outsystems

Bookings - Exercise

Table of Contents

Outline

In this project, we will create an application to manage hotel Bookings. This application will have all the rooms of the hotel and also all the bookings made in the hotel.

Our application will be able to list all rooms, as well as create new rooms and edit the information of existing rooms. Also, in each room, we will be able to add amenities, such as TV or Crib. All of this can only be done by hotel clerks or the hotel manager.

Then, the application will allow the clerk / manager to create new bookings for guests, as well as cancel them if needed. Then, when the check-in date comes, the hotel staff can check-in the guests as well as proceed with the check-out at any time after the check-in.

The application will also have an homepage which will display the bookings with check-in date and check-out date on the current date, as well as a view over the occupancy of the hotel in the next 7 days.

At the end, there is an extra challenge to add the room service functionality to the application.

Hands-on

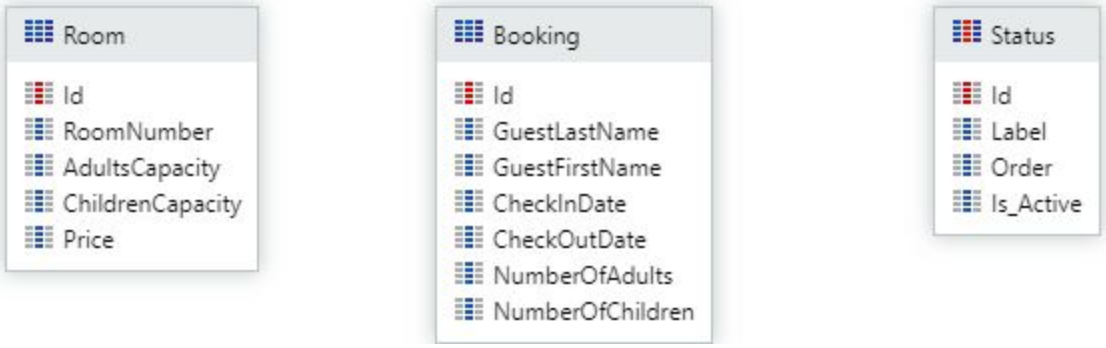
Create the Bookings application

To start this assignment, the first step is to create the Bookings reactive web application. In the resources of the exercise, there is an icon to be used for the application.

The application should have two modules: one Core, for the data model, and one for all the UI and business logic. Don't forget that the names of the modules must be unique in the environment.

Data Model

The application will require a very simple data model, with the following Entities.



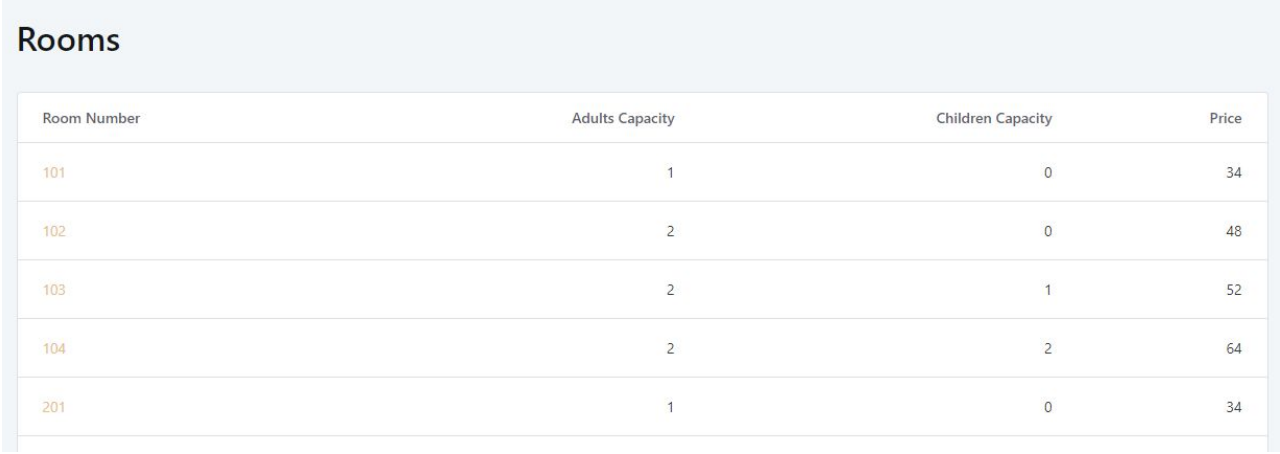
All the attributes are mandatory except for the number of children and consider the following statuses: Booked, CheckedIn, CheckedOut and Canceled.

Use the Room.xls file to populate the Room Entity with data.

Create the relationships you think are relevant for this application.

Rooms

The application will have a Screen to list all the Rooms available in the database. The Screen should only be accessible by the Hotel Manager and the Hotel Clerk and it should look something like this:



The application will also have a Screen to allow creating and editing Rooms, with a Save Button, to save the information in the database, and a Cancel Link to go back to the list of Rooms.

This Screen has a Save button and a Cancel link. The first one will save the data on the Room Form to the database, while the link will go back to the Rooms Screen. The Screen should be accessible by the hotel manager and the hotel clerk, but only the hotel manager will be able to create a new room, or edit its information.

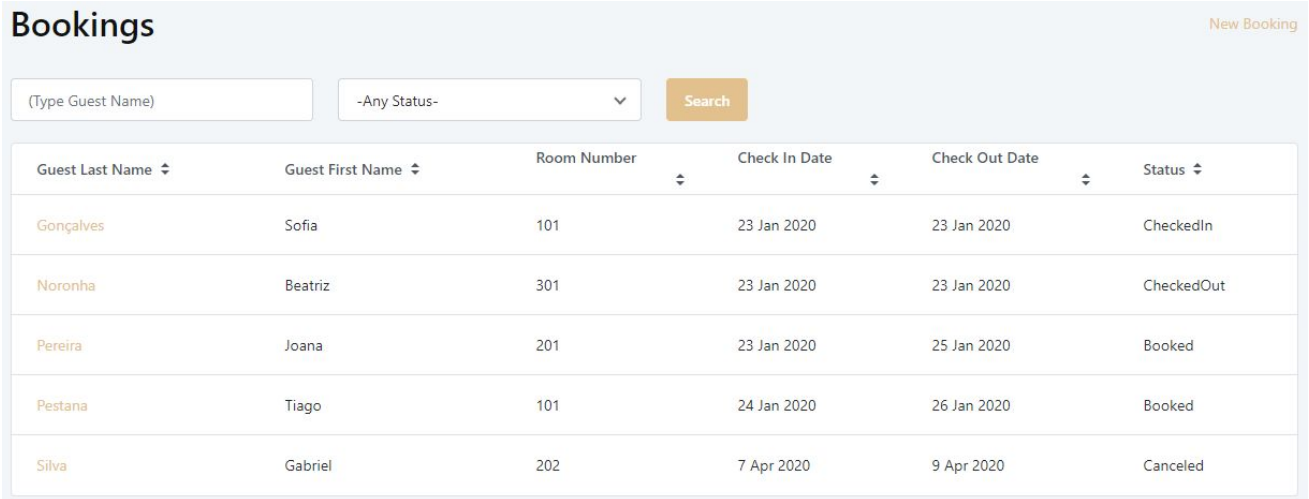
This Screen should look like the following screenshot:

Bookings

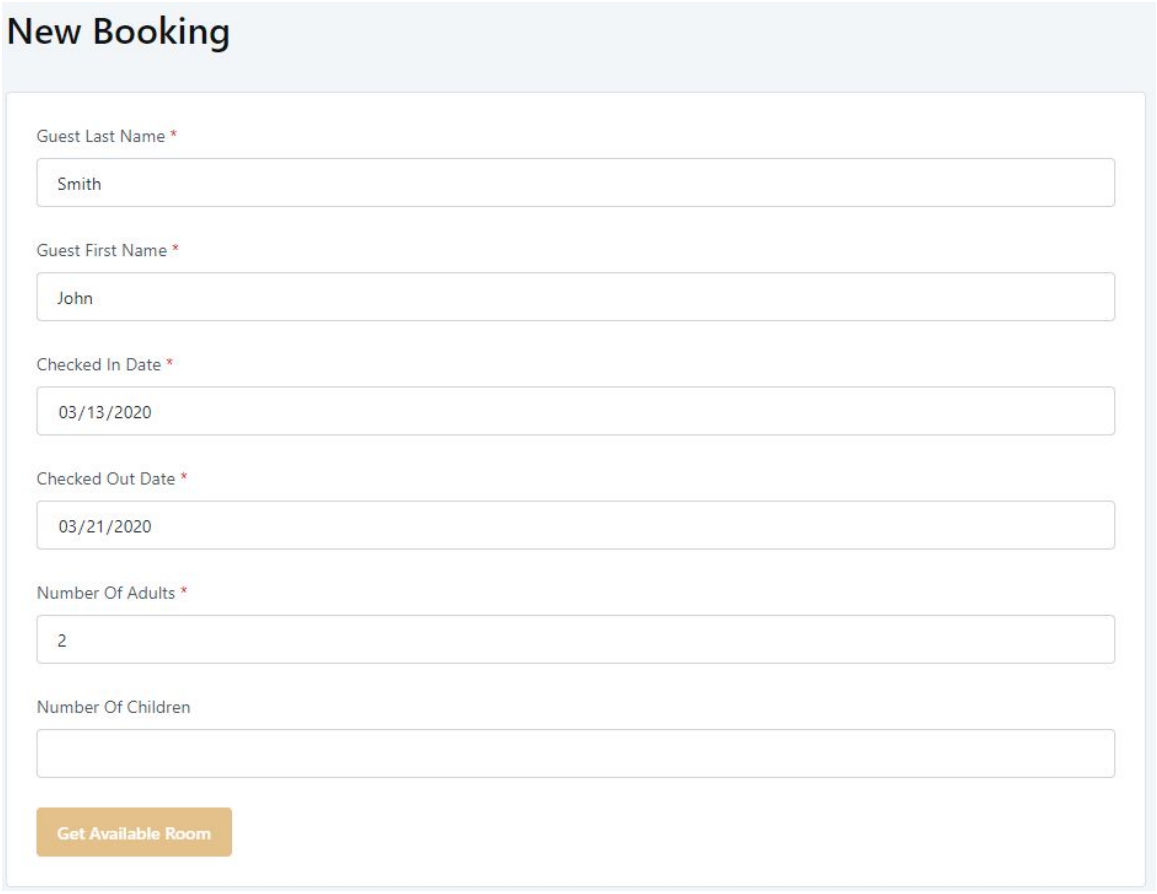
The most important part of the application is to allow a hotel clerk (or manager) to create Bookings for guests.

For that, it is important to list the existing Bookings and to create / modify Bookings in the database.

Also, the clerk / manager should be able to search for guest names and / or the status of the Booking. The list Screen should look like the following screenshot:



The Detail Screen should look like something this:

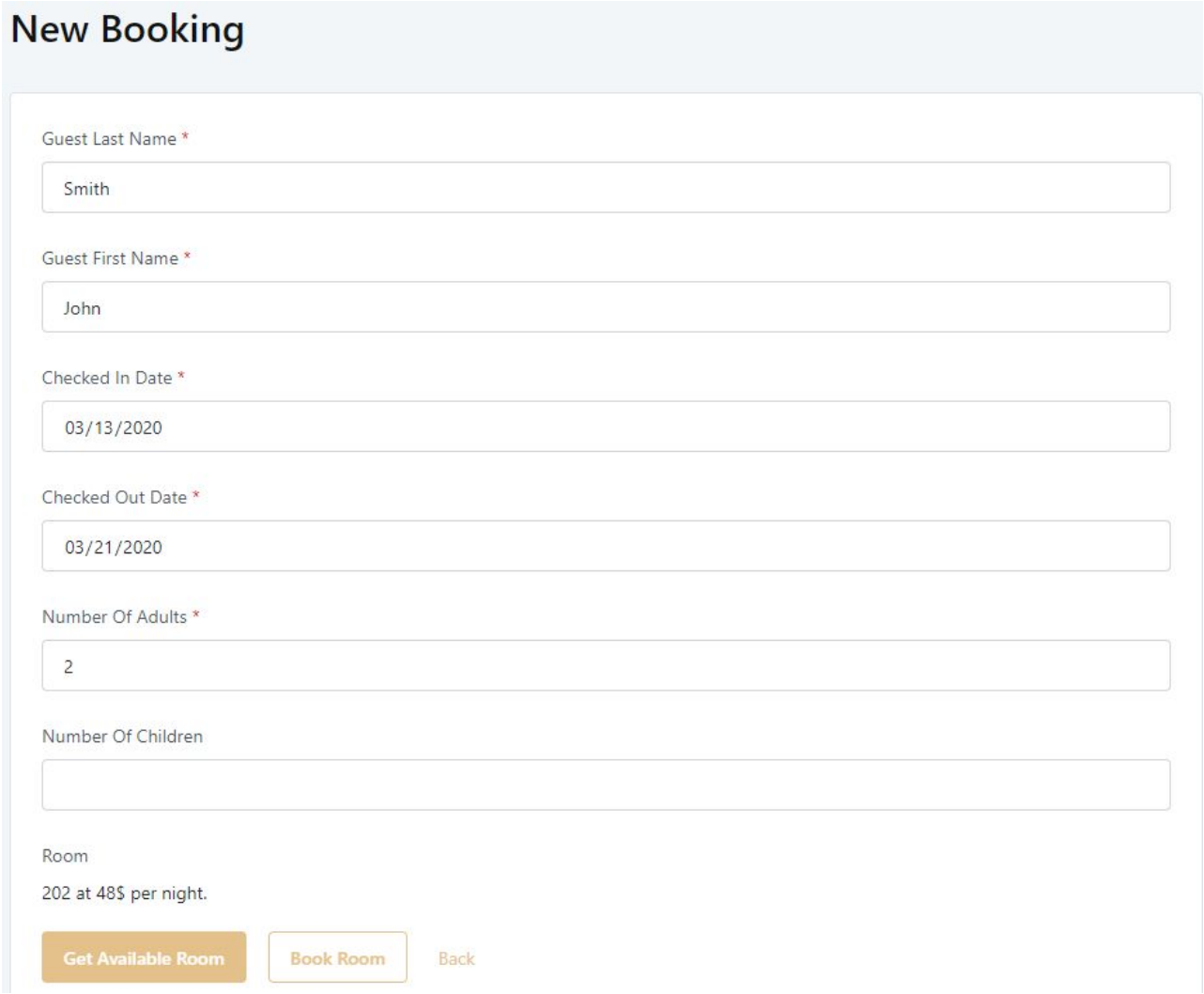


The application should automatically choose the cheapest available room option, when the clerk / manager selects the Get Available Room option. To accomplish that, we can use the following SQL:

```
SELECT {Room}.* FROM {Room}
WHERE {Room}.[AdultsCapacity] >= @NumberOfAdults
and {Room}.[AdultsCapacity] + {Room}.[ChildrenCapacity] >= @NumberOfAdults + @NumberOfChildren
and not exists
  (SELECT 1   FROM {Booking}
  WHERE {Booking}.[RoomId] = {Room}.[Id]
  and (@CheckInDate between {Booking}.[CheckInDate] and {Booking}.[CheckOutDate]-1
    or @CheckOutDate between {Booking}.[CheckInDate]+1 and {Booking}.[CheckOutDate])
  and {Booking}.[StatusId] <> @CanceledStatus)
ORDER BY {Room}.[Price] ASC
```

The Room row should only appear after the Get Available Room is selected.

Then, when the available room is selected, an option should appear to book the room or to return to the Bookings page.



While creating a new booking, don't forget these 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.

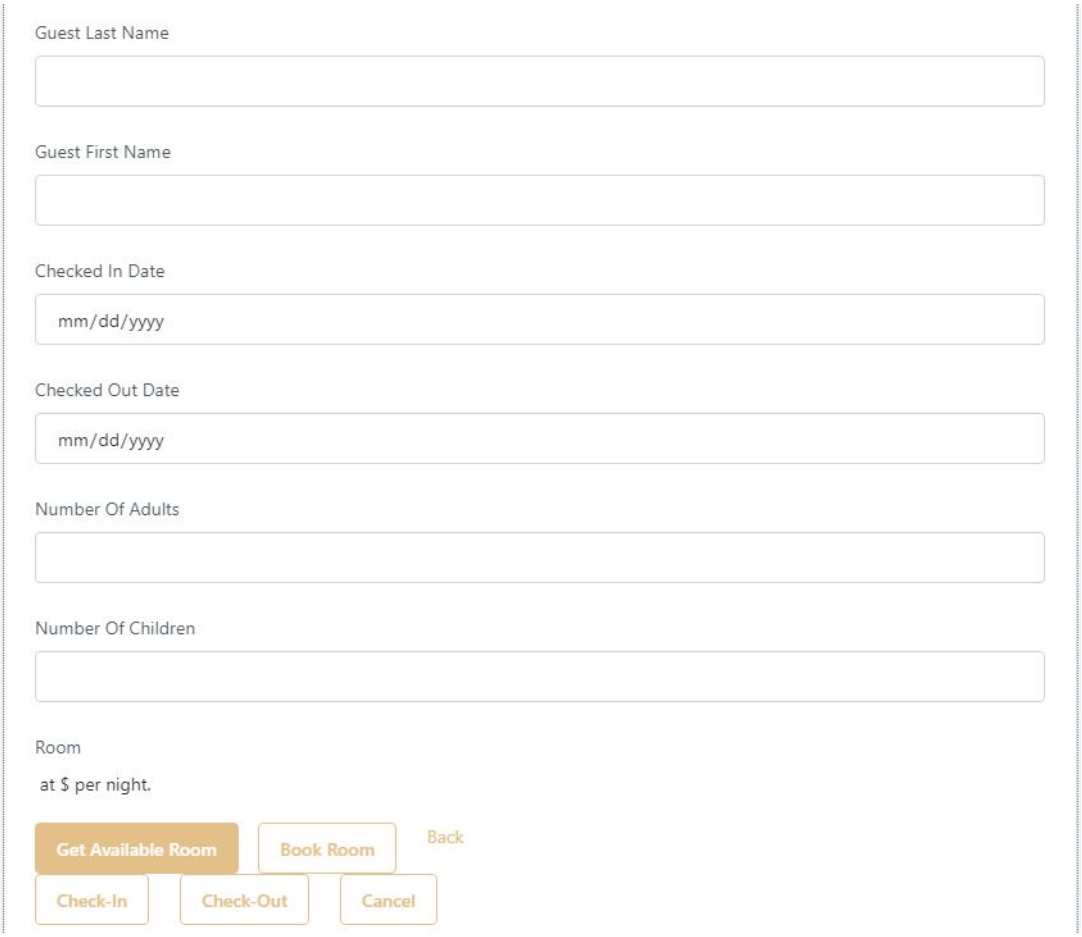
Check-in and Check-out

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

1. Clerk / manager books the room.
2. In the BookingDetail Screen, after the room is booked, a Cancel option should appear.
3. In the BookingDetail Screen, whenever the Check-in date is the actual current date, the Check-in option should appear.

For the check-out, the hotel clerk / manager can only see that option, 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.

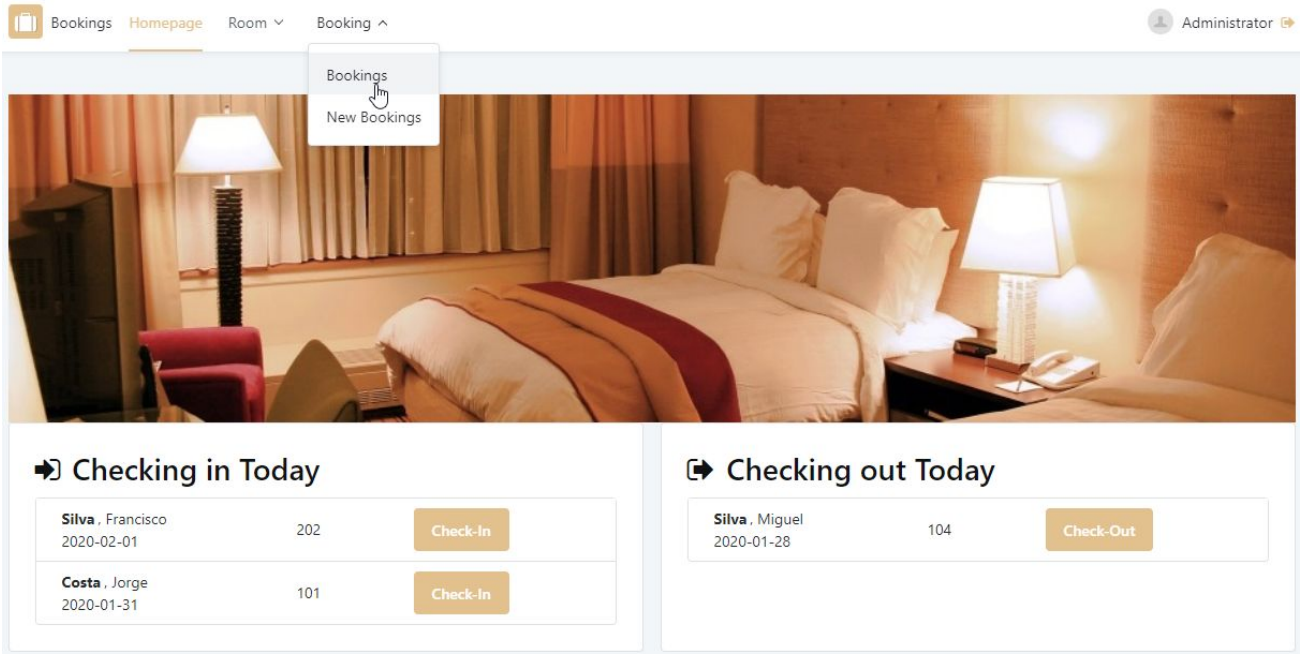
In Service Studio, the Screen will look like the following screenshot, but depending on the scenario, only the expected options should appear



Homepage

Now, we want to have a Homepage in our application. This homepage will display a banner and then a list of bookings where the check-in date is today, and a list of bookings where the check-out is today. That list should allow the hotel clerk / manager to check-in and check-out the guests directly, without going to the Booking Detail page.

These lists should have the name of the guest, the room number and the check-out date, in the Checking In Today list, and the check-in date in the Checking Out Today list. The Screen should look like the following screenshot:



The image can be found in the Resources folder of the exercise.

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:

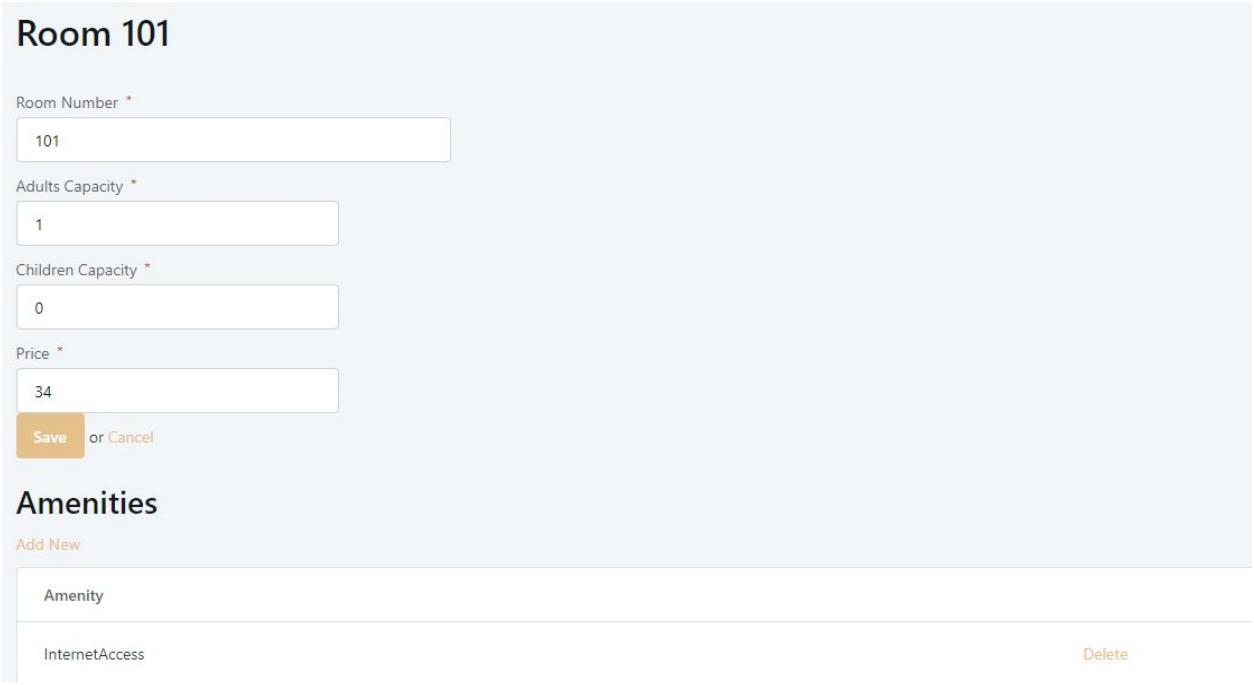
Room Amenities

Another functionality that the Bookings application will need to support is the addition of amenities to a room. A Room can have the following amenities:

- Television;
- Internet Access;
- Hair dryer;
- 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.

In the Room Detail Screen, the list of amenities in the room should appear as in the following screenshot:

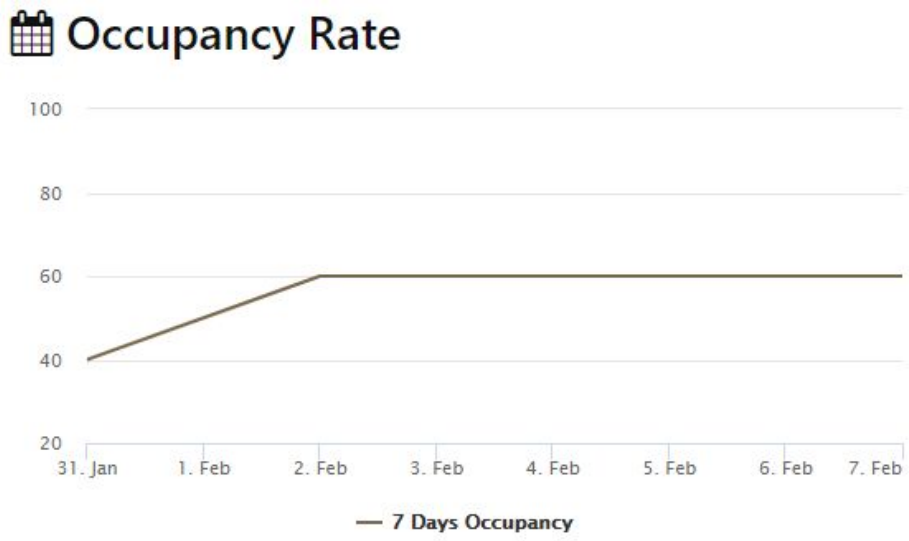


The application should also have a new Screen to allow the hotel clerk / manager to add new amenities to a room. A Room should not have the same amenity twice.

Occupancy for the Next 7 Days

In the Homepage Screen, we want to display the hotel occupancy in the next 7 days, starting from the current date. This will allow the hotel clerk /manager to plan the next few days, and better allocate the available resources.

This information should be displayed in a chart, where the percentage of occupancy will be displayed in the Y-axis and the days in the X-axis. At the end of this section, we should have a Chart looking like this:



To create the Chart, we need a widget, which is a Line Chart. Find out more about Line Charts in this section document.

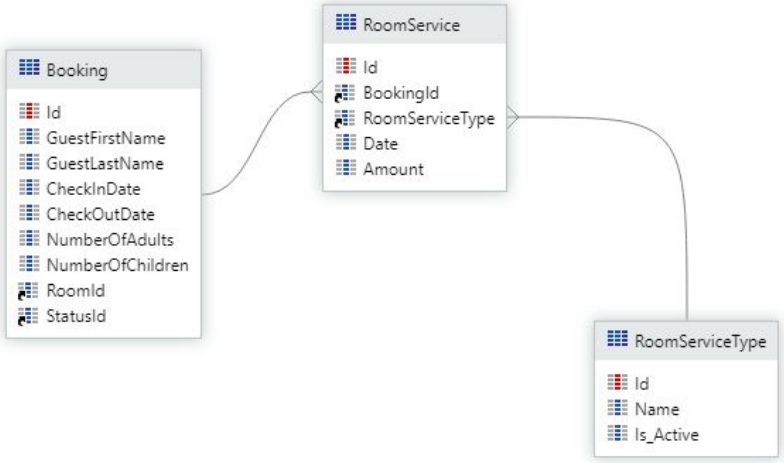
To build the Line Chart, we need a list of data points. To build the chart we need to use a list of data points. The list of data points in this case will have to represent the day and the percentage of occupancy on that day. So, to build the chart, we need to have a list of 7 elements, one per day, with the relevant information in it.

The Line Chart and the Data Point type already exist. We just need to use it and build the list we want to display the correct data on the chart.

Extra 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.



In the Homepage of the application, we will have a section to request a service to a Room. To request the room service, the hotel clerk / manager will be able to select a room, select the type of service that the customer wants and after the selection is made, the total price should appear. Then, the hotel clerk / manager should confirm the Request by clicking on a Button. The Room Service section should look like the following mockup: