# Getting Ready: The Hotel Management System

Understand the hotel management system problem and learn the questions to further simplify this problem.

**We'll cover the following**

* [Problem definition](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Problem-definition)
* [Expectations from the interviewee](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Expectations-from-the-interviewee)
  + [Room booking](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Room-booking)
  + [Payment handling](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Payment-handling)
  + [Price variance](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Price-variance)
  + [Booking cancelation](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Booking-cancelation)
* [Design approach](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Design-approach)
* [Design pattern](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Design-pattern)

## Problem definition

The **hotel management system** is a software used to manage all hotel activities efficiently and smoothly. Almost all popular hotels have an online management system to digitize the process of booking rooms, managing staff, and performing other necessary hotel management features. Through this system, the notification and payment process is made flexible and automated. In the hotel, there is usually a fixed number of rooms, and the booking system has information about all rooms present and their availability.

For the hotel manager, controlling all the hotel activities manually is a difficult task. With the help of an online hotel management system, the manager can keep track of rooms, customers, and workers through a single portal. The manager can also post available rooms to the system and generate bills. The system also allows customers to search for and book a room of their choice and cost range. The system provides complete information about rooms to the customers so they can view the available rooms and book them online. The guest is charged on the basis of the time duration the hotel room is booked. In this way, the system facilitates both customers and the manager of the hotel.

## Expectations from the interviewee

Numerous components are present in the hotel management system, each with specific constraints and requirements placed on them. The following provides an overview of some of the main things that the interviewer will want to hear you discuss in more detail during the interview.

### Room booking

Booking a room is an essential part of the hotel management system. The system has to ensure that no two people can book the same room. The interviewer expects you to ask questions to identify how the system will work in such situations:

* How will the system ensure that multiple users do not book the same room?
* What type of users are allowed to book a room in the hotel?
* Can users book a room in advance?

### Payment handling

One of the hotel management system's most significant attributes is its payment structure to its customers. This can vary, so the interviewer would expect you to ask the questions listed below:

* What payment methods can the customer use (for example, credit card or cash)?
* How is the payment performed? Does the customer pay online or through a receptionist at the hotel?
* Will the customer be able to pay in advance for a room booking, or is a just-in-time(JIT) payment system available?

### Price variance

We touched upon the payment methods of the hotel management system. Now, the pricing model needs to be clarified by the interviewer. Therefore, you may ask the questions listed below:

* How will the booking price be calculated? Or which factors affect the price of a room in the hotel?
* How does the location and size of the room affect its price?
* How does the booking duration affect the payment?

### Booking cancelation

There will be many duplicate instances in our system. The interviewer expects you to ask the questions provided below:

* Can the user cancel a room booking?
* Which type of users are allowed to request a room booking cancelation?

## Design approach

We’ll design this hotel management system using the bottom-up design approach. For this purpose, we will follow the steps below:

* Identify and design the smallest components first, such as a room.
* Use these small components to design bigger components, such as building a hotel that can be composed of multiple rooms.
* Repeat the steps above until we design the whole hotel management system.

## Design pattern

It is always a good practice to discuss the design patterns that a hotel management system falls under, during the interview. Stating the design patterns will give the interviewer a positive impression and shows that the interviewee is well-versed in the advanced concepts of object-oriented design.

The following design patterns can be used to design the hotel management system:

* Strategy design pattern
* Singleton design pattern
* Factory design pattern

Let’s explore the requirements of the hotel management system in the next lesson.

# Requirements for the Hotel Management System

Learn about all the requirements of the hotel management system problem.

**We'll cover the following**

* [Requirement collection](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Requirement-collection)

In this lesson, we’ll list the requirements of the hotel management system. This is a very crucial step since requirements define the scope of a problem, so getting them right from the interviewer and understanding them well will make the design of the rest of the system smooth and easy.

We’ll use the notational convention to identify each requirement with a unique label "Rn", where "R" is short for Requirement and "n" is a natural number.

## Requirement collection[#](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Requirement-collection)

The requirements for the hotel management system are defined below:

**R1:** There can be four types of accounts in the system such as housekeeper, receptionist, guest, or server.

**R2:** The rooms can be of different styles like standard, deluxe, family suite, or business suite.

**R3:** The system should allow the guests to search for any room and book any of the available rooms.

**R4:** During room booking, the user will enter the check-in date and the duration of the stay. The user would also have to give some advance payment.

**R5:** The customer can cancel the booking and a full refund will be provided if the booking is canceled before 24 hours of the check-in time.

**R6:** The system should send a notification to the customers about the booking status or other information.

**R7:** All the housekeeping tasks should be logged in and managed by the system.

**R8:** The system should allow the customer to add services of their own choice like room service, food or kitchen service, or amenity.

**R9:** Every room should have its own specific key, and there can be a master key that opens a specific set of rooms.

**R10:** A hotel can have multiple branches of it.

We've identified our requirements for the problem, and in the next lesson, we will define different use cases for the hotel management system.

# Use Case Diagram for the Hotel Management System

Learn how to define use cases and create the corresponding use case diagram for the hotel management system.

**We'll cover the following**

* [System](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#System)
* [Actors](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Actors)
  + [Primary actors](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Primary-actors)
  + [Secondary actors](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Secondary-actors)
* [Use cases](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Use-cases)
  + [Guest](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Guest)
  + [Receptionist](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Receptionist)
  + [Manager](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Manager)
  + [System](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#System)
  + [Housekeeper](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Housekeeper)
* [Relationships](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Relationships)
  + [Generalization](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Generalization)
  + [Associations](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Associations)
  + [Include](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Include)
* [Use case diagram](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Use-case-diagram)

Let’s build the use case diagram for the hotel management system and understand the relationship between its different components.

First, we’ll define the different elements of our hotel, followed by the complete use case diagram of the system.

## System

Our system is a "hotel."

## Actors

Now, we will define the main actors of our hotel management system.

### Primary actors

* **Guest:**This is the hotel's primary actor who can book a room, make payment, and change or cancel the reservations.
* **Receptionist:**This actor acts as the admin of the system and can perform any task a “Guest” can perform. This can also add, remove, or update the room, check in/check out guests, and issue room keys for guests.
* **Manager:**This actor can add or modify an employee and issue a card to an employee of the hotel.
* **Housekeeper:** This can add or update the room’s housekeeping status.

### Secondary actors

* **System:**This can send booking notifications to guests.
* **Server:**This can add or update room status according to the room change request.

## Use cases

In this section, we will define the use cases for the hotel. We have listed the use cases according to their respective interactions with a particular actor.

**Note:**You will see some use cases occurring multiple times because they are shared among different actors in the system.

### Guest

* **Book room:**To book a room in the hotel
* **Update booking:**To update a room booking in the hotel
* **Login/Logout:**To log in and out of the hotel management system
* **Cancel booking:** To cancel a room booking in the hotel
* **View booking:**To view and verify a room booking
* **Print booking:**To print booking details from the hotel management system
* **Search room/booking:** To search for a room or a booking in the hotel management system
* **Payment:**To pay the room rent to the hotel
* **View account:**To view account details and booking status
* **Register new account:**To register a new account for new guests
* **Return room key:** To return the room key before checkout

### Receptionist

* **Add room:**To add rooms to the hotel management system so guests can book them
* **Update room:**To update room status from available to booked or vice versa
* **Remove room:**To remove a room from the hotel management system so guests can't book it
* **Book room:**To book a room in the hotel
* **Update booking:**To update a room booking in the hotel
* **Login/Logout:**To log in and out of the hotel management system
* **Cancel booking:** To cancel a room booking in the hotel
* **View booking:**To view and verify a room booking
* **Print booking:**To print booking details from the hotel management system
* **Search room/booking:** To search for a room or a booking in the hotel management system
* **View account:**To view account details and booking status
* **Register new account:**To register a new account for new guests
* **Check in guest:**To check in guests to the hotel
* **Check out guest:**To check out guests from the hotel
* **Issue room key:**To issue room keys to guests who checked in

### Manager

* **Issue employee card:** To issue employee cards so one can be identified
* **Add/modify employee:** To add a new employee to the hotel management system or to modify the employee's status
* **Add room:**To add rooms to the hotel management system so guests can book them
* **Update room:**To update room status from available to booked or vice versa
* **Remove room:**To remove a room from the hotel management system so guests can't book it
* **Book room:**To book a room in the hotel
* **Update booking:**To update a room booking in the hotel
* **Login/Logout:**To log in and out of the hotel management system
* **Cancel booking:** To cancel a room booking in the hotel
* **View booking:**To view and verify a room booking.
* **Print booking:**To print the booking details from the hotel management system
* **Search room/booking:** To search for a room or a booking in the hotel management system
* **View account:**To view account details and booking status
* **Register new account:**To register a new account for new guests
* **Check in guest:**To check in guests to the hotel
* **Check out guest:**To check out guests from the hotel
* **Issue room key:**To issue room keys to guests who checked in

### System

* **Add/update room charge:**To update the status of the room charge
* **Send booking notification:**To send booking notification to guests

### Housekeeper

* **Add/update room housekeeping:** To update the housekeeping status of rooms

## Relationships

We describe the relationships between and among actors and their use cases in this section.

### Generalization

* The manager is the boss of the receptionist and has access to everything a receptionist has. Therefore, “Manager” has a generalization relationship with “Receptionist.”
* “Cash” and “Credit card” use cases are used for payments. Hence, both have a generalization relationship with the “Pay ticket” use case.

### Associations

The below table shows the association relationship between actors and their use cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Guest** | **Receptionist** | **Manager** | **System** | **Housekeeper** |
| Book room | Book room | Issue employee card | Send booking notification | Add/update room housekeeping |
| Payment | View account | Add/modify employee | Add/update room charge |  |
| View account | Register new account | Book room |  |
| Register new account | Print booking | View account |
| Print booking | Cancel booking | Register new account |
| Cancel booking | Login/Logout | Print booking |
| Login/Logout | Check in guest | Cancel booking |
| Search room/booking | issue room key | Login/Logout |
| Update booking | Search room/booking | Check-in guest |
| View booking | Update booking | Issue room key |
| Return room key | Check out guest | Search room/booking |
|  | View booking | Update booking |
| Add room | Check-out guest |
| Remove room | View booking |
| Update room | Add room |
|  | Remove room |
| Update room |

### Include

* Whenever a guest books a room, the payment will be processed. Hence, the “Book room” use case has an include relationship with “Payment.”
* When a receptionist checks in a guest, a key is issued to the guest. Hence, “Check-in guest” has an include relationship with “Issue room key.”
* When a guest checks out, the key is returned to the receptionist. Hence, “Check-out guest” has an include relationship with “Return room key.”
* If a booking is canceled, the payment will be refunded. Hence the “Cancel booking” use case has an include relationship with “Refund payment.”

## Use case diagram

Here’s the use case diagram of the hotel management system:

A diagram of a hotel management system

Description automatically generated

In the next lesson, we’ll discuss the class diagram with a detailed explanation of all classes and their relationship with each other.

# Class Diagram for the Hotel Management System

Learn to create a class diagram for the hotel management system problem using the bottom-up approach.

**We'll cover the following**

* [Components of a hotel management system](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Components-of-a-hotel-management-system)
  + [Address and Account](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Address-and-Account)
  + [Person](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Person)
  + [Service](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Service)
  + [Invoice](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Invoice)
  + [Room booking](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Room-booking)
  + [Notification](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Notification)
  + [Room, room key, and room housekeeping](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Room,-room-key,-and-room-housekeeping)
  + [Search interface and catalog](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Search-interface-and-catalog)
  + [Bill transaction](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Bill-transaction)
  + [Hotel and hotel branch](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Hotel-and-hotel-branch)
  + [Enumerations](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Enumerations)
* [Relationship between the classes](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Relationship-between-the-classes)
  + [Association](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Association)
    - [One-way association](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#One-way-association)
    - [Two-way association](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Two-way-association)
  + [Aggregation](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Aggregation)
  + [Composition](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Composition)
  + [Inheritance](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Inheritance)
* [Class diagram of the hotel management system](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Class-diagram-of-the-hotel-management-system)
* [Design pattern](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Design-pattern)
* [Additional requirements](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Additional-requirements)

Here, we are going to create the class diagram for our system on the basis of requirements that we gathered in one of the previous lessons. In the class diagram, we will first design and create the classes, abstract classes, and interfaces for the system, and then we’ll identify the relationship between classes in accordance with all the requirements of the hotel management system.

## Components of a hotel management system

In this section, we’ll define the classes for a hotel management system. Since we are following the bottom-up approach to designing a class diagram, we will create the classes of small components first. Next, we will integrate these components and create the class diagram for the entire hotel management system.

### Address and Account

The Address is a class that is required to store any address. The Address is a custom data type that has attributes like a street address, city, etc. In the hotel management system, this class will be used to specify the address of the users and the hotel.

Account is a class that is used to store the account information of the user. This class has three members, i.e., account ID, password, and the status of the account. The class representation of Address and Account classes is as follows:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a web page

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

AccountStatus: This status tells the status of the user account whether it is active, closed, canceled, or blocklisted.

AccountType: The account type tells the type of the account of the user, whether it is a member, guest, manager, or receptionist.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A diagram of a hotel management system

Description automatically generated

## Design pattern

The Strategy design pattern is applied here, which will design a separate strategy or algorithm to calculate the rate of each room.

The Hotel class follows the Singleton design pattern, because there will only be a single instance of the Hotelclass.

Inside the hotel management system, there can be multiple rooms in a hotel, and each room has its own formula for calculating the booking cost. Therefore, Room and related classes incorporate the properties of the Factory design pattern.

## Additional requirements

The interviewer can introduce some additional requirements in the given hotel management system, or they can ask some follow-up questions. Let's see examples of the additional requirements:

**Discount:** A discount will be applied to the payment depending on special events such as the new year, branch opening, and so on. The class diagram provided below shows the relationship of Discount with the BillTransaction class:

A screenshot of a computer screen

Description automatically generated

# Sequence Diagram for the Hotel Management System

Create a sequence diagram for online room booking in the hotel management system and solve a challenge.

**We'll cover the following**

* [Book a room](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Book-a-room)
* [Sequence challenge: Check out](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Sequence-challenge:-Check-out)

Sequence diagrams are a great way to understand the interactions between different entities and objects in the system. There can be different sequence diagrams that we can create for our hotel management system. In this lesson, we will create sequence diagrams for the following two interactions:

* **Book a room:**The guest books a hotel room online.
* **Sequence challenge:** The guest checks out of their room at the reception.

## Book a room

The sequence diagram for the room booking should have the following actors and objects that will interact with each other:

* **Actor:** Guest
* **Objects:** Catalog, Booking, Room, and Payment
* System

Here’re the steps in the book room interaction:

1. The guest searches for a room based on some price and style.
2. The catalog returns a list of rooms.
3. The guest selects a room they wish to book.
4. If the room is available:
   1. The guest creates a booking for the room.
   2. The booking fetches the booking price for the room.
   3. The guest is informed that the booking is ready for payment.
   4. The guest initiates a payment against the booking price.
   5. The payment is processed, and the guest is informed of the status.
   6. If the payment is successful:
      1. The guest is informed that the payment has succeeded.
      2. The system is informed that payment is complete.
      3. The system updates the room status to reserved.
   7. Else if the payment is unsuccessful:
      1. The guest is informed that the payment has failed.
5. Else if the room is unavailable:
   1. The system informs the guest that the room is unavailable.

Based on the order above, the sequence diagram of booking a room in a hotel management system is given below:

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Alternatively, you can also click the "Show complete diagram" button to view the complete sequence diagram of the check-out interaction.

A screenshot of a computer screen

Description automatically generated

# Activity Diagram for the Hotel Management System

Create some activity diagrams for the hotel management system problem.

**We'll cover the following**

* [Hotel check-in](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Hotel-check-in)
  + [States](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#States)
  + [Actions](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Actions)
* [Activity challenge: Cancel room booking](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Activity-challenge:-Cancel-room-booking)

Activities diagrams are a great way to visualize the flow of messages from one activity to the other in the system. There can be different activity diagrams that we can create for our hotel management system. In this lesson, we’ll create activity diagrams for the following two activities:

* Hotel check-in
* **Activity challenge:**Cancel room booking

## Hotel check-in

The following are the states and actions that will be involved in this activity diagram.

### States

**Initial state:**A guest with a room booking comes to the hotel reception for check-in.

**Final state:**The guest successfully checked in at the hotel.

### Actions

The guest has a room booked and arrives at the hotel reception. The receptionist validates the booking and checks if the room is ready. The receptionist then issues room keys and updates the room status.

Based on the order above, the activity diagram of the hotel check-in is shown below.

A diagram of a hotel

Description automatically generated

## Activity challenge: Cancel room booking

You’ll help us create an activity diagram of a customer who wants to cancel their room booking.

A skeleton of the activity diagram is given below, provided that a guest will cancel the room booking from the hotel.

A screenshot of a diagram

Description automatically generated

A screenshot of a diagram

Description automatically generated

# Code for the Hotel Management System

Write the object-oriented code to implement the design of the hotel management system problem.

**We'll cover the following**

* [Hotel management system classes](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Hotel-management-system-classes)
  + [Enumerations](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Enumerations)
  + [Address and account](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Address-and-account)
  + [Person](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Person)
  + [Service](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Service)
  + [Invoice](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Invoice)
  + [Room booking](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Room-booking)
  + [Bill transaction](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Bill-transaction)
  + [Notification](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Notification)
  + [Room, room key and room housekeeping](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Room,-room-key-and-room-housekeeping)
  + [Search and catalog](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Search-and-catalog)
  + [Hotel and hotel branch](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Hotel-and-hotel-branch)
* [Wrapping up](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Wrapping-up)

We’ve reviewed different aspects of the hotel management system and observed the attributes attached to the problem using various UML diagrams. Let’s explore the more practical side of things, where we will work on implementing the hotel management system using multiple languages. This is usually the last step in an object-oriented design interview process.

We have chosen the following languages to write the skeleton code of the different classes present in the hotel management system:

* Java
* C#
* Python
* C++
* JavaScript

## Hotel management system classes

In this section, we’ll provide the skeleton code of the classes designed in the class diagram lesson.

**Note:**For simplicity, we are not defining getter and setter functions. The reader can assume that all class attributes are private and accessed through their respective public getter methods and modified only through their public methods function.

### Enumerations

First, we will define all the enumerations required in the hotel management system. According to the class diagram, there are six enumerations used in the system, i.e., RoomStyle, RoomStatus, BookingStatus, AccountStatus, AccountType, and PaymentStatus. The code to implement these enumerations is as follows:

**Note:** JavaScript does not support enumerations, so we will be using the Object.freeze() method as an alternative that freezes an object and prevents further modifications.

Jav

// definition of enumerations used in hotel management system

enum RoomStyle {

STANDARD,

DELUXE,

FAMILY\_SUITE,

BUSINESS\_SUITE

}

enum RoomStatus {

AVAILABLE,

RESERVED,

OCCUPIED,

NOT\_AVAILABLE,

BEING\_SERVICED,

OTHER

}

enum BookingStatus {

REQUESTED,

PENDING,

CONFIRMED,

CANCELLED,

ABANDONED

}

enum AccountStatus {

ACTIVE,

CLOSED,

CANCELED,

BLACKLISTED,

BLOCKED

}

enum AccountType {

MEMBER,

GUEST,

MANAGER,

RECEPTIONIST

}

enum PaymentStatus {

UNPAID,

PENDING,

COMPLETED,

FILLED,

DECLINED,

CANCELLED,

ABANDONED,

SETTLING,

SETTLED,

REFUNDED

}

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

represented below:

public class Invoice {

private double amount;

public boolean createBill();

}

### Room booking

RoomBooking is a class responsible for managing the bookings for a room. The implementation of this class is given below:

Java

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);

}

### Bill transaction

After generating an invoice, a customer needs to pay the bill to confirm the booking of the room. A BillTransaction class is required to store the information of bill payment. Three ways to pay the bill are check transactions, cash transactions, and credit card transactions.

Java

C#

Python

C++

// BillTransaction is an abstract class

public abstract class BillTransaction {

private Date creationDate;

private double amount;

private PaymentStatus status;

public abstract void initiateTransaction();

}

class CheckTransaction extends BillTransaction {

private String bankName;

private String checkNumber;

public void initiateTransaction() {

// functionality

}

}

class CreditCardTransaction extends BillTransaction {

private String nameOnCard;

private int zipcode;

public void initiateTransaction() {

// functionality

}

}

class CashTransaction extends BillTransaction {

private double cashTendered;

public void initiateTransaction() {

// functionality

}

}

A screenshot of a computer program

Description automatically generated

### Room, room key and room housekeeping

The Room class represents a room in the hotel. RoomKey is a class used to express the electronic key card and the RoomHousekeeping is a class used to keep track of all the housekeeping records for the rooms. The implementation of these classes is given below:

public class Room {

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 class RoomKey {

private String keyId;

private String barcode;

private Date issuedAt;

private boolean isActive;

private boolean isMaster;

public boolean assignRoom(Room room);

}

public class RoomHousekeeping

{

private String description;

private Date startDatetime;

private int duration;

private Housekeeper housekeeper;

public boolean addHousekeeping(Room room);

}

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated