

Bookings

Final Project

Outline	1
Hands-on	1
Create the Bookings app	1
Data Model	2
Roles	2
Rooms	2
Rooms Screen	2
RoomDetail Screen	3
Bookings	4
Bookings Screen	4
BookingDetail Screen	5
Check-in and Check-out	7
Homepage	8
Room Amenities	9
Bonus Challenge: Room Service	10

Outline

In this project, we will create an app named Bookings that will be used to manage hotel rooms and bookings.

The app will be able to list and edit the information of existing rooms, as well as create new rooms. Also, in each room, we will be able to add amenities, such as an extra TV or a Crib.

Then, the app will allow you to create new bookings for guests and cancel them if needed. When the check-in date comes, the hotel clerk / manager should be able to check in the guests as well as proceed with the check-out at any time after the check-in.

The Bookings app will also have a homepage that displays the bookings with the check-in / check-out dates on the current date.

In the end, there are two extra challenges, one to add the room service functionality to the app, and another one to add a chart with the hotel's occupancy rate in the next 7 days.

Hands-on

Create the Bookings app

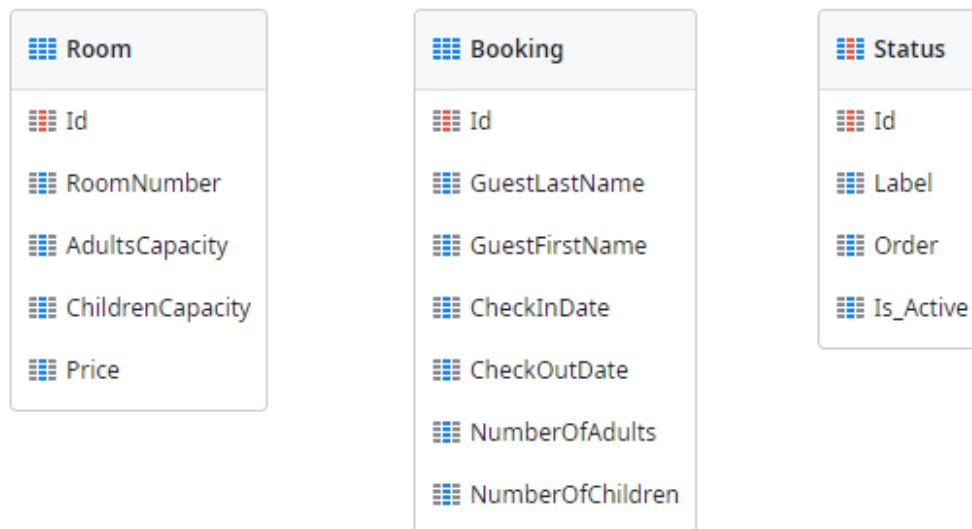
To start this assignment, the first step is to create the Bookings web app. There is an icon in the exercise's resources that can be used for the app.



Note: Don't forget that the names of the apps must be unique in the organization!

Data Model

At first, the app will have a very simple data model with the *Room*, *Booking*, and *Status* Entities shown in the picture below.



- All the attributes are mandatory except for the number of children.
- The possible statuses are *Booked*, *CheckedIn*, *CheckedOut*, and *Canceled*.
- Use the **Room.xls** file to populate the Room Entity with data.
- Create the relationships you think are relevant to this app.

Roles

Before jumping into the UI part of the app, let's first go through the roles that will be able to access and interact with the app. The two roles are:

- Hotel manager
- Hotel clerk

We will let you know what each role can do in each section of the app, when applicable.

Rooms

The app will have two Screens to manage the hotel's rooms, one to display all the rooms and another to view or edit the room's details. We recommend you to start with the Rooms Screen.

Rooms Screen

This Screen should display a list of all the Rooms in the database with the following requirements:

- The Table should display the Room Number, Adults Capacity, Children Capacity and Price per night;
- The Screen should have a link to the Room's Detail Screen;
- The Screen should have a link to create a new Room;
- The Screen should only be accessible by the hotel manager and the hotel clerk.

The Rooms Screen should look somewhat like this:

Rooms

New Room

Room Number ↕	Adults Capacity ↕	Children Capacity ↕	Price ↕
101	1	0	\$34.00
102	2	0	\$48.00
103	2	1	\$52.00
104	2	2	\$64.00
201	1	0	\$34.00

1 to 5 of 10 items

<
1
2
>

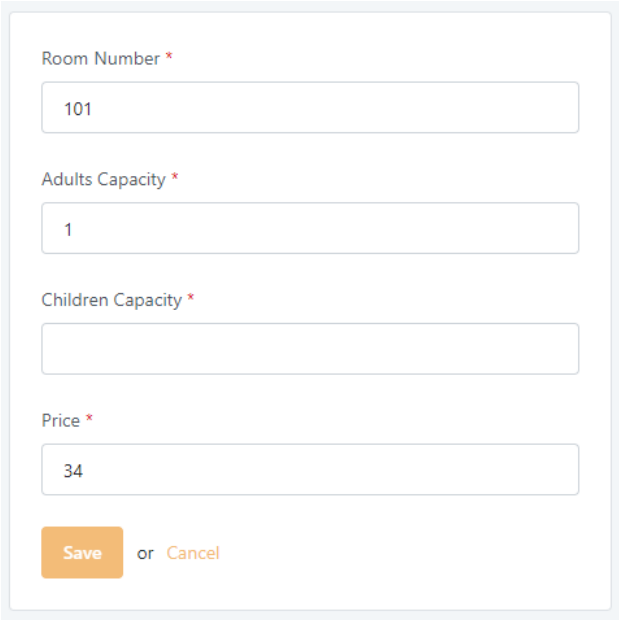
RoomDetail Screen

This Screen will allow viewing and editing the information about existing Rooms, as well adding new Rooms to the hotel. This information should be displayed using a Form, with an input field for the Room Number, Adults Capacity, Children Capacity and Price per night.

The Form should also have a Save button and a Cancel link. The first one saves the data typed in the Form to the database, while the link sends the user back to the Rooms Screen.

- The Screen should be accessible by the hotel manager and the hotel clerk;
- Only the hotel manager should be able to create a new room or edit the Room's information.

The RoomDetail Screen should look like the following screenshot:



The screenshot shows a form titled 'RoomDetail' with the following fields and controls:

- Room Number ***: A text input field containing the value '101'.
- Adults Capacity ***: A text input field containing the value '1'.
- Children Capacity ***: An empty text input field.
- Price ***: A text input field containing the value '34'.
- Buttons**: An orange 'Save' button followed by the text 'or' and a 'Cancel' button.

Bookings

The most important part of the app is to allow a user to create Bookings for guests. For that, it is important to have two Screens: one to list the existing Bookings and another one to create / modify Bookings in the database.

Bookings Screen

The Bookings Screen should list all the bookings in the database in a table. The table should display the Guest's first name and last name, plus the check-in and check-out dates, the room number and the booking status. The Screen should be accessible by the hotel manager and the hotel clerk.

You should be able to search Bookings by guest name and/or the Booking status, and also have links to access existing Bookings (in the Guest Last Name) or to create new Bookings. The list Screen should look like the following screenshot:

Bookings

[New Booking](#)

<input type="text" value="(Type Guest Name)"/>	<input type="text" value="-Any Status-"/>	<input type="button" value="Search"/>
--	---	---------------------------------------



Guest Last Name ↕	Guest First Name ↕	Room Number ↕	Check In Date ↕	Check Out Date ↕	Status ↕
Gonçalves	Sofia	101	23 Jan 2020	23 Jan 2020	CheckedIn
Noronha	Beatriz	301	23 Jan 2020	23 Jan 2020	CheckedOut
Pereira	Joana	201	23 Jan 2020	25 Jan 2020	Booked
Pestana	Tiago	101	24 Jan 2020	26 Jan 2020	Booked
Silva	Gabriel	202	7 Apr 2020	9 Apr 2020	Canceled

BookingDetail Screen

The BookingDetail Screen should allow a user to create a new Booking, as well as view or edit the information about an existing booking. Here are some important aspects of the Screen that you should consider:

- The Screen should only be accessible by the Hotel Manager and the Hotel Clerk;
- The hotel Manager and the Hotel Clerk can create or modify bookings;
- The Screen should have a link to go back to the Bookings list;

The Detail Screen should look something like this:

Guest Last Name *	<input type="text" value="Smith"/>
Guest First Name *	<input type="text" value="David"/>
Check In Date *	<input type="text" value="11/10/2022"/> 
Check Out Date *	<input type="text" value="11/18/2022"/> 
Number Of Adults *	<input type="text" value="2"/>
Number Of Children	<input type="text" value="1"/>
<input type="button" value="Get Available Room"/> Back to Bookings	

Now, this Screen in particular has a button, Get Available Room, that has an important information. When this Button is clicked, we expect to show the cheapest available room in the hotel, considering the dates and the number of people. To accomplish the last requisite we can use the following SQL query to fetch the information we need:

```
SELECT {Room}.* FROM {Room}

WHERE @NumberOfAdults > 0

AND {Room}.[AdultsCapacity] >= @NumberOfAdults

AND {Room}.[AdultsCapacity] + {Room}.[ChildrenCapacity] >= 0 + @NumberOfAdults +
@NumberOfChildren

AND NOT EXISTS

    (SELECT 1 FROM {Booking}

        WHERE {Booking}.[RoomId] = {Room}.[Id] AND ({Booking}.[CheckInDate] <= @CheckOutDate AND
{Booking}.[CheckOutDate] >= @CheckInDate) AND {Booking}.[StatusId] <> @CanceledStatus)

LIMIT 1
```

Don't forget to save the cheapest available room somewhere on your Screen, since you will need it to display the information.

After clicking on the Get Available Room button, the interface should change slightly:

- Right above the Button, the information about the room should be displayed. We suggest something like: Room (Room Number) at (Price)\$ per night.
- The Get Available Room should be replaced by a Book Room Button, which will then effectively create the new Booking.



While creating a new booking, or even checking for an available room, don't forget the following validations:

- Check-in and Check-out dates are mandatory;
- Check-out must happen after the Check-in;
- There must be at least one adult in the room;

- The number of children cannot be negative.

Check In Date *

11/08/2022

Check Out Date *

11/06/2022

Date of checkout must be after checkin date

The form contains two date input fields. The first field, labeled 'Check In Date *', contains the date '11/08/2022'. The second field, labeled 'Check Out Date *', contains the date '11/06/2022'. A red border highlights the second field, and a red error message 'Date of checkout must be after checkin date' is displayed below it.

Check-in and Check-out

Still in the BookingDetail Screen, there's a couple of extra things we can do.

When the booking is created, the hotel clerk / manager can check in the guests or cancel the booking. These options should appear in the BookingDetail Screen **only AFTER the Booking is created**. Also, the check-in shouldn't be possible before the Check-in date. In summary, the experience should be:

1. Clerk / manager books the room.
2. After the room is booked, a Cancel option should immediately appear on the BookingDetail Screen.
3. Whenever the Check-in date is the actual current date, the Check-in option also should appear.

Room 301 at 107\$ per night.

Check In Cancel Reservation Back to Bookings

The screenshot shows a booking detail screen for 'Room 301 at 107\$ per night.'. Below the room information, there are two orange buttons: 'Check In' and 'Cancel Reservation'. To the right of these buttons is a link labeled 'Back to Bookings'.

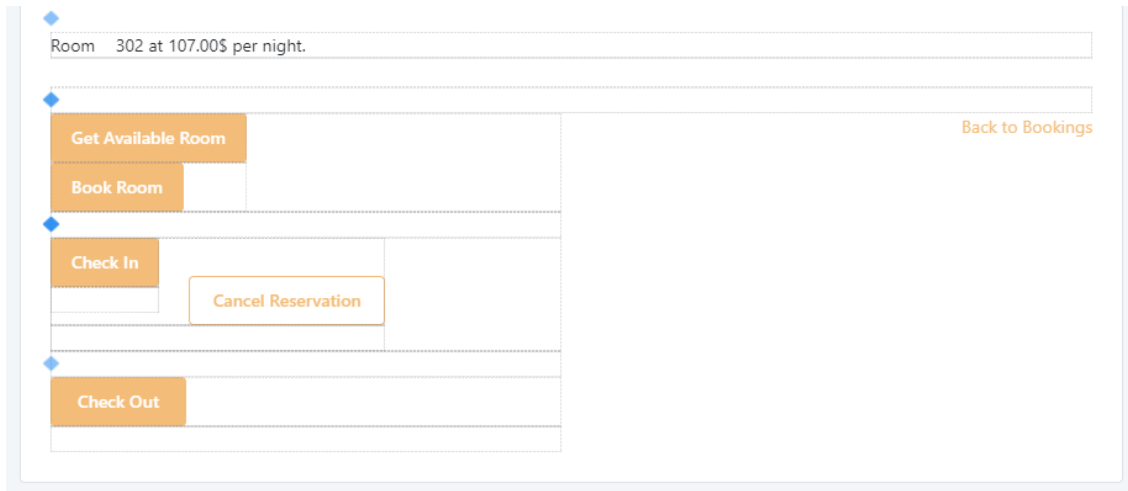
For the check-out, the hotel clerk / manager can see that option only when the user is already checked-in. In this case, since the guests can leave the hotel earlier if they want to, the check-out can happen anytime between the check-in and check-out dates. This option should also appear in the BookingDetail Screen.

Room 102 at 48\$ per night.

Check Out Back to Bookings

The screenshot shows a booking detail screen for 'Room 102 at 48\$ per night.'. Below the room information, there is an orange button labeled 'Check Out'. To the right of this button is a link labeled 'Back to Bookings'.

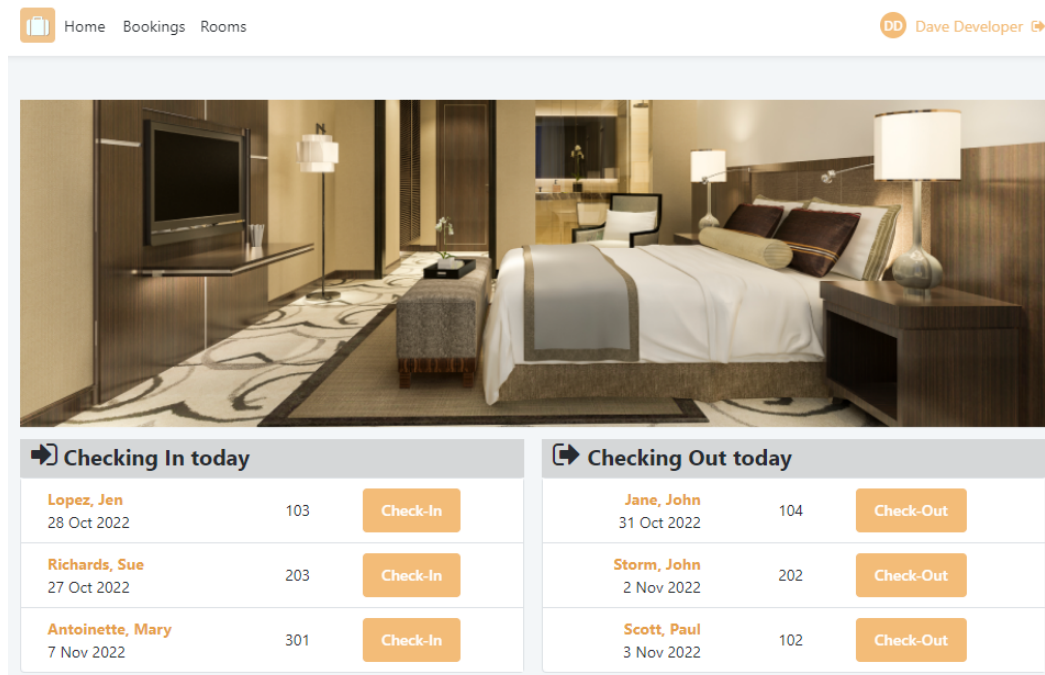
The Screen preview in ODC Studio should look similar to the following screenshot, but when you open it in the browser, only the expected options should appear



Homepage

Let's now create one more Screen: the Homepage to your app. This Homepage will display a banner and then the lists of bookings checking-in and checking-out today. The image for the banner can be found in the Resources folder of the exercise. The lists should appear side by side and they should display the name of the guest, the room number, the check-out date (for the check-in list) and the check-in date (for the check-out list).

The Screen should look like the following screenshot and should be accessible by hotel clerks and manager:



As you can see from the screenshot, there are a couple of aspects that caught our attention:

- The Checking In today and Checking Out today use some specific stylings. This is just a recommendation. Feel free to add your CSS or apply any of the existing styles to the Containers to create your look and feel.
- Each row of the List will have a Link and a Button. The Link appears on the guest's name to access the booking information. The Button allows an hotel clerk or manager to check-in or check-out a guest, depending on the list we're looking at.
- Notice that the Guest Name appears differently in these two lists. So, let's standardize the way a guest name appears on the app to something like this:

Montgomery, Jane

Millard, Anthony

Stuart, Pamela

Baker, John

Room Amenities

With the Homepage fully implemented, let's move on to a different aspect of an hotel: the room amenities. For this application, a Room can have the following amenities:

- Television;
- Internet Access;
- Hairdryer;
- Premium Towels;
- Crib;
- Safe.

To support this new functionality, the data model needs to be changed to include the amenities. Don't forget to create the respective relationships with the Room Entity.

The list of amenities associated with the room should appear in the **RoomDetail** Screen as the screenshot:

Room 101

Room Number *

Adults Capacity *

Children Capacity *

Price *

Save or Cancel

Amenities

Add New

Amenity
InternetAccess

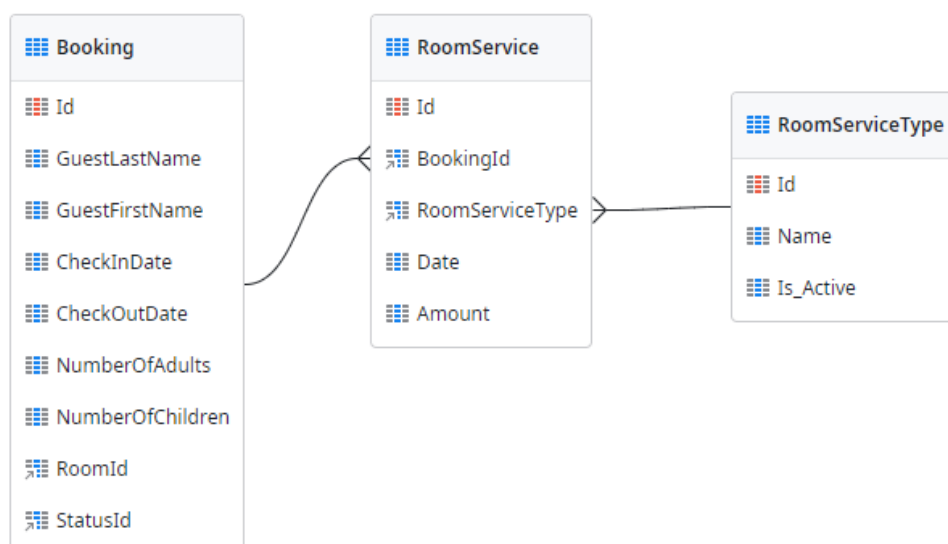
Delete

- The app should also have a new Screen to allow adding new amenities to a room.
- The hotel clerk / manager can add amenities to a room.
- **(Bonus)** A Room should not have the same amenity twice.

And that's it! The application is finished. Want more and have some time and desire? So, let's jump to some extra challenge.

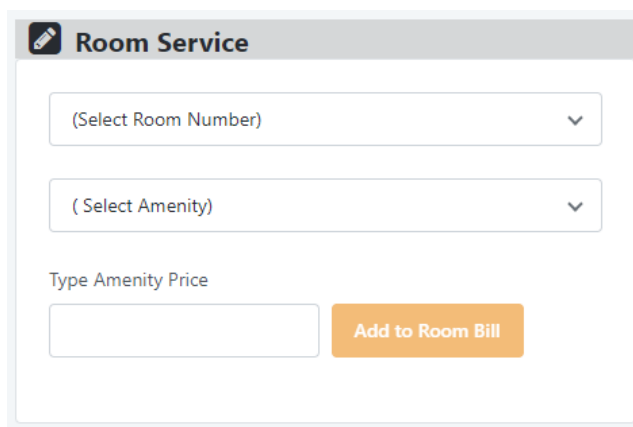
Bonus Challenge: Room Service

In this extra section of the assignment, we want to create a Room Service functionality. To do that, we first need to change the Data Model to support the Room Service.



On the Homepage of the app, we will have a section to request a service to a Room. To do so, the hotel clerk / manager should select a room, the type of service that the customer wants, and

the price. Then, the hotel clerk / manager should confirm the request by clicking on a Button.
The Room Service section should look like the following image:



The image shows a web form titled "Room Service" with a pencil icon. It contains two dropdown menus: the first is labeled "(Select Room Number)" and the second is labeled "(Select Amenity)". Below these is a text input field labeled "Type Amenity Price". To the right of the input field is an orange button labeled "Add to Room Bill".