Concurrency Handling In Booking users users should book seat same seat Book - Seat should be in booked	
usur Even if pays Book Seat state Seat sexual son	-
users users Seat should book Seat should book Seat should Book - Seat should be in booked	_
Seat should book sur! Book - Sailed, the slat slat should be in booked	
Seat should book sur! Book - Sailed, the slat slat should be in booked	
Seat should book sur! Book - Sat should be in booked	
Book - Seat should be in booked	,
• usur! Even if pays 18 due on Book - Bailed, the seat stat should be in booked	
Book - Book of Seat should be in booked	
Book - Book of Seat should be in booked	
Book - seat should be in booked	rent
seat stat should be in booked	
be in booked	
State for 10	
minytes	
Con Colod Colod Colod	
For Concurrency Control we have 2 types	
of locks	
optimistic Pesimistic	
-> In optimistic mo locks In Pesstimistic, tocks as	,
placed when data 9s placed when data is read	_
bread. prevent other transaction	
from modifying it until	
> Before write operation, from modifying it until	
It checks if another	
It checks if another transaction has modified the data since 9+	
that data since 9+	
was last read.	-
	`
Scanned by Scanner	; ;

Mechanism oftimistic Pessimistic ir Read . Real and Lock 2. Edit 2. Edit z validate 3. Commit and Rollback 4- Ros Commit and Rollback Advantages + Reduce Locking overheas. 1. Prevent other from 2- suitable for system with Low likelihood of Conflicts 2-Simpler Lack Ey. useal ---) Read users Read weil -> Read - lock 410g2 upda Herk Then failed if vs Applied Lock wer select the seat else users will release user booked dock after updatation spat · While update we tock ? change the version

Scanned by Scanner Go

Page No. Date / Best Concurrency. Control for Real Read In Dessimistic while user's in Reading 9t is locking the seats and this preven other users to book seats. So for Booking Systems optimistic control method is most suitable since it applies the clock on the time of updation only.