Low Level Design for BookMyShow

Shivansh Dubey Follow 3 min read · Mar 28, 2024

Movie booking application platform are one of the most common low level design interview question. Lets look at basic design for this application.

Requirements :

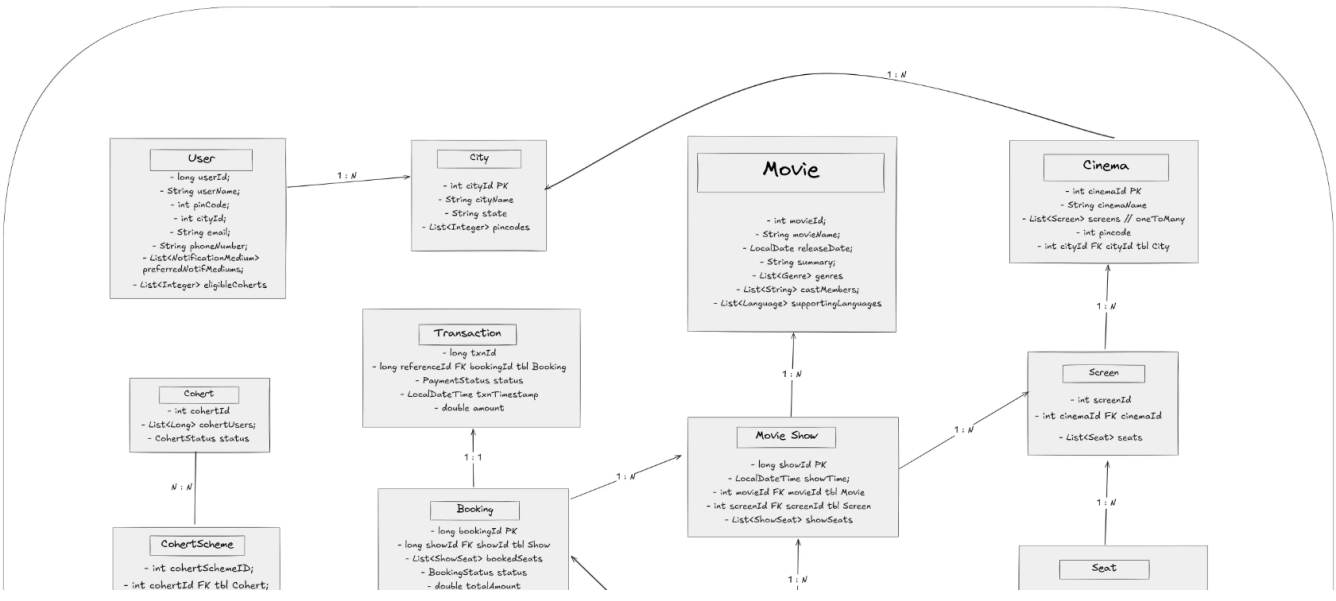
- Admin can add/remove movies, add/remove/cancel movie shows, register new cinema
- User can search/browse movies/cinema based on various criterias without login
- To create and complete booking, user needs to signup if acc doesn't exist.
- Users should be notified on events like Booking confirm/cancelled/show cancellation etc to his preferred modes of communication.
- User should be able to choose mode of payment(UPI, Netbanking, CC..)

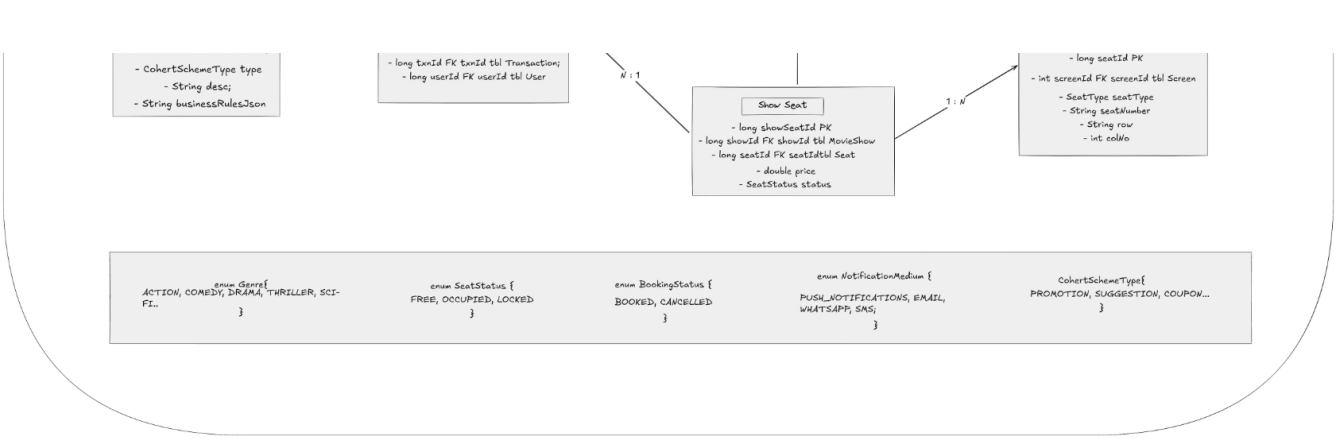
Entities :

```
Movie
- int movieId;
- String movieName;
- LocalDate releaseDate;
- String summary;
- List<Genre> genres
- List<String> castMembers;
- List<Language> supportingLanguages

enum Genre{
  ACTION, COMEDY, DRAMA, THRILLER, SCI-FI..
}

City
- int cityId PK
- String cityName
- String state
- List<Integer> pincodes

Cinema
- int cinemaId PK
- String cinemaName
- List<Screen> screens // oneToMany
- int pincode
- int cityId FK cityId tbl City

Screen
- int screenId
- int cinemaId FK cinemaId tbl Cinema// ManyToOne
- List<Seat> seats // oneToMany

Seat
```

```
- long seatId PK
- int screenId FK screenId tbl Screen //ManyToOne
- SeatType seatType
- String seatNumber
- String row
- int colNo


MovieShow
- long showId PK
- LocalDateTime showTime;
- int movieId FK movieId tbl Movie
- int screenId FK screenId tbl Screen
- List<ShowSeat> showSeats


ShowSeat extends Seat  // via inheritence, can also be done via composition
- long showId FK showId tbl MovieShow
- double price
- SeatStatus status


enum SeatStatus {
    FREE, OCCUPIED, LOCKED
}


Booking
- long bookingId PK
- long showId FK showId tbl Show
- List<ShowSeat> bookedSeats
- BookingStatus status
- double totalAmount
- long txnId FK txnId tbl Transaction;
- long userId FK userId tbl User


enum BookingStatus {

    BOOKED, CANCELLED
}

Transaction
- long txnId
- long referenceId FK bookingId tbl Booking
- PaymentStatus status
- LocalDateTime txnTimestamp
- double amount

User
- long userId;
- String userName;
- int pinCode;
- int cityId;
- String email;
- String phoneNumber;
- List<NotificationMedium> preferredNotifMediums;
- List<Integer> eligibleCoherts

enum NotificationMedium {

 PUSH_NOTIFICATIONS, EMAIL, WHATSAPP, SMS;

}


Cohert
- int cohertId
- List<Long> cohertUsers;
- CohertStatus status

CohertScheme
- int cohertSchemeID;
- CohertSchemeType type
- String desc;
- String businessRulesJson



CohertSchemeType{
  PROMOTION, SUGGESTION, COUPON...
}
```

. . .





Entity Relationship Diagram

. . .

Services :

```
AdminService {

    boolean registerCinema(RegisterCinemaRequest request)

    boolean addMovie(AddMovieRequest request)

    boolean addShows(AddShowRequest request)

    boolean removeMovie(int movieId)

    boolean removeOrCancelShow(List<CancelShowRequest> requests)
}


SearchEngineService {

  List<Movie> searchMovie(SearchCriteria criteria, Object... params)

}

enum SearchCriteria {
  TITLE, GENRE, CAST_MEMBER, RELEASE_YEAR...
}


BoxOfficeService {

  List<MovieShow> findShows(ShowSearchRequest request)

  List<Cinema> findCinemas(int movieId, int cityId, int pinCode)


}


UserService {

  boolean signup(SignupRequest request)

  User findUser(int userId)

  boolean updateUserStatus(int userId, UserStatus status)

  boolean addUsersToCohert(int cohertId, List<Long> users)

  boolean removeUsersFromCohert(int cohertId, List<Long> users)

}


BookingService{

    BookingResponse createBooking(BookingReq request)


    BookingResponse cancelBooking(ModifyBookingReq request)

}


TransactionService {

  boolean createTransaction(CreateTxnReq request)

  boolean processTransaction(int tnxId, PaymentMetadata paymentMetadata)

}

PaymentHandlerFactory{

  Map<PaymentInstrument, AbstractPaymentHandler> paymentHandlerMap;

  void register(AbstractPaymentHandler handler){
      paymentHandlerMap.putIfAbsent(handler.getInstrument(), handler);
  }

  AbstractPaymentHandler get(PaymentInstrument instrument){
      return paymentHandlerMap.get(instrument);
  }

}

AbstractPaymentHandler{

  abstract PaymentInstrument getInstrument();

  public boolean process(long txnId, PaymentMetadata metadata);


} -> UpiPaymentHandler, CreditCardPaymentHandler, NetbankingPaymentHandler...
```

```
NotificationEngine {

    NotificationResponse send(NotificationRequest request)

    List<NotificationResponse> bulkSend(List<NotificationRequest> requests);

}

NotificationHandlerFactory{

  Map<Notification_Medium, AbstractNotificationHandler> notificationHandlerMap

  // similar to as PaymentHandlerFactory

}

AbstractNotificationHandler{

  abstract Notification_Medium getNotificationMedium();

  abstract NotificationResponse send(NotificationRequest);

  abstract List<NotificationResponse> sendAll(List<NotificationRequest> requests

} -> PushNotificationHandler, EmailNotificationHandler, SMSNotificationHandler,
```

. . .

Database choice :

Movie, Genre, MovieShow, Cinema, Screen, Seat, ShowSeat, User entities have relational data characteristics associated with them. These data also doesnt grow at exponential rate and we have complex query patterns for which Relational Databases are well suited.

Booking entity data grows expotentially, neither we require any ACID properties while performing any operation on these data. Hence we can store them in NoSql DB to handle scale.

Transaction entity data requires ACID properties hence should be stored in RDS but this data also grows at high rate. This could be handled with mixed approach. At RDS level, we can scale horozontally with help of sharding. Also for transactions which are in past can be periodically moved to cold storage or Nosql for audit purpose and purge such records from RDS on regular basis during Non active BAU hours since this will require DB Downtime for indexes restructing.

Top highlight

. . .

You can download entire source code for above Problem statement from link :
Download Link

Low Level Design | Movie Ticket Booking | Bookmyshow | System Design Interview | Interview

213 5

Written by Shivansh Dubey
100 followers · 8 following
Currently working as SDE-III @ Meesho
Follow

Responses (5)

Adityakumarbari

What are your thoughts?

Surinder
Apr 1, 2024

How do handle multiple users booking same show or seats. Concurrent bookings ?
You mentioned booking doesn't need ACID, Can you shed some light how would concurrent bookings be handled without ACID database.
Your DB design also doesn't tell... more

91 2 replies Reply

Arunesh Sri
Dec 26, 2024

where are these claases BookingReq , ModifyBookingReq etc defined ,

there are quite a few such classes

7 Reply

Tushar Khandelwal he
Jul 23, 2024

> Transaction entity data requires ACID properties hence should be stored in RDS but thi

what is rds

56 Reply

See all responses

More from Shivansh Dubey

Shivansh Dubey

Low Level Design | Ecommerce

Lets look at basic low level design for ecommerce platform like Bigbasket, Amazo...

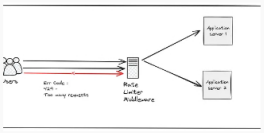
Mar 28, 2024 63 1

Shivansh Dubey

Low level Design for Twitter

One of commonly asked interview question is how to design Twitter. Lots of resources are...

Jun 6, 2021 41

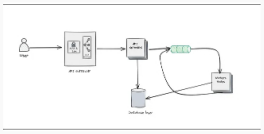


Shivansh Dubey

Rate Limiter System Design

Design a rate limiter

May 14, 2024 4



Shivansh Dubey

Web Crawler System Design

Problem statement:

May 13, 2024 4

See all from Shivansh Dubey

Recommended from Medium

Ankit Kumar Srivastava

Designing a Warehouse Management System (WMS) — ...

Introduction :

Aug 20, 2025 10

Gaddam.Naveen

Everything Worked Until We Used block() in WebFlux

if you are not a medium member then Click here to read free

3d ago 4

In CodeToDeploy by Kunal Sinha

System Design: Building a Payment Processing System

Tim Ozdemir

System Design: Food Delivery System

Online payments on e-commerce websites are now ubiquitous. A website presents you...

Dec 23, 2025 60

Proximity Search Deep Dive

6d ago

RocketScience

Thread Safety & Synchronization in Java

In the era of multi-core processors and parallel processing, writing concurrent...

Jul 26, 2025

Microsoft

Imran Wahid

Microsoft SDE 2 (level 61) interview experience

Hello everyone. I recently got the opportunity to interview with Microsoft, India. In this...

Oct 30, 2025 4

See more recommendations