educative(/learn)

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed

interview/g7Lw3O0A2Aj)

Sequence diagram
(/courses/grokking-the-object-
oriented-design-
interview/7nX38BMK9NO)

Activity Diagrams
(/courses/grokking-the-object-
oriented-design-
interview/B8RPL3VEl8N)

Object Oriented Design Case Studies

Design a Library Management System

Design a Hotel Management System

Let's design a hotel management system.

We'll cover the following ^

- System Requirements

- Use case diagram

- Class diagram

- Activity diagrams

- Code

A Hotel Management System is a software built to handle all online hotel activities easily and safely. This System will give the hotel management power and flexibility to manage the entire system from a single online portal. The system allows the manager to keep track of all the available rooms in the system as well as to book rooms and generate bills.

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed

Search Course

System Requirements

#

We'll focus on the following set of requirements while designing the Hotel Management System:

1. The system should support the booking of different room types like standard, deluxe, family suite, etc.

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed   ↻

🔍 Search Course

interview/g7Lw3O0A2Aj)

● Sequence diagram
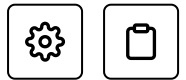(/courses/grokking-the-object-
oriented-design-
interview/7nX38BMK9NO)

● Activity Diagrams
(/courses/grokking-the-object-
oriented-design-
interview/B8RPL3VEl8N)

Object Oriented
Design Case Studies   ⌃

● Design a Library Management
System

2. Guests should be able to search the room inventory and book any available room.

3. The system should be able to retrieve information, such as who booked a particular room, or what rooms were booked by a specific customer.

4. The system should allow customers to cancel their booking - and provide them with a full refund if the cancelation occurs before 24 hours of the check-in date.

5. The system should be able to send notifications whenever the booking is nearing the check-in or check-out date.

6. The system should maintain a room housekeeping log to keep track of all housekeeping tasks.

7. Any customer should be able to add room services and food items.

8. Customers can ask for different amenities.

9. The customers should be able to pay their bills through credit card, check or cash.

Use case diagram
#

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed

## Object Oriented Design Case Studies

Here are the main Actors in our system:

- **Guest:** ~~All guests can search the available rooms, as well as~~ make a booking.
- **Receptionist:** Mainly responsible for adding and modifying rooms, creating room bookings, check-in, and check-out customers.
- **System:** Mainly responsible for sending notifications for room booking, cancellation, etc.
- **Manager:** Mainly responsible for adding new workers.
- **Housekeeper:** To add/modify housekeeping record of rooms.
- **Server:** To add/modify room service record of rooms.

Here are the top use cases of the Hotel Management System:

- **Add/Remove/Edit room:** To add, remove, or modify a room in the system.
- **Search room:** To search for rooms by type and availability.
- **Register or cancel an account:** To add a new member or cancel the membership of an existing member.
- **Book room:** To book a room.
- **Check-in:** To let the guest check-in for their booking.

- **Check-out:** To track the end of the booking and the return of the room keys.

- **Add room charge:** To add a room service charge to the customer's bill.

- **Update housekeeping log:** To add or update the housekeeping entry of a room.

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed   ↻

🔍 Search Course

interview/g7Lw3O0A2Aj)

🟢 Sequence diagram
(/courses/grokking-the-object-
oriented-design-
interview/7nX38BMK9NO)

🟢 Activity Diagrams
(/courses/grokking-the-object-
oriented-design-
interview/B8RPL3VEI8N)

Object Oriented
Design Case Studies   ⌃

🟢 Design a Library Management
System

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017)
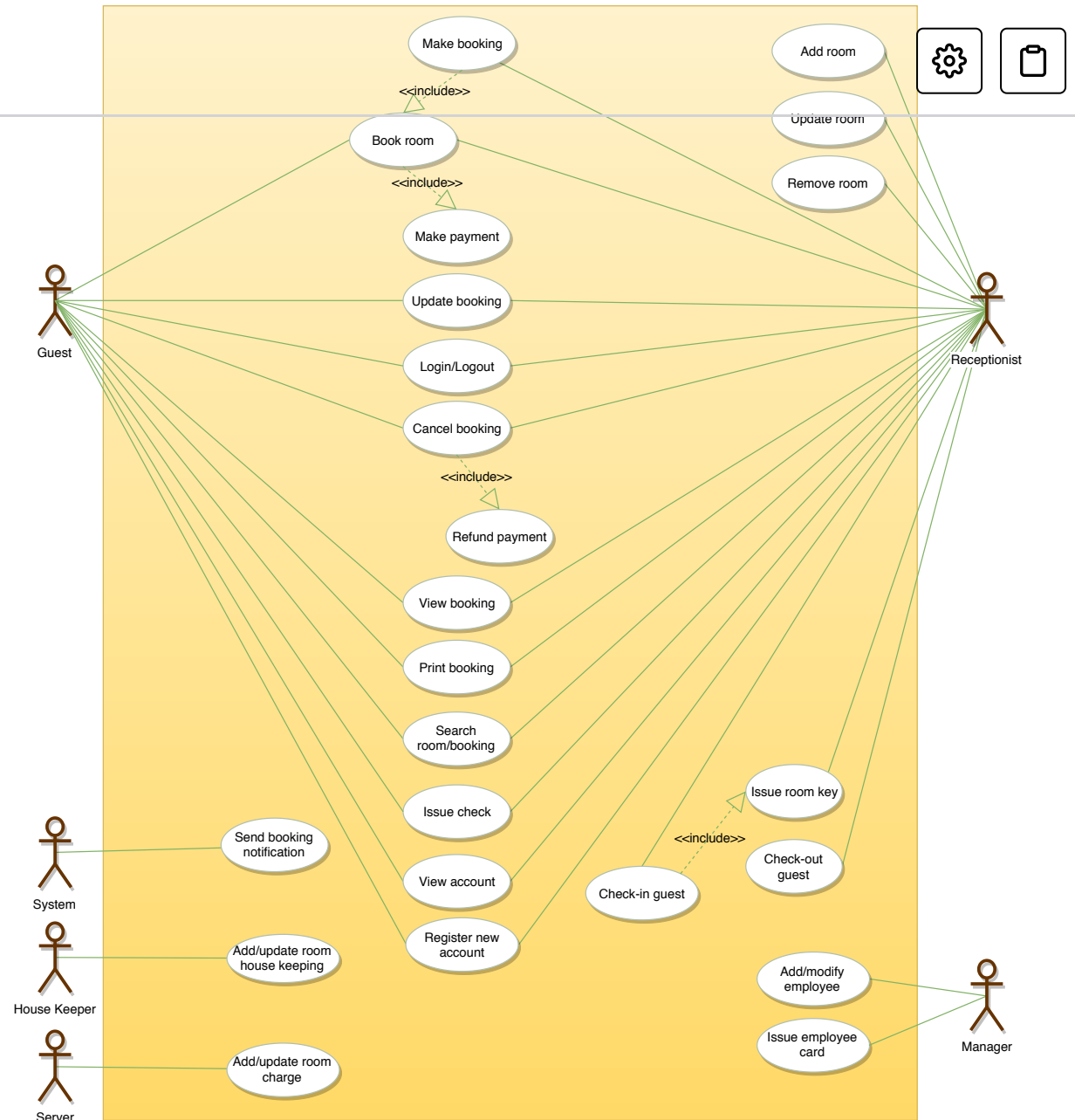
64% completed

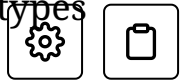Search Course

interview/g7Lw3O0A2Aj)

Sequence diagram
(/courses/grokking-the-object-
oriented-design-
interview/7nX38BMK9NO)

Activity Diagrams
(/courses/grokking-the-object-
oriented-design-
interview/B8RPL3VEl8N)

Object Oriented Design Case Studies

Design a Library Management System

Use case diagram

Grokking the Object Oriented Design Interview

64% completed

Search Course

Object Oriented Design Case Studies ^

Here are the main classes of our Hotel Management System:

- **Hotel and HotelLocation:** Our system will support multiple locations of a hotel.

- **Room:** The basic building block of the system. Every room will be uniquely identified by the room number. Each Room will have attributes like Room Style, Booking Price, etc.

- **Account:** We will have different types of accounts in the system: one will be a guest to search and book rooms, another will be a receptionist. Housekeeping will keep track of the housekeeping records of a room, and a Server will handle room service.

- **RoomBooking:** This class will be responsible for managing bookings for a room.

- **Notification:** Will take care of sending notifications to guests.

- **RoomHouseKeeping:** To keep track of all housekeeping records for rooms.

- **RoomCharge:** Encapsulates the details about different types of room services that guests have requested.

- **Invoice:** Contains different invoice-items for every charge against the room.

- **RoomKey:** Each room can be assigned an electronic key card. Keys will have a barcode and will be uniquely identified by a key-ID.

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed   ↺

🔍 Search Course

interview/g7Lw3O0A2Aj)

● Sequence diagram
(/courses/grokking-the-object-
oriented-design-
interview/7nX38BMK9NO)

● Activity Diagrams
(/courses/grokking-the-object-
oriented-design-
interview/B8RPL3VEI8N)

Object Oriented Design Case Studies   ⌃

● Design a Library Management System

educative(/learn)

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017
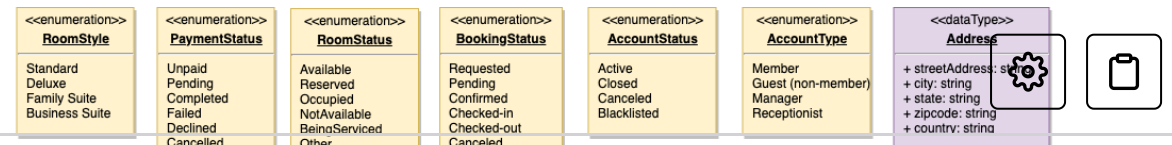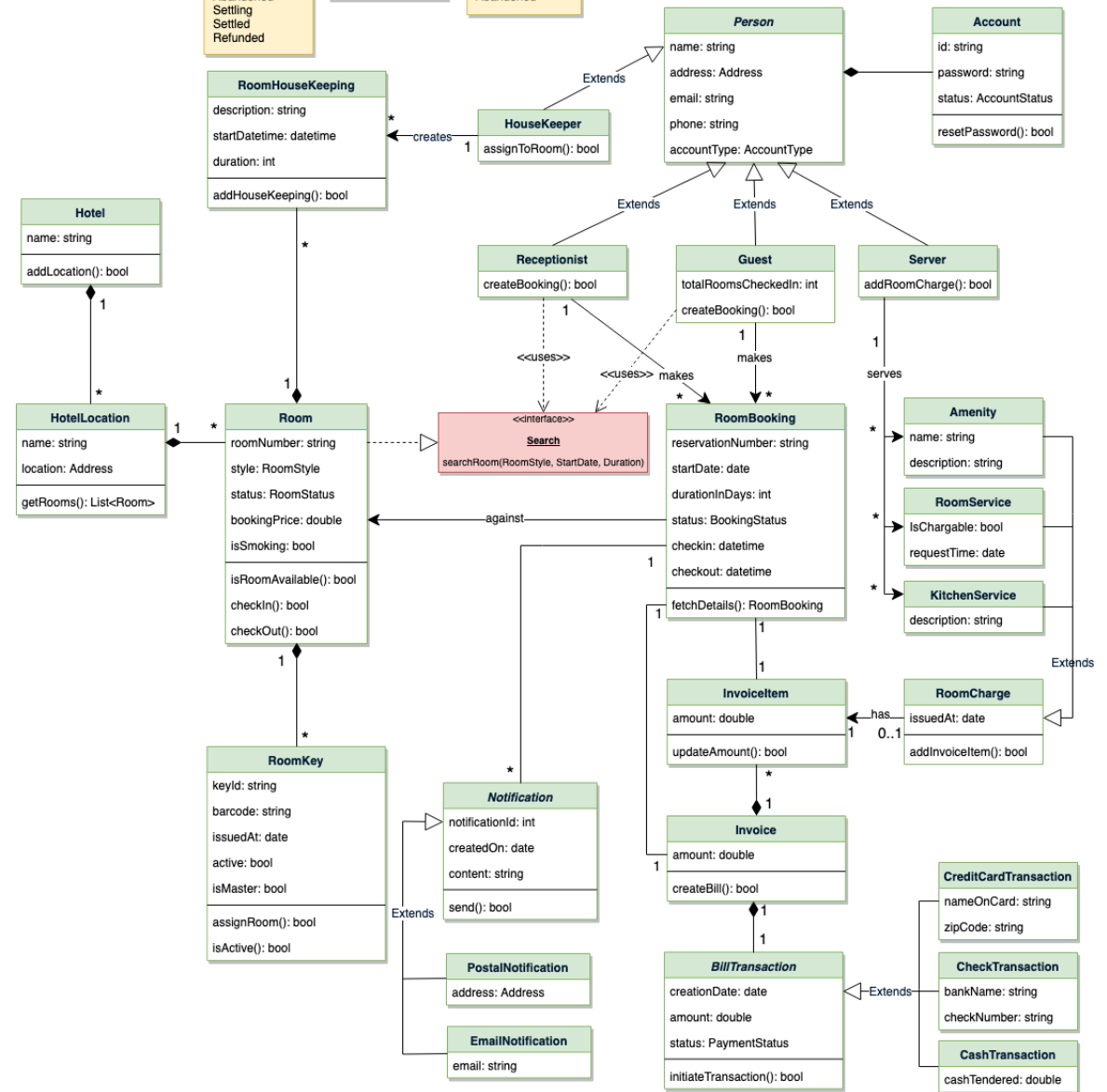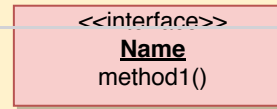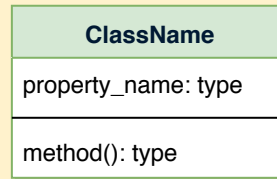
64% completed

interview/g7Lw3O0A2Aj)

- Sequence diagram (/courses/grokking-the-object-oriented-design-interview/7nX38BMK9NO)
- Activity Diagrams (/courses/grokking-the-object-oriented-design-interview/B8RPL3VEl8N)

Object Oriented Design Case Studies

- Design a Library Management System

Search Course

<<enumeration>> RoomStyle
Standard
Deluxe
Family Suite
Business Suite

<<enumeration>> PaymentStatus
Unpaid
Pending
Completed
Failed
Declined
Cancelled
Abandoned
Settling
Settled
Refunded

<<enumeration>> RoomStatus
Available
Reserved
Occupied
NotAvailable
BeingServiced
Other

<<enumeration>> BookingStatus
Requested
Pending
Confirmed
Checked-in
Checked-out
Canceled
Abandoned

<<enumeration>> AccountStatus
Active
Closed
Canceled
Blacklisted

<<enumeration>> AccountType
Member
Guest (non-member)
Manager
Receptionist

<<dataType>> Address
+ streetAddress: string
+ city: string
+ state: string
+ zipcode: string
+ country: string

Person
name: string
address: Address
email: string
phone: string
accountType: AccountType

Account
id: string
password: string
status: AccountStatus
resetPassword(): bool

RoomHouseKeeping
description: string
startDatetime: datetime
duration: int
addHouseKeeping(): bool

HouseKeeper
assignToRoom(): bool

Hotel
name: string
addLocation(): bool

Receptionist
createBooking(): bool

Guest
totalRoomsCheckedIn: int
createBooking(): bool

Server
addRoomCharge(): bool

HotelLocation
name: string
location: Address
getRooms(): List<Room>

Room
roomNumber: string
style: RoomStyle
status: RoomStatus
bookingPrice: double
isSmoking: bool
isRoomAvailable(): bool
checkIn(): bool
checkOut(): bool

<<interface>> Search
searchRoom(RoomStyle, StartDate, Duration)

RoomBooking
reservationNumber: string
startDate: date
durationInDays: int
status: BookingStatus
checkin: datetime
checkout: datetime
fetchDetails(): RoomBooking

Amenity
name: string
description: string

RoomService
IsChargable: bool
requestTime: date

KitchenService
description: string

RoomKey
keyId: string
barcode: string
issuedAt: date
active: bool
isMaster: bool
assignRoom(): bool
isActive(): bool

Notification
notificationId: int
createdOn: date
content: string
send(): bool

PostalNotification
address: Address

EmailNotification
email: string

InvoiceItem
amount: double
updateAmount(): bool

RoomCharge
issuedAt: date
addInvoiceItem(): bool

Invoice
amount: double
createBill(): bool

BillTransaction
creationDate: date
amount: double
status: PaymentStatus
initiateTransaction(): bool

CreditCardTransaction
nameOnCard: string
zipCode: string

CheckTransaction
bankName: string
checkNumber: string

CashTransaction
cashTendered: double

Class diagram

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

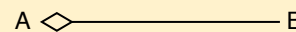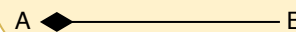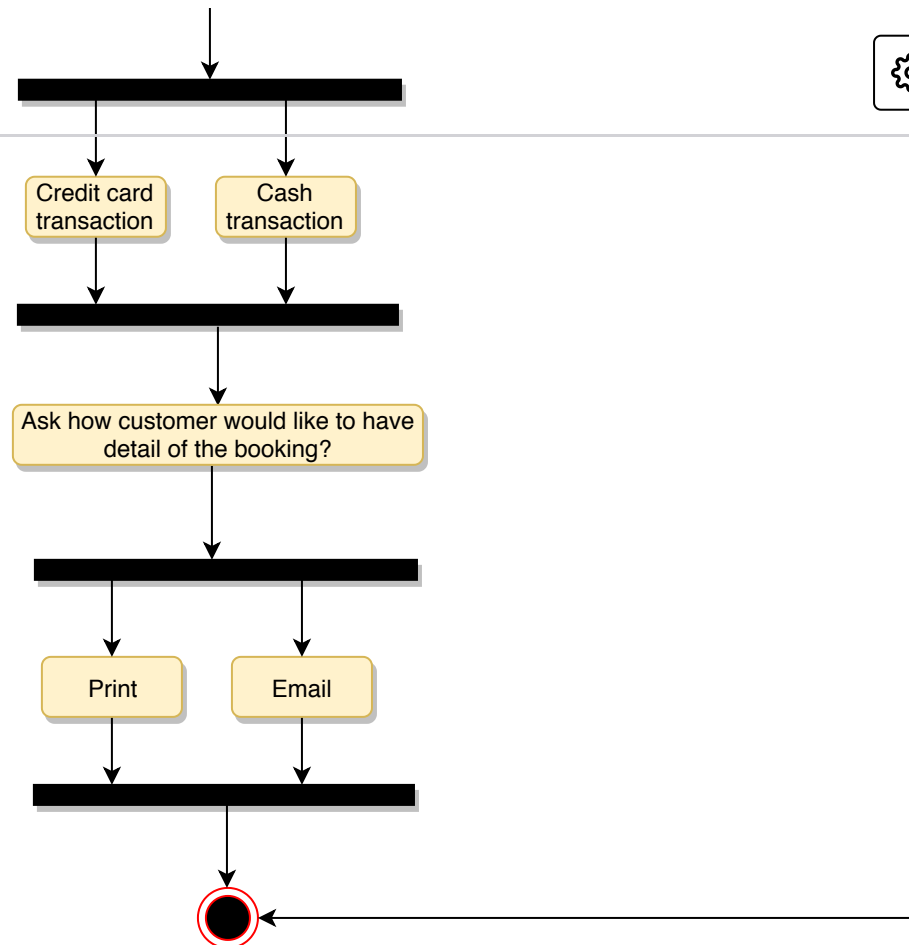64% completed ↻

🔍 Search Course

interview/g7Lw3O0A2Aj)

● Sequence diagram (/courses/grokking-the-object-oriented-design-interview/7nX38BMK9NO)

● Activity Diagrams (/courses/grokking-the-object-oriented-design-interview/B8RPL3VEl8N)

Object Oriented Design Case Studies ∧

● Design a Library Management System

UML conventions ⚙ 📋

Interface: Classes implement interfaces, denoted by Generalization.

Class: Every class can have properties and methods. Abstract classes are identified by their *Italic* names.
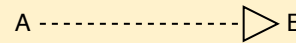
A - - - - - - - - ▷ B **Generalization**: A implements B.
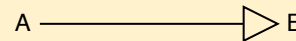
A ————————▷ B **Inheritance**: A inherits from B. A "is-a" B.
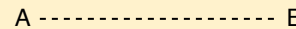
A - - - - - - - - - B **Use Interface:** A uses interface B.

A ———————— B **Association**: A and B call each other.

A ————————▶ B **Uni-directional Association**: A can call B, but not vice versa.

A ◇———————— B **Aggregation**: A "has-an" instance of B. B can exist without A.

A ◆———————— B **Composition**: A "has-an" instance of B. B cannot exist without A.

Activity diagrams

#

Make a room booking: Any guest or receptionist can perform this activity. Here are the set of steps to book a room:

educative(/learn)

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed

interview/g7Lw3O0A2Aj)

- Sequence diagram (/courses/grokking-the-object-oriented-design-interview/7nX38BMK9NO)
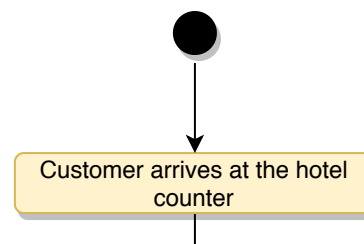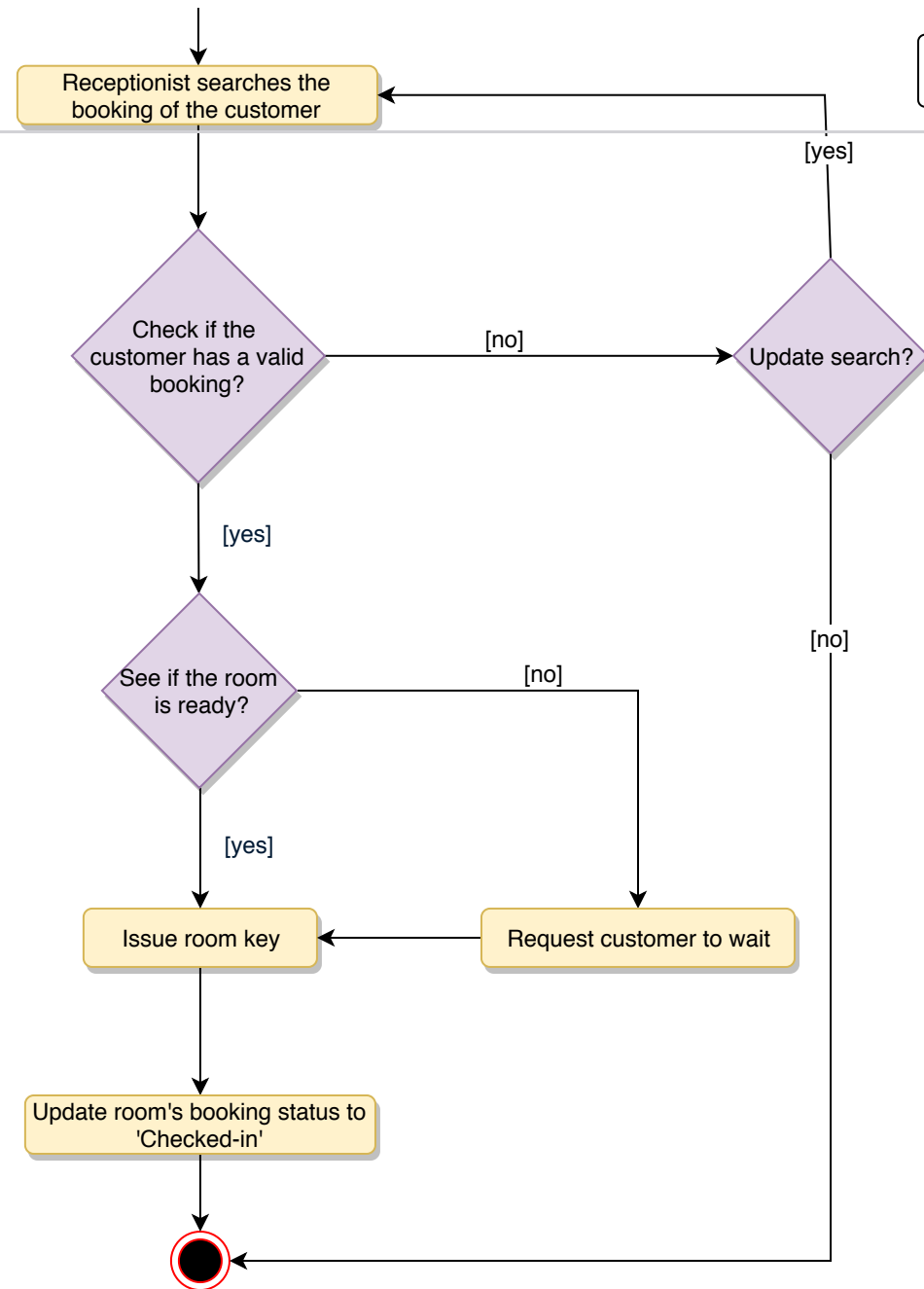
- Activity Diagrams (/courses/grokking-the-object-oriented-design-interview/B8RPL3VEI8N)

Object Oriented Design Case Studies

- Design a Library Management System

Search Course

Search Course

Check in: Guest will check in for their booking. The Receptionist can also perform this activity. Here are the steps:

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017
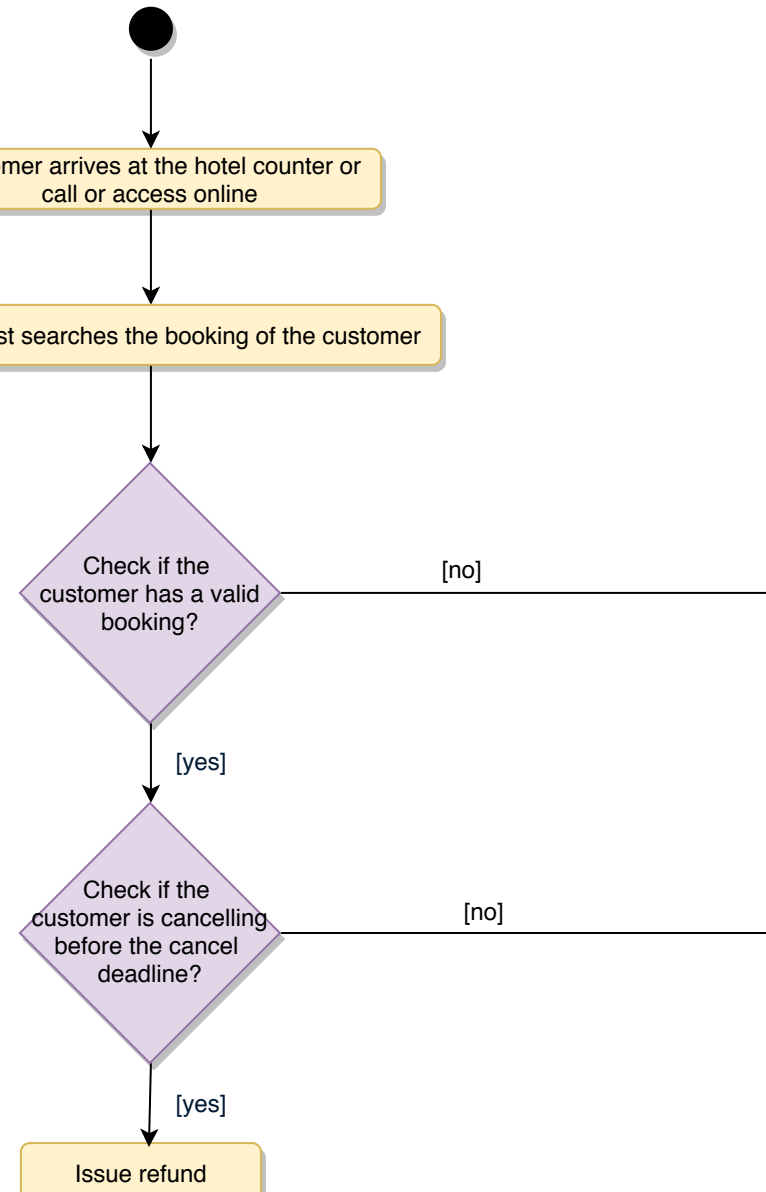
64% completed ↻

🔍 Search Course

interview/g7Lw3O0A2Aj)

🟢 Sequence diagram
(/courses/grokking-the-object-oriented-design-interview/7nX38BMK9NO)

🟢 Activity Diagrams
(/courses/grokking-the-object-oriented-design-interview/B8RPL3VEl8N)

Object Oriented
Design Case Studies ⌃

🟢 Design a Library Management
System

educative(/learn)

Grokking the Object Oriented Design Interview

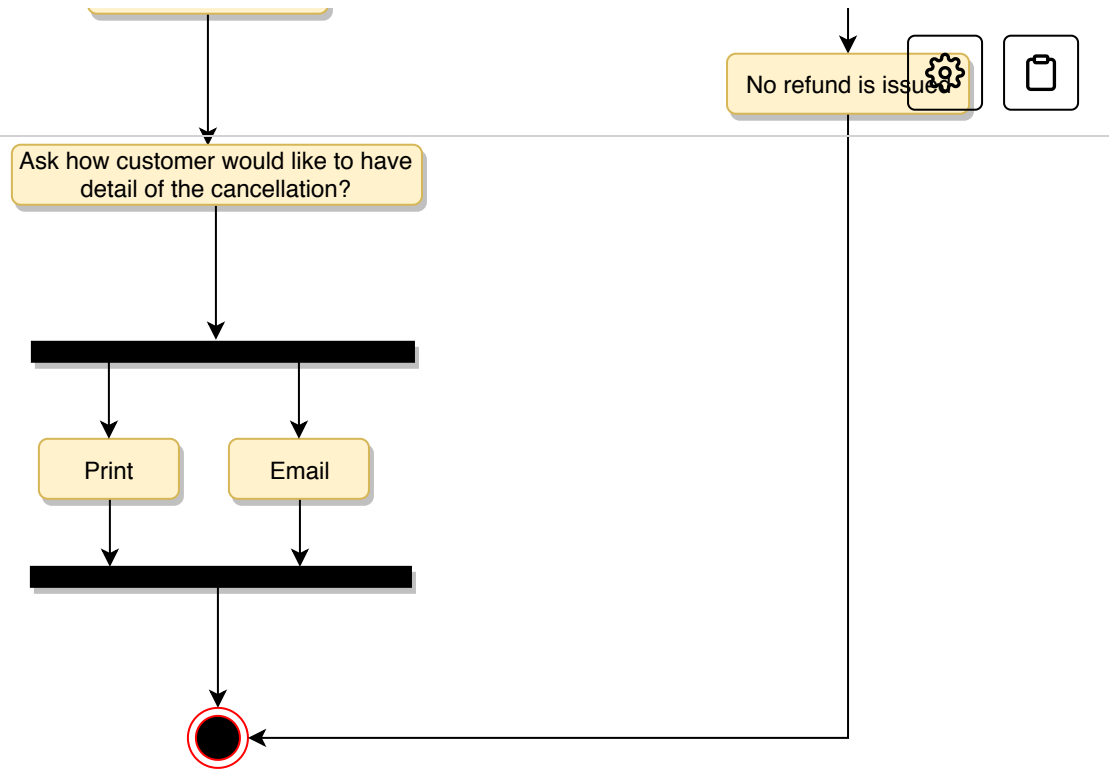(/collection/5668639101419520/56922017

64% completed

interview/g7Lw3O0A2Aj)

● Sequence diagram
(/courses/grokking-the-object-oriented-design-interview/7nX38BMK9NO)

● Activity Diagrams
(/courses/grokking-the-object-oriented-design-interview/B8RPL3VEl8N)

Object Oriented Design Case Studies

● Design a Library Management System

Cancel a booking: Guest can cancel their booking. Receptionist can perform this activity. Here are the different steps of this activity:

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed

Object Oriented Design Case Studies

No refund is issued

Ask how customer would like to have detail of the cancellation?

Print   Email

Code
#

Here is the high-level definition for the classes described above.

Enums, data types, and constants: Here are the required enums, data types, and constants:

Java	Python

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed

● Sequence diagram
(/courses/grokking-the-object-
oriented-design-
interview/7nX38BMK9NO)

● Activity Diagrams
(/courses/grokking-the-object-
oriented-design-
interview/B8RPL3VEI8N)

Object Oriented Design Case Studies ∧

● Design a Library Management System

```java
public enum RoomStyle {
    STANDARD, DELUXE, FAMILY_SUITE, BUSINESS_SUITE
}

public enum RoomStatus {
    AVAILABLE, RESERVED, OCCUPIED, NOT_AVAILABLE, BEING_SERVICED, OTHER
}

public enum BookingStatus {
    REQUESTED, PENDING, CONFIRMED, CHECKED_IN, CHECKED_OUT, CANCELLED, ABAN
}

public enum AccountStatus {
    ACTIVE, CLOSED, CANCELED, BLACKLISTED, BLOCKED
}

public enum AccountType {
    MEMBER, GUEST, MANAGER, RECEPTIONIST
}

public enum PaymentStatus {
    UNPAID, PENDING, COMPLETED, FILLED, DECLINED, CANCELLED, ABANDONED, SE
}

public class Address {
    private String streetAddress;
    private String city;
    private String state;
    private String zipCode;
    private String country;
}
```

Account, Person, Guest, Receptionist, and Server: These classes represent the different people that interact with our system:

☕ Java   🐍 Python

PK

educative(/learn)

Explore
(/explore)

Tracks
(/tracks)

My Courses
(/mycourses)

Edpresso
(/edpresso)

Refer a
Friend
(/refer-a-
friend)

Create

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed

```
// For simplicity, we are not defining getter and setter functions. The
// assume that all class attributes are private and accessed through the
// public getter method and modified only through their public setter me

public class Account {
  private String id;
  private String password;
  private AccountStatus status;

  public boolean resetPassword();
}

public abstract class Person {
  private String name;
  private Address address;
  private String email;
  private String phone;

  private Account account;
}

public class Guest extends Person {
  private int totalRoomsCheckedIn;

  public List<RoomBooking> getBookings();
}

public class Receptionist extends Person {
  public List<Member> searchMember(String name);
  public boolean createBooking();
}

public class Server extends Person {
  public boolean addRoomCharge(Room room, RoomCharge roomCharge);
}
```
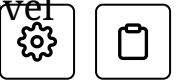
Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed

Search Course

interview/g7Lw3O0A2Aj)

● Sequence diagram
(/courses/grokking-the-object-
oriented-design-
interview/7nX38BMK9NO)

● Activity Diagrams
(/courses/grokking-the-object-
oriented-design-
interview/B8RPL3VEl8N)

Object Oriented Design Case Studies ⌃

● Design a Library Management
System

Hotel and HotelLocation: These classes represent the top-level classes of the system:

Java	Python

```java
public class HotelLocation {
  private String name;
  private Address location;

  public Address getRooms();
}

public class Hotel {
  private String name;
  private List<HotelLocation> locations;

  public boolean addLocation(HotelLocation location);
}
```
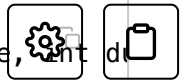
Room, RoomKey, and RoomHouseKeeping: To encapsulate a room, room key, and housekeeping:

Java	Python

educative(/learn)

Grokking the Object Oriented Design Interview

Search Course

Object Oriented
Design Case Studies

```java
public interface Search {
    public static List<Room> search(RoomStyle style, Date startDate, int du
}

public class Room implements Search {
    private String roomNumber;
    private RoomStyle style;
    private RoomStatus status;
    private double bookingPrice;
    private boolean isSmoking;

    private List<RoomKey> keys;
    private List<RoomHouseKeeping> houseKeepingLog;

    public boolean isRoomAvailable();
    public boolean checkIn();
    public boolean checkOut();

    public static List<Room> search(RoomStyle style, Date startDate, int du
        // return all rooms with the given style and availability
    }
}

public class RoomKey {
    private String keyId;
    private String barcode;
    private Date issuedAt;
    private boolean active;
    private boolean isMaster;

    public boolean assignRoom(Room room);
    public boolean isActive();
}

public class RoomHouseKeeping
 {
    private String description;
    private Date startDatetime;
    private int duration;
```

```
    private HouseKeeper houseKeeper;

    public boolean addHouseKeeping(Room room);
}
```

RoomBooking and RoomCharge: To encapsulate a booking and different charges against a booking:

☕ Java	🐍 Python

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed   ⟳

🔍 Search Course

interview/g7Lw3O0A2Aj)

● Sequence diagram
(/courses/grokking-the-object-
oriented-design-
interview/7nX38BMK9NO)

● Activity Diagrams
(/courses/grokking-the-object-
oriented-design-
interview/B8RPL3VEl8N)

Object Oriented
Design Case Studies   ⌃

● Design a Library Management
System

Grokking the Object Oriented Design Interview

64% completed   ↻

Object Oriented Design Case Studies   ︿

```
public class RoomBooking {
  private String reservationNumber;
  private Date startDate;
  private int durationInDays;
  private BookingStatus status;
  private Date checkin;
  private Date checkout;

  private int guestID;
  private Room room;
  private Invoice invoice;
  private List<Notification> notifications;

  public static RoomBooking fectchDetails(String reservationNumber);
}

public abstract class RoomCharge {
  public Date issueAt;
  public boolean addInvoiceItem(Invoice invoice);
}

public class Amenity extends RoomCharge {
  public String name;
  public String description;
}

public class RoomService extends RoomCharge {
  public boolean isChargeable;
  public Date requestTime;
}

public class KitchenService extends RoomCharge {
  public String description;
}
```

educative(/learn)

the-object-oriented-design-interview/YQ5gm2APRnp)
Design Blackjack and a Deck of Cards

the-object-oriented-design-interview/xV8p1GA6K0r)
Design a Restaurant Management sys...

Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017

64% completed

🔍 Search Course

interview/g7Lw3O0A2Aj)

● Sequence diagram (/courses/grokking-the-object-oriented-design-interview/7nX38BMK9NO)

● Activity Diagrams (/courses/grokking-the-object-oriented-design-interview/B8RPL3VEl8N)

Object Oriented Design Case Studies ⌃

● Design a Library Management System