City: cityId, name, location (lat, long)
Theatre: theatreId, totalScreens, cityId, theatreName, <Other meta data like when this theatre got onboard to app>
Screen: screenId, theatreId, capacity <Other metadata>
Movie: movieId, name, GENRE, releaseData, <Other meta data>
Show: showId, startTime, duration, screenId, <Other meta data>

ScreenSeat: seatId, seatNumber, type -> (T1, T2,.....), <meta data>
ShowSeat: showSeatId, showId, screenSeatId, price, status -> (BOOKED, AVAIL, PROCESSING), **bookingId**, <meta data>

User: userId, name, mobile <meta data>
PS-SQL Oracle
Booking: bookingId, transcationId, date(epoch), status, **showId**, userId, |seatId<vector>|

transaction: transactionId, <other meta data>
status -> IN_PROGRESS, COMPLETED, FAILED, CANCELLED,

APIs:
v1/search(location, movie, date, 1 mile around city) ->

v2/search(theatre, movie, time)

dataInsertion: post /
{movie}
put /
{movie}
delete /
{movie}

GET /seatView (userToken, showId)
POST /booking (userToken, vector<seatIds>, showId)
/payment (userToken, MODE, bookingId)

HIGH LEVEL DESIGN:

Concurrency handling?
Passimitistic lock
Optimistic lock
2 phase commit protocol

Transaction begin
res = SELECT * FROM SHOW WHERE SHOWID='$showId AND
'SEATID IN (s1,s2,s3)
if res.size()!=3:
    rollback throw error
UPDATE SHOW SET s1=UNDER_PROCESS, s2=_, s3=_
do payment
if payment succ:

```
        UPDATE SHOW SET s1=SUCCESS, s2=_, s3=_
do payment
        commit
else:
        ROLLBACK
commit
```