

Technical Design Document (TDD)

Power Apps Prototype - Hotel Booking System

1. Document Information

Item	Description
Project Name	Hotel Booking System
Application Type	Microsoft Power Apps Canvas App
Version	1.0
Author	Adeline Lee
Date	8 September 2026
Purpose	Prototype application for capturing and storing hotel booking requests

2. Project Overview

2.1 Purpose

The Hotel Booking System is a Power Apps Canvas application designed to allow users to:

- Enter customer contact details.
- Select guest, child, room, view and room-type requirements.
- Choose check-in and check-out dates.
- Record special requests or guest notes.
- Save a booking record to an Excel Online table.

2.2 Objectives

- Provide a clear, single-screen booking interface.
- Validate the customer name and email before confirmation.
- Store consistent booking data for later review and processing.
- Reduce manual transcription of customer booking information.

3. System Architecture

3.1 Technology Stack

Component	Technology / Implementation
Front end	Microsoft Power Apps Canvas App
Data storage	Excel Online table
Data connector	Excel Online (Business) connector
Primary action	SaveAs button OnSelect property

3.2 Logical Workflow

1. The user enters customer and booking details on Screen1.
2. The SaveAs button checks whether the customer name is blank and whether the email format is valid.
3. The Patch function creates a new record using Defaults(HotelBookTable1).
4. The Excel table stores the submitted booking details as a new row.

4. Data Model Design

4.1 Booking Entity

Table: HotelBookTable1

Field Name	Data Type	Power Apps Source
CustomerName	Text	CusName.Text
CustomerPhone	Text	CusPhone.Text
CustomerEmail	Text	CusEmail.Text
NumGuest	Number	GuestNumber.Selected.Value
NumKids	Number	KidNumber.Selected.Value
NumRooms	Number	NumberRooms.Selected.Value
RoomView	Text	RoomView.Selected.Value
RoomType	Text	RoomType.Selected.Value
CheckInDate	Date	CheckInDate.SelectedDate
CheckOutDate	Date	CheckOutDate.SelectedDate
SpecialRequests	Text	SpecialRequestsInput.Text

4.2 Example Records

A	B	C	D	E	F	G	H	I	J	K	L
CustomerName	CustomerPhone	CustomerEmail	NumGuest	NumKids	NumRooms	RoomView	RoomType	CheckInDate	CheckOutDate	SpecialRequests	PowerAppsID
Kelly	81234567	Kelly@gmail.com	1	0	1	Seaview	Twin	8/8/2026	10/8/2026	None	96030e83-d16:
Harriet	98765432	harriet@gmail.com	1	1	1	Sea View	Single	8/4/2026	8/6/2026		ec2b14cd-d3c1
Harriet	98765432	harriet@gmail.com	1	1	1	Sea View	Single	8/4/2026	8/6/2026		fa9b701d4624

Figure 1. HotelBookTable1 records in Excel Online

The table contains customer, occupancy, room preference, stay-date and request fields. The additional Power Apps identifier column is maintained by the platform and is not populated by the Patch formula.

5. User Interface Design

The prototype contains one primary screen: Screen1 - Hotel Booking Screen.

5.1 Screen1 - Hotel Booking Screen

Figure 2. Hotel booking entry screen in Power Apps Studio

Purpose: Capture customer details, room preferences, stay dates and special requests, then confirm the booking.

Screen Name: Screen1

Primary Action: SaveAs / Confirm Booking

5.2 Control Inventory

Control Name	Type	Purpose
MainTitle	Label	Displays the Hotel Booking System title.
CusName	Text input	Captures the customer name.
CusPhone	Text input	Captures the customer phone number.
CusEmail	Text input	Captures the customer email address.
GuestNumber	Dropdown	Selects the number of guests.
KidNumber	Dropdown	Selects the number of children.
NumberRooms	Dropdown	Selects the number of rooms.
RoomView	Dropdown	Selects the preferred room view.
RoomType	Dropdown	Selects the preferred room type.
CheckInDate	Date picker	Selects the check-in date.
CheckOutDate	Date picker	Selects the check-out date.
SpecialRequestsInput	Text input	Captures optional requests or guest notes.
SaveAs	Button	Validates inputs and creates a booking record.

5.3 App Tree Structure

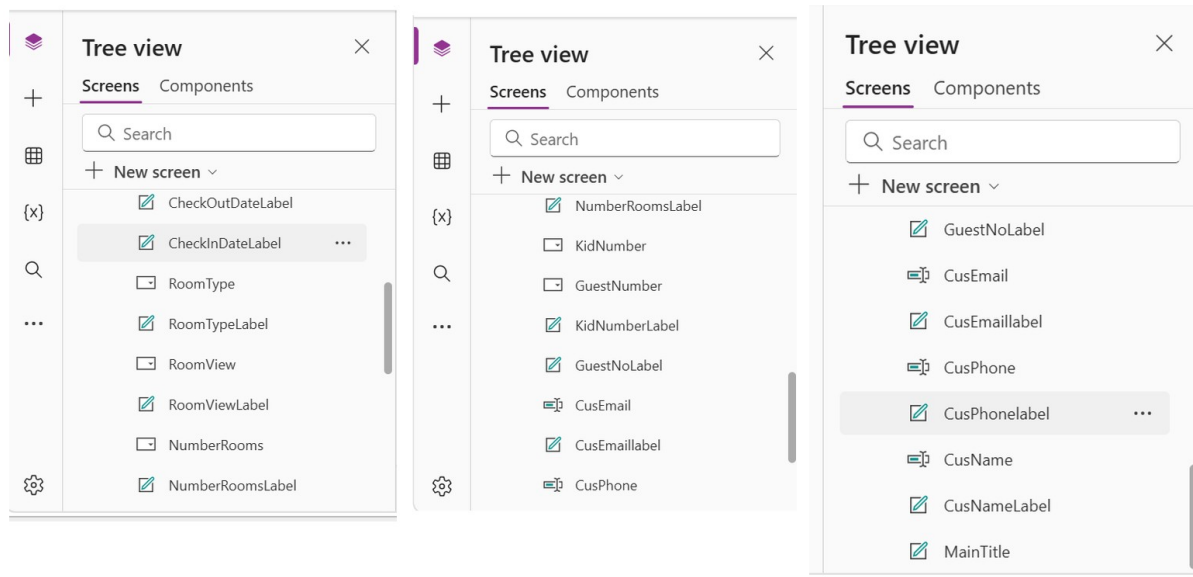


Figure 3. Selected controls shown in the Power Apps Tree view

6. Save Button Logic

6.1 Validation Rules

Condition	User Notification	Type
CusName.Text is blank	Please enter the Customer Name	Error
CusEmail.Text does not match Match.Email	Please enter the valid email address	Error

6.2 OnSelect Formula

```
If(
    IsBlank(CusName.Text),
    Notify(
        "Please enter the Customer Name",
        NotificationType.Error
    )
);

If(
    !IsMatch(CusEmail.Text, Match.Email),
    Notify(
        "Please enter the valid email address",
        NotificationType.Error
    )
);

Patch(
    HotelBookTable1,
    Defaults(HotelBookTable1),
    {
        CustomerName: CusName.Text,
        CustomerPhone: CusPhone.Text,
        CustomerEmail: CusEmail.Text,
        NumGuest: GuestNumber.Selected.Value,
        NumKids: KidNumber.Selected.Value,
        NumRooms: NumberRooms.Selected.Value,
        RoomView: RoomView.Selected.Value,
        RoomType: RoomType.Selected.Value,
        CheckInDate: CheckInDate.SelectedDate,
        CheckOutDate: CheckOutDate.SelectedDate,
        SpecialRequests: SpecialRequestsInput.Text
    }
);
```

6.3 Patch Operation

Patch creates one new record in HotelBookTable1. Defaults(HotelBookTable1) supplies a blank record, and the field-value object maps each Power Apps control to its matching Excel column.

7. Functional Test Scenarios

Test ID	Scenario	Expected Result
T01	Customer name is blank	Name error notification appears.
T02	Email is not in a valid format	Email error notification appears.
T03	All booking fields are valid	A new Excel row is created with matching values.
T04	Dates and dropdown values are selected	Selected values are stored in the correct columns.
T05	Special request is entered	The request is saved in SpecialRequests.

8. Technical Considerations

- Store the Excel workbook in OneDrive for Business or SharePoint and format the booking range as an Excel table.
- Keep the Power Apps field names aligned with the Excel column headers.
- Validate that CheckOutDate is later than CheckInDate before saving.
- Consider adding a success notification and Reset controls after a successful Patch.
- For larger or multi-user production solutions, consider Dataverse or SharePoint instead of Excel Online.

8.1 Important Validation Observation

Current behaviour: The two validation If statements display errors, but they do not stop the later Patch statement. As written, an invalid record may still be saved. For production use, place Patch in the valid branch of a combined If condition.

8.2 Recommended Save Pattern

```
If(
    IsBlank(CusName.Text) || !IsMatch(CusEmail.Text, Match.Email),
    Notify("Please enter a customer name and valid email address", NotificationType.Error),
    Patch(HotelBookTable1, Defaults(HotelBookTable1), { ... });
    Notify("Booking saved successfully", NotificationType.Success)
)
```

End of Technical Design Document