



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



 Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies



Design a Library Management System

(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)

Design a Parking Lot

(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

## Design a Restaurant Management system

Let's design a restaurant management system.

We'll cover the following

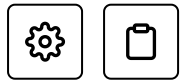


- System Requirements
- Use case diagram
- Class diagram
- Activity diagrams
- Code

A Restaurant Management System is a software built to handle all restaurant activities in an easy and safe manner. This System will give the Restaurant management power and flexibility to manage the entire system from a single portal. The system allows the manager to keep track of available tables in the system as well as

PK

the reservation of tables and bill generation.



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



Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)



## System Requirements

#

We will focus on the following set of requirements while designing the Restaurant Management System:

1. The restaurant will have different branches.
2. Each restaurant branch will have a menu.



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





 Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies



-  Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
-  Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

- The menu will have different menu sections, containing different menu items.
- The waiter should be able to create an order for a table and add meals for each seat.
- Each meal can have multiple meal items. Each meal item corresponds to a menu item.
- The system should be able to retrieve information about tables currently available to seat walk-in customers.
- The system should support the reservation of tables.
- The receptionist should be able to search for available tables by date/time and reserve a table.
- The system should allow customers to cancel their reservation.
- The system should be able to send notifications whenever the reservation time is approaching.
- The customers should be able to pay their bills through credit card, check or cash.



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





 Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies



-  Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
-  Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

12. Each restaurant branch can have multiple seating arrangements of tables.

## Use case diagram

#

Here are the main Actors in our system:

- **Receptionist:** Mainly responsible for adding and modifying tables and their layout, and creating and canceling table reservations.
- **Waiter:** To take/modify orders.
- **Manager:** Mainly responsible for adding new workers and modifying the menu.
- **Chef:** To view and work on an order.
- **Cashier:** To generate checks and process payments.
- **System:** Mainly responsible for sending notifications about table reservations, cancellations, etc.

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

- **Add/Modify tables:** To add, remove, or modify a table in the system.



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



 Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies



- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

- **Search tables:** To search for available tables for reservation.



- **Place order:** Add a new order in the system for a table.
- **Update order:** Modify an already placed order, which can include adding/modifying meals or meal items.
- **Create a reservation:** To create a table reservation for a certain date/time for an available table.
- **Cancel reservation:** To cancel an existing reservation.
- **Check-in:** To let the guest check in for their reservation.
- **Make payment:** Pay the check for the food.

# Grokking the Object Oriented Design Interview

(/collection/5668639101419520/56922017)

64% completed

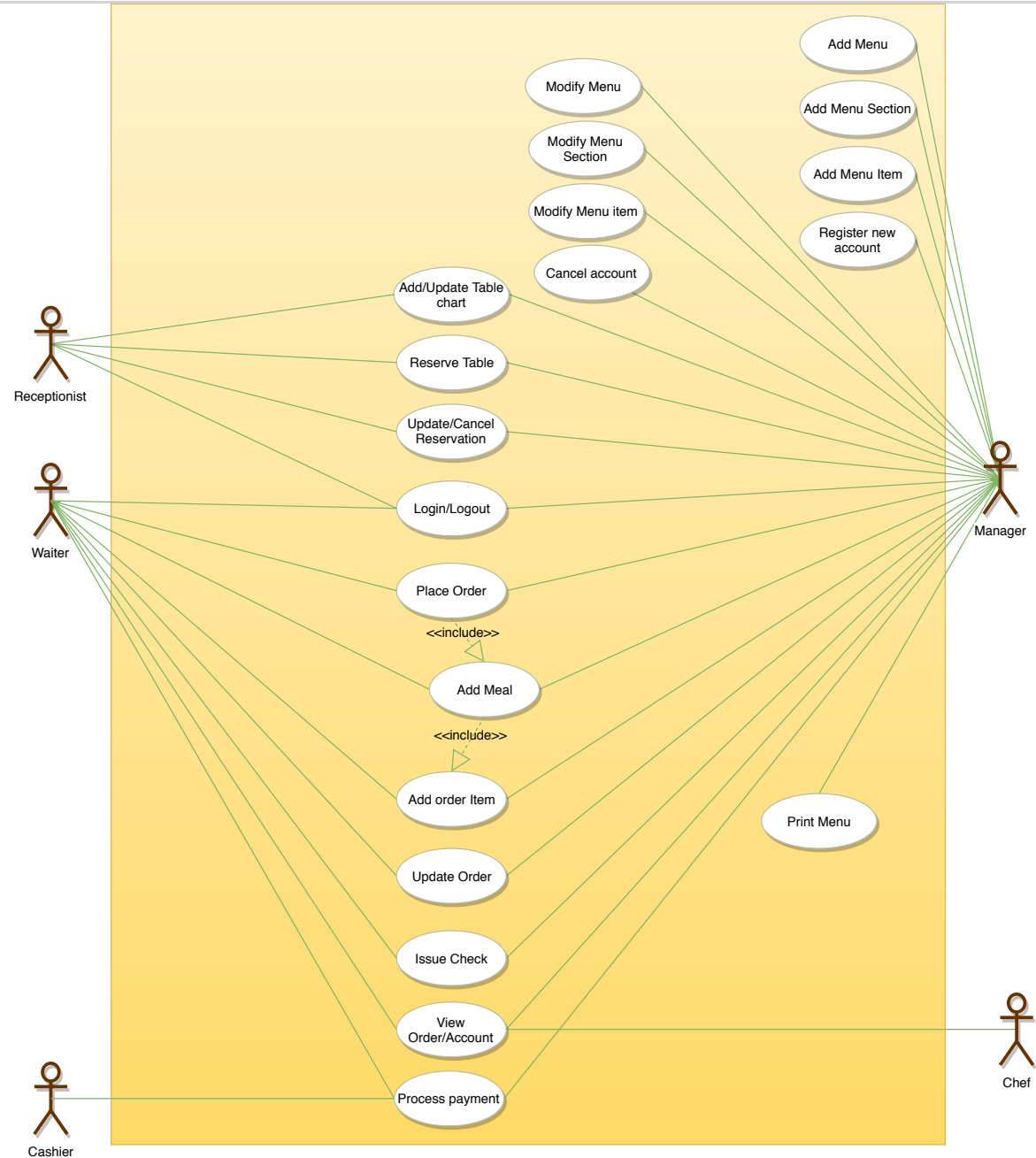


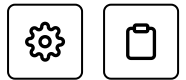
Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)





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



Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies



- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

## Class diagram

#

Here is the description of the different classes of our Restaurant Management System:

- **Restaurant:** This class represents a restaurant. Each restaurant has registered employees. The employees are part of the restaurant because if the restaurant becomes inactive, all its employees will automatically be deactivated.
- **Branch:** Any restaurants can have multiple branches. Each branch will have its own set of employees and menus.
- **Menu:** All branches will have their own menu.
- **MenuSection and MenuItem:** A menu has zero or more menu sections. Each menu section consists of zero or more menu items.
- **Table and TableSeat:** The basic building block of the system. Every table will have a unique identifier, maximum sitting capacity, etc. Each table will have multiple seats.
- **Order:** This class encapsulates the order placed by a





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



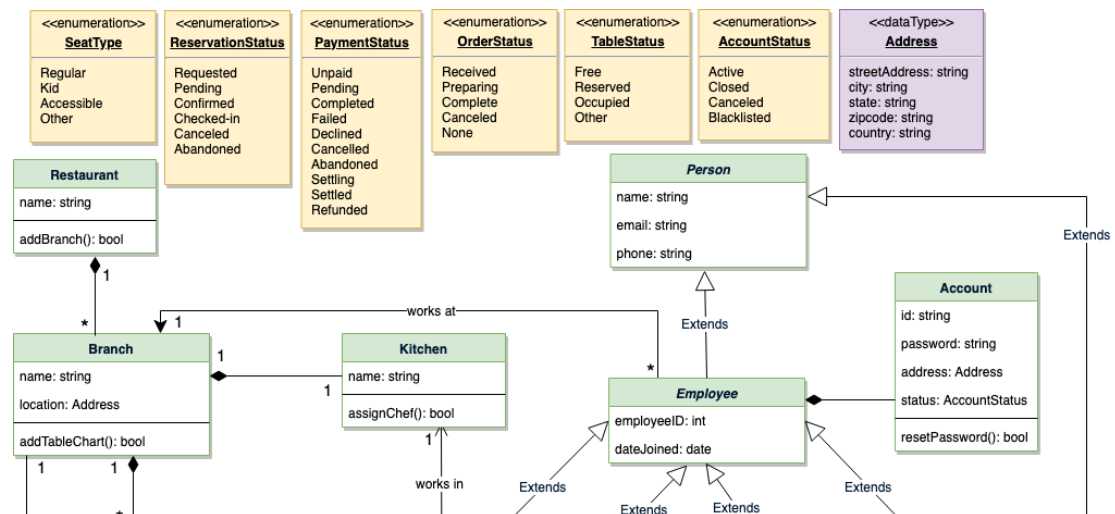
Q Search Course

oriented-design-  
interview/B8RPL3VEI8N)

# Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

- **Order:** This class encapsulates the order placed by a customer.
- **Meal:** Each order will consist of separate meals for each table seat.
- **Meal Item:** Each Meal will consist of one or more meal items corresponding to a menu item.
- **Account:** We'll have different types of accounts in the system, one will be a receptionist to search and reserve tables and the other, the waiter will place orders in the system.
- **Notification:** Will take care of sending notifications to customers.
- **Bill:** Contains different bill-items for every meal item.





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)

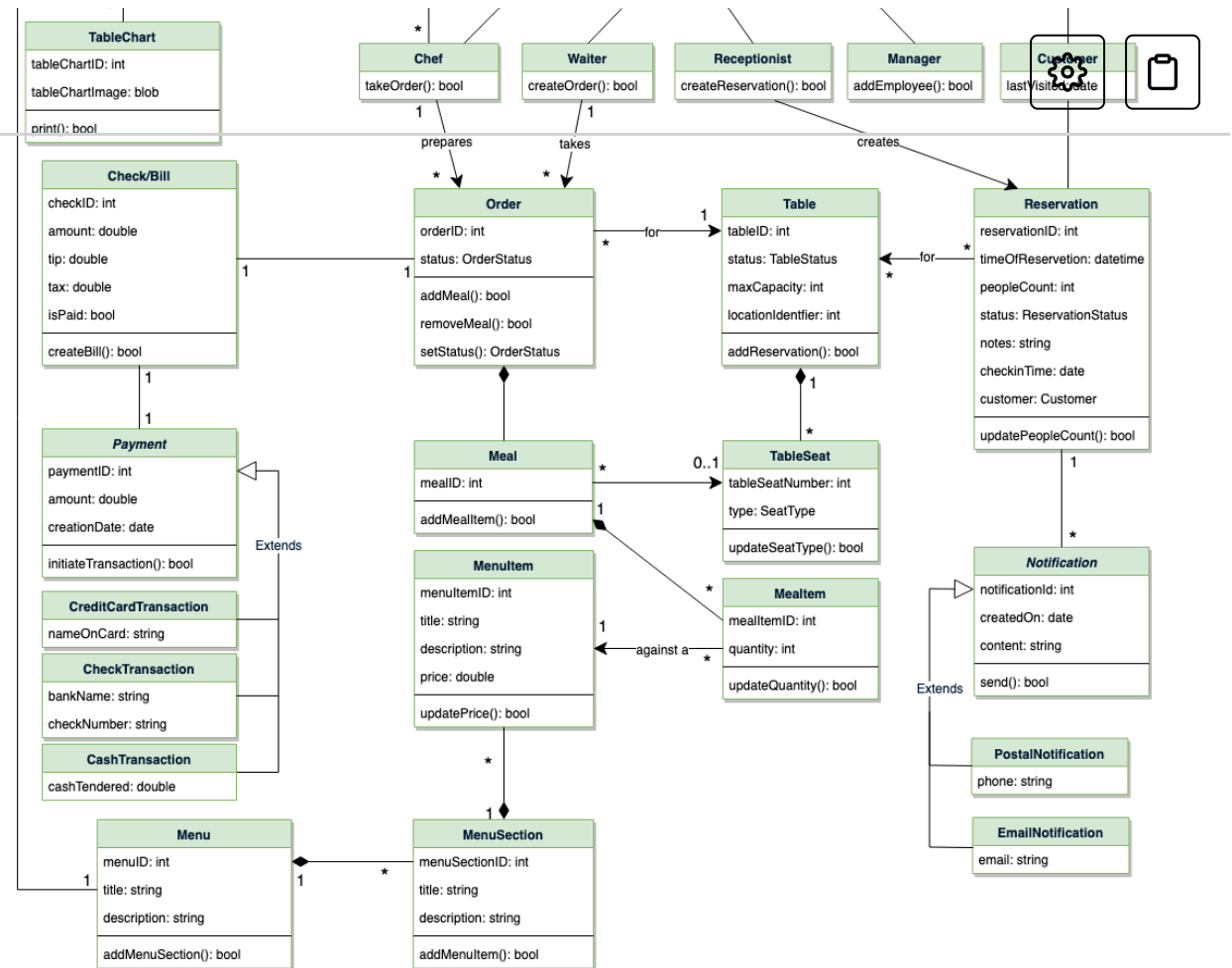
64% completed


 Search Course

 oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies

 Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)


 Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)



Class diagram

  
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



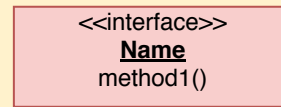
 Search Course

oriented-design-  
interview/B8RPL3VEI8N)

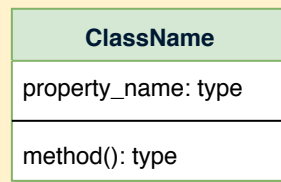
## Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

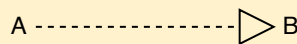
### UML conventions



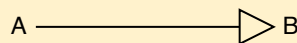
**Interface:** Classes implement interfaces, denoted by Generalization.



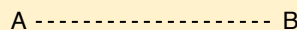
**Class:** Every class can have properties and methods.  
Abstract classes are identified by their *Italic* names.



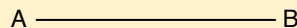
**Generalization:** A implements B.



**Inheritance:** A inherits from B. A "is-a" B.



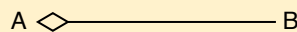
**Use Interface:** A uses interface B.



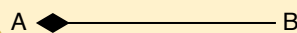
**Association:** A and B call each other.



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



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



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

### Activity diagrams

#

**Place order:** Any waiter can perform this activity. Here are the

PK



 **educative**(/learn)

steps to place an order:



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)

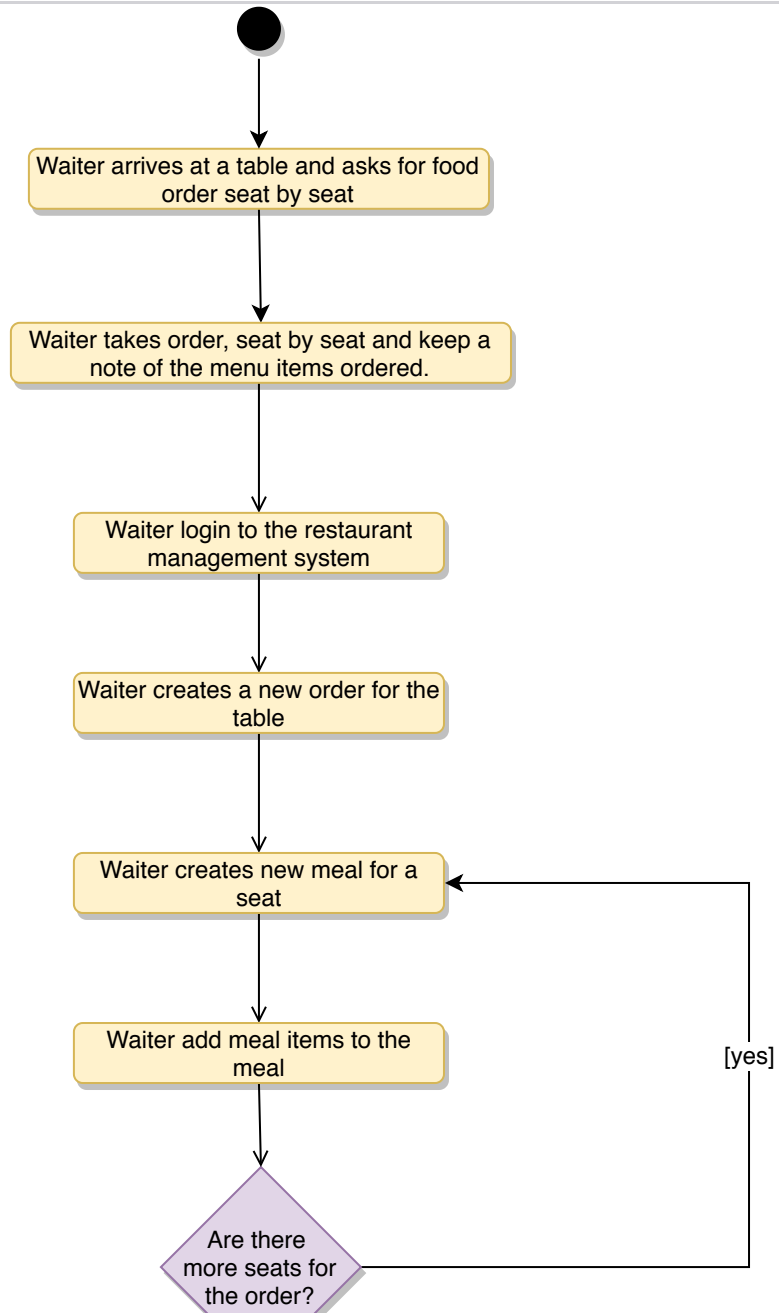
64% completed



oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)



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

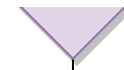


Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)



[no]

Send order to kitchen



**Make a reservation:** Any receptionist can perform this activity.  
Here are the steps to make a reservation:



Customer arrives at the restaurant counter  
or calls for table reservation

Customer asks for table reservation for a  
certain number of people and date/time

Receptionist searches tables for  
date/time and people count

Is table  
available?

[no]

Tell customer about the  
unavailability of table

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

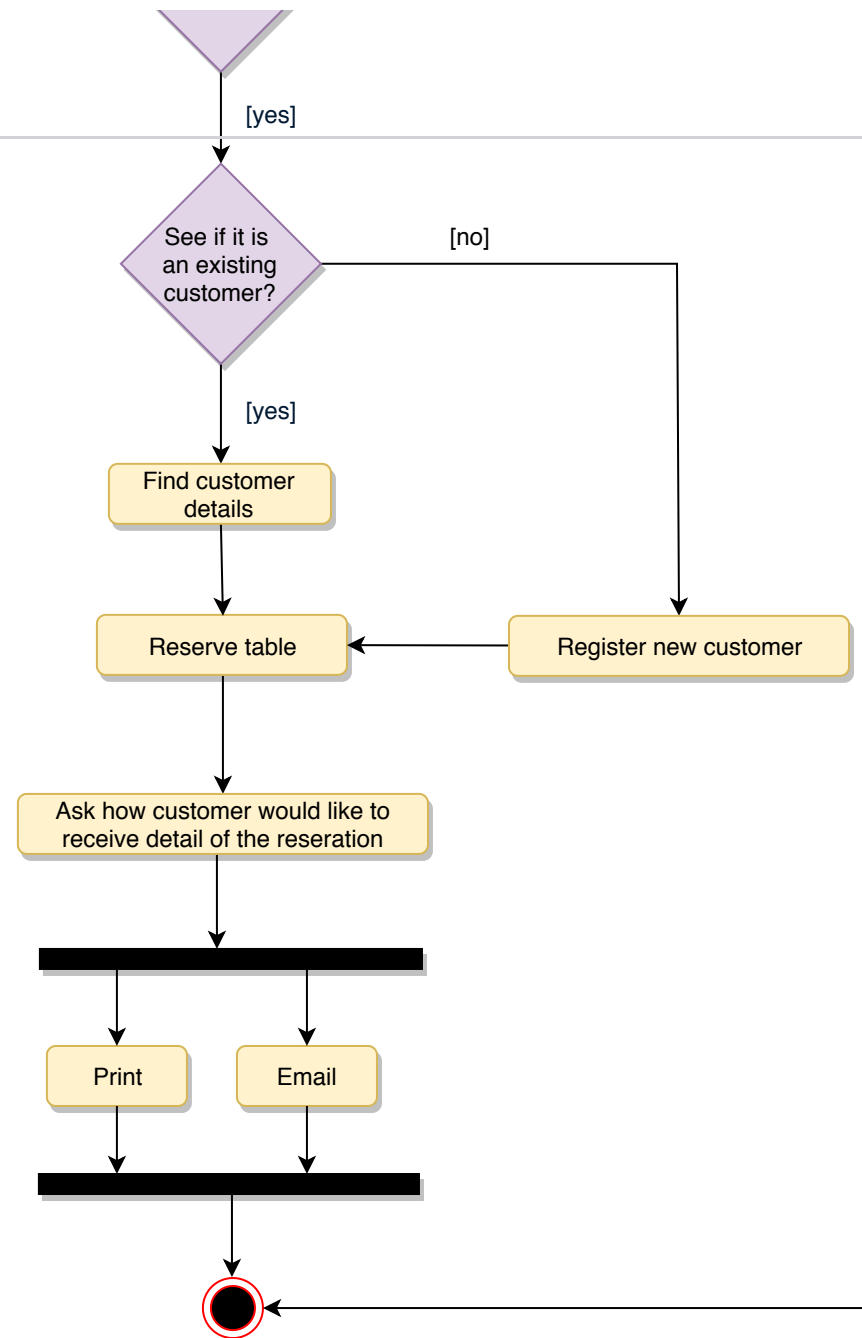


Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)



**Cancel a reservation:** Any receptionist can perform this activity.

PK



educative(/learn)


Here are the steps to cancel a reservation:




  
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

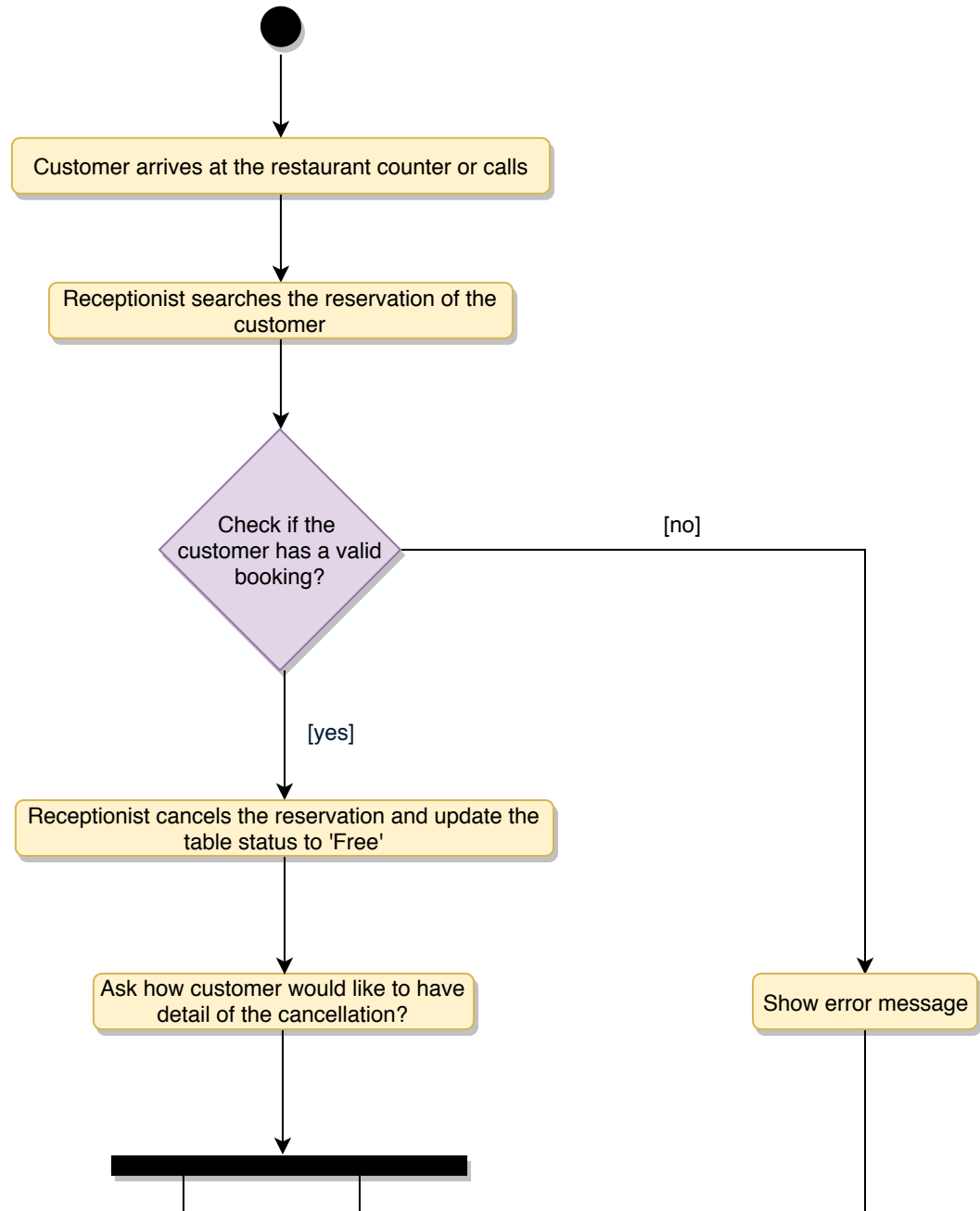
 Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies

Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)

Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)



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

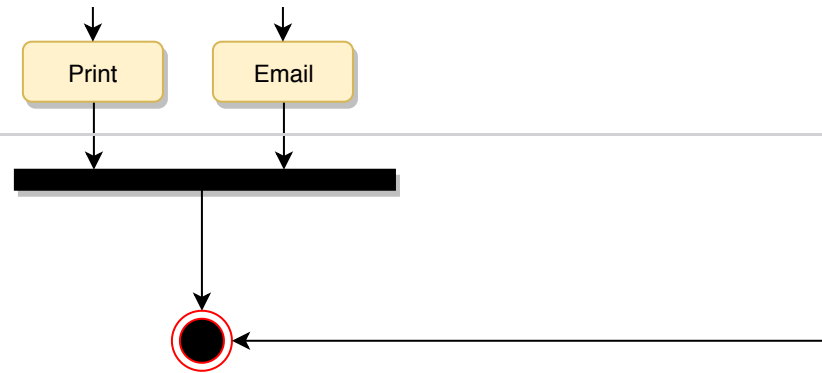


Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)



## 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

```

1 public enum ReservationStatus {
2     REQUESTED, PENDING, CONFIRMED, CHECKED_IN, CANCELED, ABANDONED
3 }
4
5 public enum SeatType {
6     REGULAR, KID, ACCESSIBLE, OTHER
7 }
8
9 public enum OrderStatus {
10    RECEIVED, PREPARING, COMPLETED, CANCELED, NONE
11 }
12
  
```



PK



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



 Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

```
--
13 public enum TableStatus {
14     FREE, RESERVED, OCCUPIED, OTHER
15 }
16
17 public enum AccountStatus {
18     ACTIVE, CLOSED, CANCELED, BLACKLISTED, BLOCKED
19 }
20
21 public enum PaymentStatus {
22     UNPAID, PENDING, COMPLETED, FILLED, DECLINED, CANCELLED, ABANDONED
23 }
24
25 public class Address {
26     private String streetAddress;
27     private String city;
28     private String state;
```



### Account, Person, Employee, Receptionist, Manager, and Chef:

These classes represent the different people that interact with our system:

 Java

 Python

```
1 // For simplicity, we are not defining getter and setter functions
2 // assume that all class attributes are private and accessed through
3 // public getter methods and modified only through their public set
4
5 public class Account {
6     private String id;
7     private String password;
8     private Address address;
9     private AccountStatus status;
10
```

PK



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



Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

```
11 public boolean resetPassword();
12 }
13
14 public abstract class Person {
15     private String name;
16     private String email;
17     private String phone;
18 }
19
20
21 public abstract class Employee extends Person {
22     private int employeeID;
23     private Date dateJoined;
24
25     private Account account;
26 }
27
28 public class Receptionist extends Employee {
```



**Restaurant, Branch, Kitchen, TableChart:** These classes represent the top-level classes of the system:

Java

Python

```
1 public class Kitchen {
2     private String name;
3     private Chef[] chefs;
4
5     private boolean assignChef();
6 }
7
8 public class Branch {
9     private String name;
```



PK

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

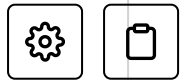
Search Course

oriented-design-  
interview/B8RPL3VEI8N)

### Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

```
10 private Address location;
11 private Kitchen kitchen;
12
13 public Address addTableChart();
14 }
15
16 public class Restaurant {
17     private String name;
18     private List<Branch> branches;
19
20     public boolean addBranch(Branch branch);
21 }
22
23 public class TableChart {
24     private int tableChartID;
25     private byte[] tableChartImage;
26
27     public bool print();
28 }
```



**Table, TableSeat, and Reservation:** Each table can have multiple seats and customers can make reservations for tables:

Java

Python

```
1 public class Table {
2     private int tableID;
3     private TableStatus status;
4     private int maxCapacity;
5     private int locationIdentifier;
6
7     private List<TableSeat> seats;
8
9     public boolean isTableFree() {
```



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



Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies

- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

```
9 public boolean isTableFree(),
10 public boolean addReservation();
11
12 public static List<Table> search(int capacity, Date startTime) {
13     // return all tables with the given capacity and availability
14 }
15 }
16
17 public class TableSeat {
18     private int tableSeatNumber;
19     private SeatType type;
20
21     public boolean updateSeatType(SeatType type);
22 }
23
24 public class Reservation {
25     private int reservationID;
26     private Date timeOfReservation;
27     private int peopleCount;
28     private ReservationStatus status;
```



**Menu, MenuSection, and MenuItem:** Each restaurant branch will have its own menu, each menu will have multiple menu sections, which will contain menu items:



Java



Python



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



 Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies



- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

```
public class MenuItem {
    private int menuItemID;
    private String title;
    private String description;
    private double price;

    public boolean updatePrice(double price);
}

public class MenuSection {
    private int menuSectionID;
    private String title;
    private String description;
    private List<MenuItem> menuItems;

    public boolean addMenuItem(MenuItem menuItem);
}

public class Menu {
    private int menuID;
    private String title;
    private String description;
    private List<MenuSection> menuSections;

    public boolean addMenuSection(MenuSection menuSection);
    public boolean print();
}
```



**Order, Meal, and MenuItem:** Each order will have meals for table seats:



Java



Python



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



 Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies



- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

```
public class MenuItem {
    private int menuItemID;
    private int quantity;
    private MenuItem menuItem;

    public boolean updateQuantity(int quantity);
}

public class Meal {
    private int mealID;
    private TableSeat seat;
    private List<MenuItem> menuItems;

    public boolean addMealItem(MealItem mealItem);
}

public class Order {
    private int OrderID;
    private OrderStatus status;
    private Date creationTime;

    private Meal[] meals;
    private Table table;
    private Check check;
    private Waiter waiter;
    private Chef chef;

    public boolean addMeal(Meal meal);
    public boolean removeMeal(Meal meal);
    public OrderStatus getStatus();
    public boolean setStatus(OrderStatus status);
}
```



← Back

(/courses/grokking-

Next →

(/courses/grokking-

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



 Search Course

oriented-design-  
interview/B8RPL3VEI8N)

## Object Oriented Design Case Studies



- Design a Library Management System  
(/courses/grokking-the-object-oriented-design-interview/RMIM3NgjAyR)
- Design a Parking Lot  
(/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

/courses/grokking

the-  
object-

oriented-

Design a Hotel Management System

interview/39Ek39vZBy9)



Report  
an Issue



Ask a Question

(https://discuss.educative.io/c/grokking-the-object-oriented-design-interview-design-gurus/object-oriented-design-case-studies-design-a-restaurant-management-system)



COMPLETED



object-

oriented-

Design a Chess

interview/JP7BXYkj3DK)