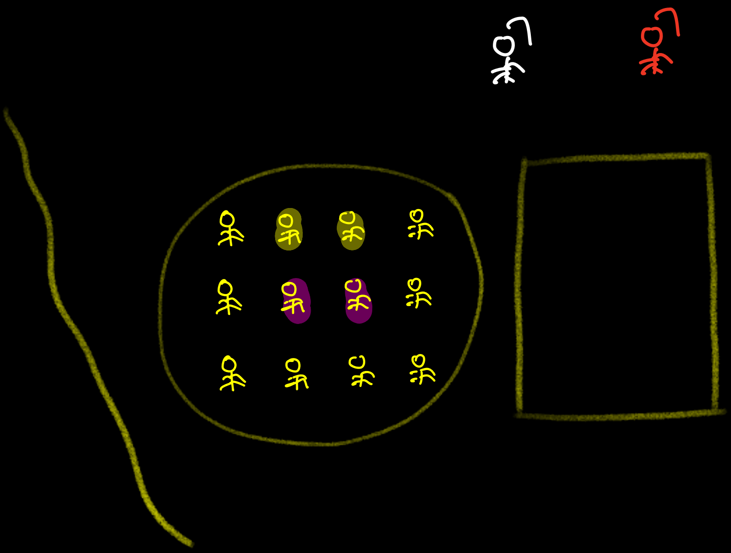
Its not a good idea to keep transactions running this long. (lot of resources | memory).

② Soft locking

idea: Don't keep the transactions running till the payment is done [keep it only for sometime].

Transaction level → serializable.

Simulate DB lock.



a field named status

[available, locked, booked]

Imagine the data like below.

show-seats

id	seat-id	show-id	status [available, locked, booked]
	1	10	~~Available~~ locked
	2	10	Available
	3	10	Available
	4	10	~~Available~~ locked
	5	10	Available

Two request

[1, 3]	}	only one person should set locking
[3, 5]		

[1,3] →

1. check if status is available
2. Then update the status of [1,3] → locked.
3. end transaction
4. I'll move to payment page.

Q^n:

$P_1 \xrightarrow{1,5}$

$P_2 \xrightarrow{5,7}$

$P_3 \xrightarrow{7,8,9}$

Max tickets.

2 people

P_1 P_3

Min tickets

1 person

P_2

Qn: What if the first guy never made the payment.

let's we allow one person to make the payment for 10 mins.

show-seats

id	seat_id	show_id	status (available, locked, booked)	locked At	
	1	10	~~Available~~ locked	~~8:12:00~~	8:23:00
	2	10	Available		
	3	10	Available		
	4	10	~~Available~~ locked	~~8:12:00~~	8:23:00
	5	10	Available		

Another user (1, 4) , 8:23:00.

→ (1,4) status

if it's locked ,

currentTime - LockedAt > 10

8:23:00 − 8:12:00

11:00 > $\underline{\underline{10}}$.

=> Allow him to move to confirmation page.

Booking Service

i/p : userId, showId, showSeatIds.

1. get user details for userId
2. get show details for showId

↙ start trans

---

3. get List<ShowSeat> for given Ids
4. check if all of them have status as available.

if not, check if they're locked & lock has expired

5. Mark the status as locked.

update lockedAt

6. Save List<ShowSeat> in DB. ← end transaction

---

7. create corresponding booking obj / status -> payment pend

8. Return response (Redirect to confirm).

How to create transactions in spring boot (spring data jpa).

1. whatever needs to be done in a transaction, extract into a public method.

2. @Transactional(isolationlevel = ________)

How do you interact with db?.

Repositories.

But, because of spring data jpa, interactions with db will become very easy.

Student

id	name	psp
1	Keerthi	90

1. establish db connection
2. write query
3. execute query
4. map to your objects
5. get the data

Repositories

T T

interface StudentRepository extends JpaRepository<Student, Long>

Student Controller

studentRepository.
- .save(Student)
- .findById(123)
- .count()
- .findAll()
-