

Bookings

Final Project - Mobile

Outline	2
Hands-on	2
Create the Bookings app	2
Data Model	3
Roles	3
Rooms	3
Rooms Screen	4
RoomDetail Screen	4
Bookings	5
Bookings Screen	6
BookingDetail Screen	6
Check-in and Check-out	9
Homepage	10
Room Amenities	11
Bonus Challenge: Room Service	12

Outline

In this project, we will create a phone 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

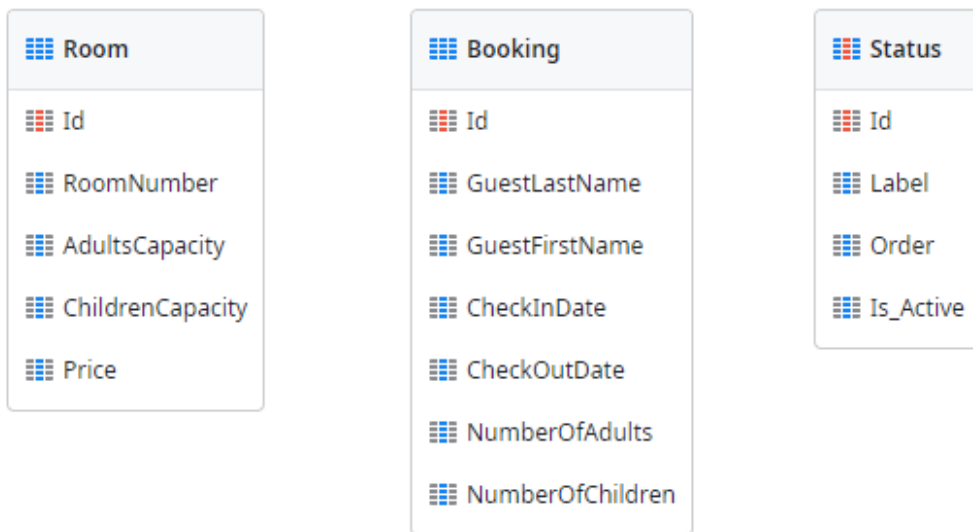
To start this assignment, the first step is to create the Bookings phone app. There is an icon in the exercise's resources that can be used for the app.



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

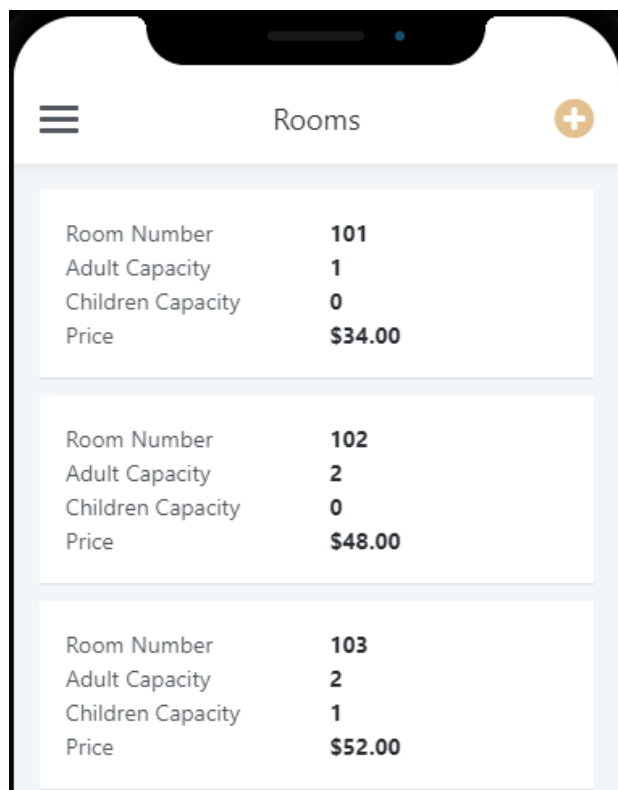
The app will have two Screens to manage the hotel's rooms, one to display all the rooms and another to view or edit the room's details. We recommend that you start with the Rooms Screen.

Rooms Screen

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

- The List 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:



RoomDetail Screen

This Screen will allow viewing and editing the information about existing Rooms, as well as 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 screenshot shows a mobile application interface for editing a room. The title bar at the top has a back arrow on the left and the text "Edit Room" in the center. Below the title bar is a light blue container with a white background. Inside this container, there are four input fields, each with a label and a red asterisk indicating a required field: "Room Number *", "Adults Capacity *", "Children Capacity", and "Price *". The "Room Number" field contains the value "102", the "Adults Capacity" field contains "2", and the "Price" field contains "48". The "Children Capacity" field is empty. At the bottom of the form, there is an orange "Save" button and a link that says "or Cancel".

Bookings

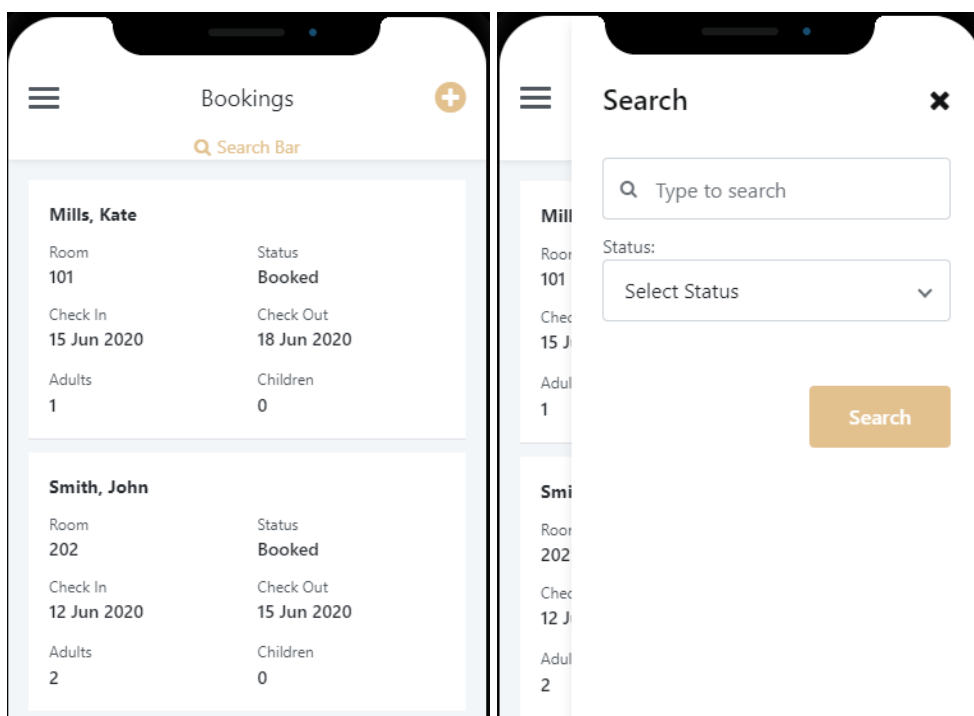
The most important part of the app is to allow a user to create Bookings for guests. For that, it is important to have two Screens: one to list the existing Bookings and another one to create/modify Bookings in the database.

Bookings Screen

The Bookings Screen should display all the bookings in the database in a List. The List should show 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, using a Sidebar widget triggered by a Search Bar link at the top of the Screen. The screen should also have links to access existing Bookings or to create new Bookings. Feel free to create these links wherever you prefer and make sense for the usability of the app.

The list Screen should look like the following screenshot:



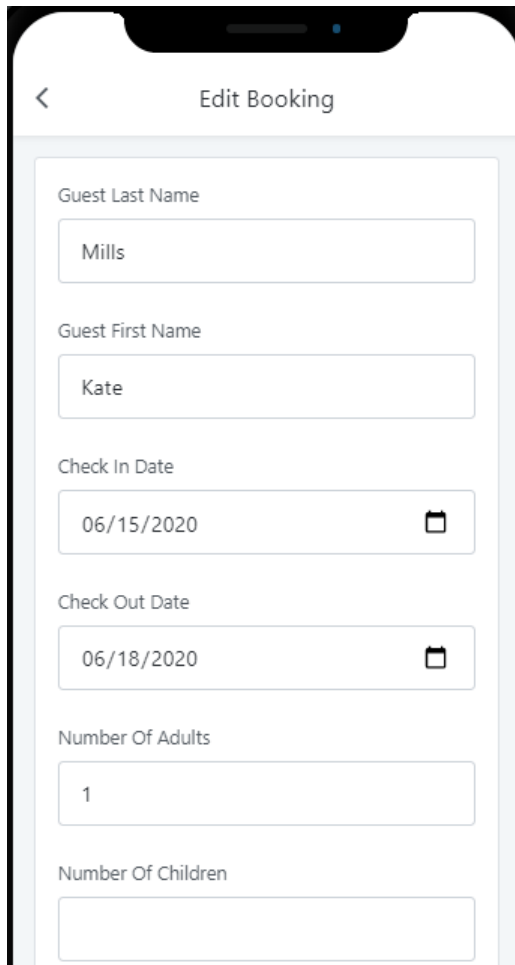
BookingDetail Screen

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

Mills

Guest First Name

Kate

Check In Date

06/15/2020

Check Out Date

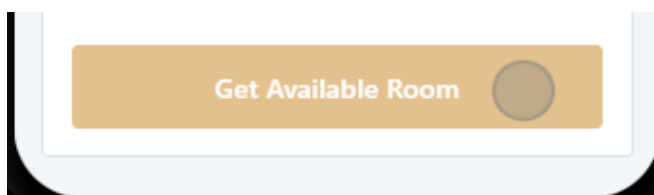
06/18/2020

Number Of Adults

1

Number Of Children

Now, this Screen, in particular, has a button **Get Available Room**, that has 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)

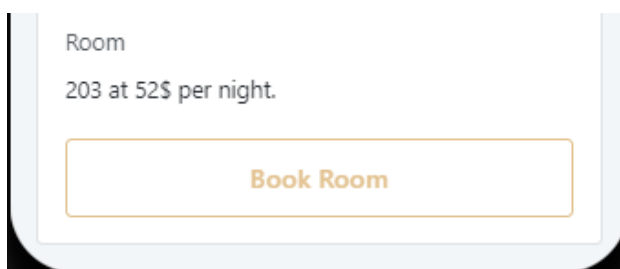
LIMIT 1

```

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;

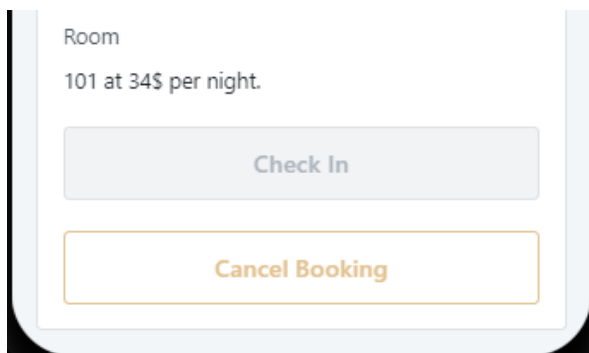
- The number of children cannot be negative.

Check-in and Check-out

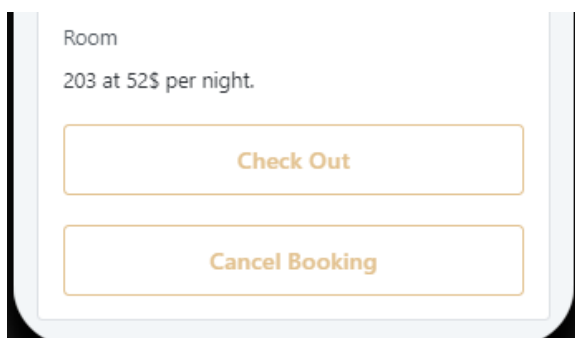
Still in the BookingDetail Screen, there are a couple of extra things we can do.

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.



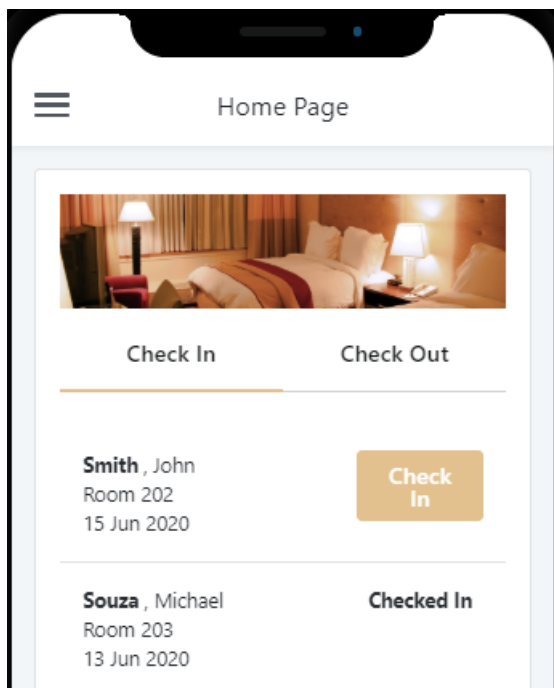
For the check-out, the hotel clerk/manager can see that option only when the user is already checked-in. In this case, since the guests can leave the hotel earlier if they want to, the check-out can happen anytime between the check-in and check-out dates. This option should also appear in the BookingDetail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



Homepage

Now let's create one more Screen: the Homepage of your app. This Homepage will display a banner and 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 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 the hotel clerk and manager:



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

- 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

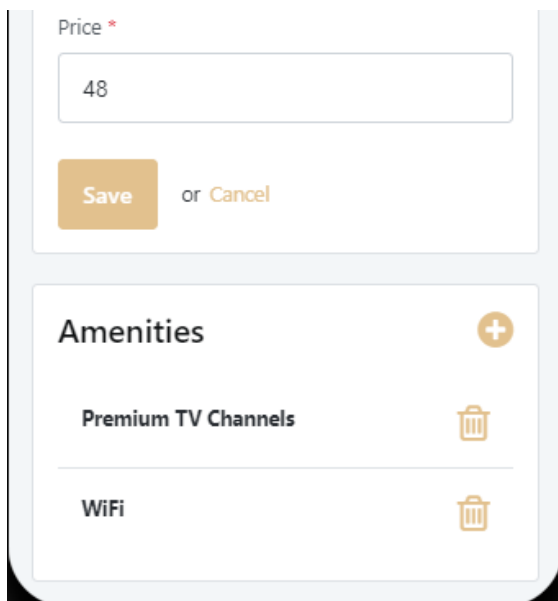
Room Amenities

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

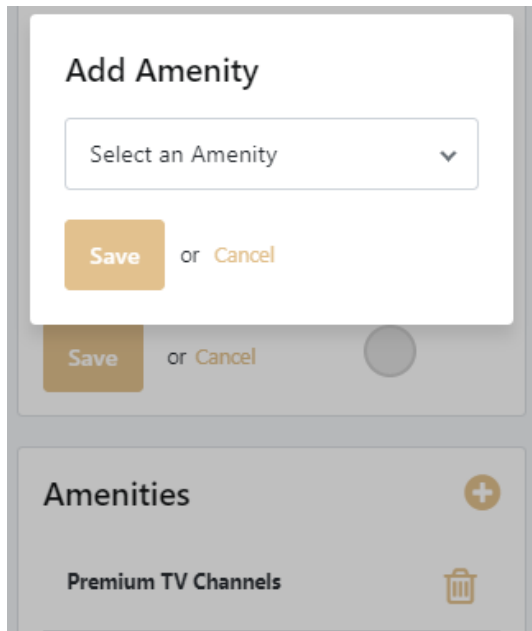
- Television;
- Internet Access;
- Hairdryer;
- Premium Towels;
- Crib;
- Safe.

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 following screenshot:



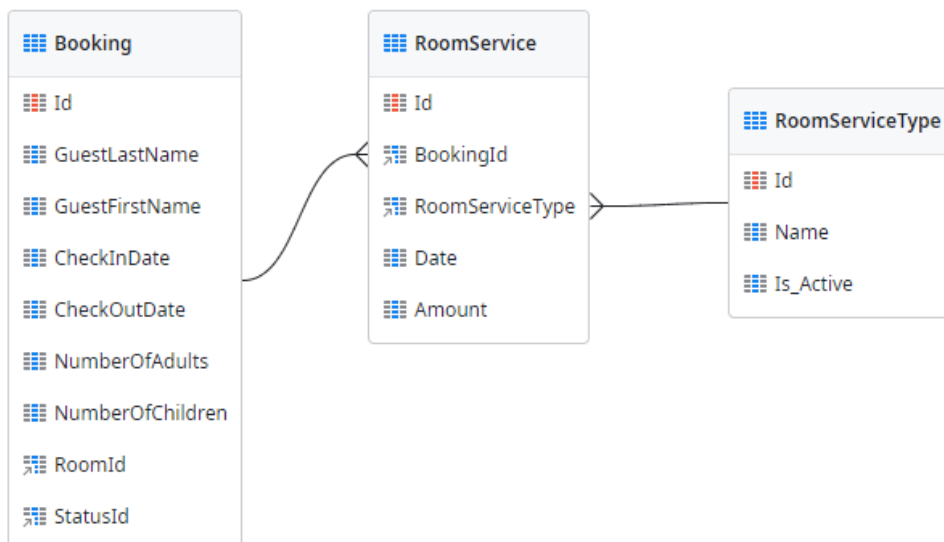
- The app should also have a link (plus icon) to open a Popup to allow adding new amenities to a room.
- The hotel clerk/manager can add amenities to a room.
- **(Bonus)** A Room should not have the same amenity twice.



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

Bonus Challenge: Room Service

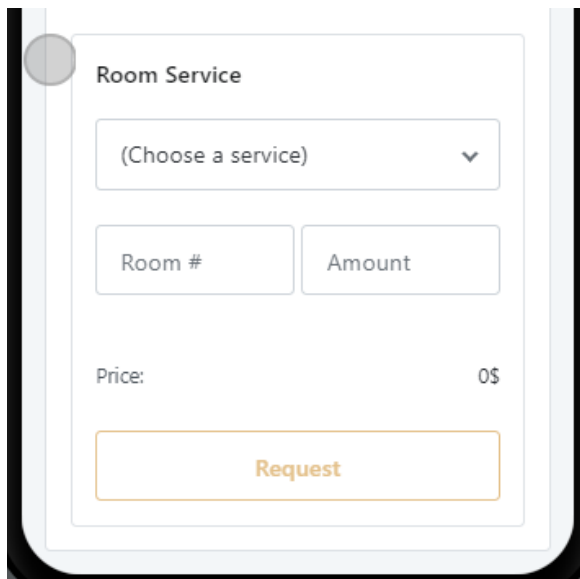
In this bonus section of the assignment, we want to create a Room Service functionality. To do that, we need to first change the Data Model to support the Room Service.



There is an Excel file in the Resources folder, with important data that can be used to populate the RoomService Entity.

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 mobile application interface for a 'Room Service' section. It features a title bar with a back arrow and the text 'Room Service'. Below the title bar is a dropdown menu with the placeholder text '(Choose a service)'. Underneath the dropdown are two input fields: 'Room #' and 'Amount'. Below these fields is a label 'Price:' followed by a text input field containing '0\$'. At the bottom of the form is a large orange button with the text 'Request'.