**[Design a Movie Ticket Booking System](https://github.com/tssovi/grokking-the-object-oriented-design-interview/blob/master/object-oriented-design-case-studies/design-a-movie-ticket-booking-system.md" \l "design-a-movie-ticket-booking-system)**

Our ticket booking service should meet the following requirements:

1. It should be able to list the cities where affiliate cinemas are located.
2. Each cinema can have multiple halls and each hall can run one movie show at a time.
3. Each Movie will have multiple shows.
4. Customers should be able to search movies by their title, language, genre, release date, and city name.
5. Once the customer selects a movie, the service should display the cinemas running that movie and its available shows.
6. The customer should be able to select a show at a particular cinema and book their tickets.
7. The service should show the customer the seating arrangement of the cinema hall. The customer should be able to select multiple seats according to their preference.
8. The customer should be able to distinguish between available seats and booked ones.
9. The system should send notifications whenever there is a new movie, as well as when a booking is made or canceled.
10. Customers of our system should be able to pay with credit cards or cash.
11. The system should ensure that no two customers can reserve the same seat.
12. Customers should be able to add a discount coupon to their payment.

enum AccountType {

CUSTOMER, BOOKING\_AGENT, MANAGER, ADMIN

}

enum SeatingStatus {

AVAILABLE, NOT\_AVAILABLE, BLOCKED, BOOKED, CANCELLED

}

Enum SeatType {

RECLINER, MIDDLE\_ROW, FRONT\_ROW, SIDE\_ROWS, SOFA

}

enum PaymentStatus{

PAID, REQUESTED, PENDING\_APPROVAL, NOT\_PAID, REJECTED, REFUNDED, CANCELLED

}

Enum ShowStatus {

SHOWING, AWAITING, HOUSEFULL

}

Enum PaymentMode {

CREDIT, DEBIT, CASH, PAYPAL, UPI

}

Class Cinema {

String title;

String language;

String genre;

DateTime releaseDate;

List<CinemaHall> listOfHalls;

}

Class CinemaHall {

List<Show> shows;

String cityName;

String CinemaHallID;

String cinemaHallName;

}

Class Show{

DateTime startTime;

DateTime endTime;

Cinema cinema;

List<Seat> listOfSeats;

ShowStatus status;

CinemaHall cinemaHall;

}

Class Seat{

SeatType seatType;

SeatingStatus seatStatus;

Long SeatNum;

Long cost;

}

Class Person {

String name;

String email;

String phoneNum;

Address address;

}

Class Account {

String accountNum;

String password;

Person personId;

AccountType accountType;

DateTime accountCreationTime;

List<Booking> bookingsList;

List<Booking> getListOfBookings() {

}

createBooking(CinemaHall hallId, String cinemaHallname, Show showT, Seat seat) {

if(seat.getSeatStatus == AVAILABLE) {

this.sendNotificationToUser();

Booking booking = new Booking ();

this.bookingList.add(booking);

seat.setStatus(BOOKED);

}

}

cancelBooking(String bookingId) {

this.sendNotificationToUser();

this.bookingList.remove(bookingId);

this.getBookingById(bookingId).setBookingStatus(CANCELLED);

}

}

Class Coupon {

String couponName;

String description;

Long percentage;

DateTime expiryDate;

Long validityDuration;

DateTime startDate;

}

Class Address{

String streetName;

String country;

String county;

String state;

String apartmentNum;

}

Class Booking {

String bookingId;

Cinema cinema;

CinemaHall cinemaHall;

BookingStatus bookingStatus;

Account accountId;

Bill receipt;

List<Seat> seatsList;

List<Coupon> couponsList;

}

Class Bill {

PaymentType paymentType;

PaymentStatus paymentStatus

Long bookingCost;

DateTime bookingTime;

Long tax;

String personName;

Address address;

}

Class TicketBookingSystem {

List<Cinema> cinemas;

List<CinemaHall> cinemaHalls;

List<Account> accounts;

List<Booking> allBookings;

Public TicketBookingSystem(){

Cinemas = ArrayList<>();

cinemaHalls = ArrayList<>();

accounts = ArrayList<>();

}

addNewCinema() {

for(int i=0; i<accounts.size();i++) {

this.account.sendNotifcationForNewMovie();

}

}

getCinemaHallsByCinemaName(String cinemaName) {

//query on cinemas with cinemaName;

}

getCinemaHallsByGenre(String genre) {

//query on cinemas with genre

}

getCinemasByCityName(String cityName) {

//query with city name

}

displayAvailableShowsByCinemaName(String cinemaName) {

}

}