

Problem Statement

Managing accommodations for diverse groups such as tourists, students, and working professionals is often challenging due to fragmented platforms and lack of personalization.

- **Tourists & Travelers** need short-term stays with flexibility.
- **Students & Professionals** require long-term, affordable, and reliable housing like PGs, hostels, or rentals.
Traditional property management systems lack centralized communication, efficient booking handling, transparent ratings/reviews, and host–guest coordination.

This results in:

- Inefficient property discovery
- Lack of trust between host and guest
- Manual management of payments & complaints
- Limited visibility of guest/host performance



StayEase CRM solves these issues by creating a **Salesforce-powered centralized platform** for property management, booking, guest handling, and payments.

Industry Type

- **B2C (Business-to-Consumer):**

StayEase directly connects property hosts with end-users (tourists, students, working professionals).

Target Users

1. **Hosts** – Individuals or businesses listing their properties.
2. **Guests** – Tourists, students, or professionals booking stays.
3. **Admins** – Oversee operations, resolve disputes, and manage CRM functionalities.

Key Objectives

- Centralize property listings, bookings, payments, and complaints.
- Provide a **trust-building mechanism** via guest ratings and host performance.
- Offer different stay types: **short-term (Airbnb-like)** and **long-term (student/professional rentals)**.
- Enhance **customer satisfaction** with transparent processes and automation

Value Proposition

- For **Guests** → Easy property discovery, secure payments, and transparent ratings.
- For **Hosts** → Better visibility, guest management, and simplified bookings.
- For **Admin/Organization** → A scalable CRM system on Salesforce that tracks all activities in one place.

◆ Phase 1: Problem Understanding & Scope

Introduction

Accommodation management is a common challenge for multiple groups:

- **Tourists & Travellers** – need short-term stays for leisure, business trips, or events.
- **Students & Working Professionals** – require long-term affordable rentals such as PGs, hostels, or shared flats.

The **existing market** has several limitations:

- **Unverified Listings** – Many platforms allow fake or outdated property details.
- **Broker Dependence** – Manual, broker-driven processes inflate costs.
- **Communication Gaps** – Hosts and guests lack structured communication channels.
- **Low Transparency** – Guests face uncertainty regarding booking status, payments, and reviews.
-

Proposed Solution

StayEase CRM is a Salesforce-based CRM project designed to create a structured and trustworthy **accommodation and rental platform**. It takes inspiration from Airbnb's short-term rental model while extending to long-term housing for students and professionals.

Key Features of the Solution:

- Centralized database of verified properties.
- Seamless booking flow for **short-term** and **long-term** stays.
- Role-based data access for **Admin, Hosts, Guests, and Managers**.
- Automation for booking status, payments, and complaint resolution.
- Reports and dashboards to monitor occupancy, revenue, and service quality.



Objectives

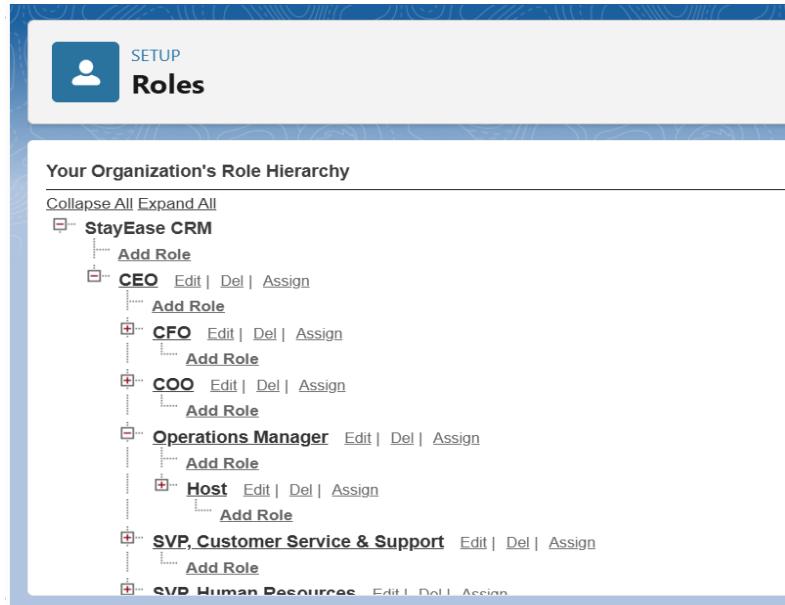
1. Create a **unified accommodation platform** with transparent workflows.
2. Support **short-term bookings** for tourists and **long-term rentals** for students/professionals.
3. Provide **role-based security** for data integrity.
4. Use Salesforce automation (Flows, Validation Rules, Approvals) to improve efficiency.
5. Deliver **business insights** through dashboards and reports.

◆ Phase 2: Org Setup & Security

To implement the foundation of StayEase CRM, **roles, profiles, org-wide defaults, and permissions** are configured.

2.1 Roles (Role Hierarchy)

- **CEO** → Top-level access; visibility into all records, reports, and dashboards.
- **Operations Manager** → Oversees all bookings, complaints, and escalations across the org.
- **Host** → Can only manage their own properties, bookings, and related complaints.
- **Guest (Traveller/Student/Professional)** → Can book properties and raise complaints related to their own bookings.
- **System Administrator** → Full access, controls setup and configurations (you).



2.2 Profiles

Profiles define **what a user can do** inside Salesforce.

- **Host Profile**
 - Can create, view, and edit only their properties.
 - Can view bookings related to their properties.
 - Limited access to complaints (only related ones)

[Edit](#) | [Del](#) | ... [Host Profile](#)

Salesforce Platform



- **Guest Profile**
 - Can create and manage only their own bookings.
 - Can raise complaints for their own bookings/properties.
 - Read-only access to property listings.

[Edit](#) | [Del](#) | ... [Guest Profile](#)

Salesforce Platform



- **Operations Manager Profile**
 - Full access to all bookings and complaints.
 - Can intervene/escalate complaints.
 - Cannot perform system-level configurations.
- **CEO Profile**
 - Read-only access to all data.
 - Access to dashboards and reports for strategic decision-making.
- **System Admin Profile**
 - Full CRUD (Create, Read, Update, Delete) permissions on all objects.
 - Handles automation, security, and backend setup.



2.3 Org-Wide Defaults (OWD)

Default record-level security settings:

- **Property_c** → Private (Hosts can only see their properties).
- **Booking_c** → Private (Guests can see their own bookings, Hosts can see bookings related to their properties).
- **Complaint_c** → Private (Guests see their complaints; Ops Manager sees all).
- **Payment_c** → Private (Linked to bookings; visible only to Guest, Host, and Ops Manager).

2.4 Sharing Rules

- **Hosts** get automatic access to bookings made on their properties.
- **Ops Managers** get access to all complaints and escalated bookings.
- **CEO** gets read-only access to all objects for reporting.

2.5 Permission Sets

Additional flexibility beyond profiles:

- **Host_Extra_Access** → If needed, allow hosts to see limited guest details (like name & contact for bookings).
- **Ops_Manager_Analytics** → Access to advanced dashboards and reports.

❖ Outcome of Phase 2:

At this stage, the org is structured with a secure role hierarchy, profiles, permissions, and OWD rules. Each actor (Admin, CEO, Operations Manager, Host, Guest) has **only the data visibility and access they need**, ensuring data privacy, integrity, and smooth system operation.

❖ Phase 3 Objects & Relationships

StayEase CRM

Guests ▾ Hosts ▾ Bookings ▾ Complaints ▾ Payments ▾ Properties ▾

Guests

Recently Viewed ▾

1 item • Updated a minute ago

	Guest ID
1	G-0001

1. Host Object

Purpose: Represents the property owner.

Fields:

- Name (Text) – Name of the owner.
- Host id – Auto Number (H - {0000})
- Contact Email – Email field type
- Phone Number – Phone field type
- Total Properties – Number field (will auto update on the addition or deletion of properties associated with the host)

Relationship : Master-Detail relationship with property. Host – master & Property – detail

Host
H-0004

Related Details

Name	Ram Kumar	Owner	Shruti Deshpande
Host ID	H-0004		
Contact Email	hostemail@gmail.com		
Contact Phone	(928) 499-9990		
Total Properties	2		
Created By	Shruti Deshpande, 9/20/2025, 12:48 AM	Last Modified By	Shruti Deshpande, 9/20/2025, 1:02 AM

2. Property

Purpose: Represents the accommodation available for booking.

Fields:

- Property ID – Auto Number (P-{0000}).
- Name (Text) – Name of the property.
- Type (Picklist – villa , resort , hostel, PG, flat)
- Location (Text).
- Rent (Currency).
- Rent type (Picklist field – Monthly rent , pre night rent)
- Availability (Checkbox).
- Amenities (Picklist Multiselect – Wifi , AC, Parking , kitchen, cook, semi/ fully furnished)
- Max Occupancy (Number).
- Booking Ratings Sum (Number, hidden).
- Booking Ratings Count (Number, hidden).
- Average Rating (Formula).

Relationship:

- Master-Detail with **Host** (Host → Property).
- Lookup with **Booking** (Booking → Property).
- Lookup with **Complaint** (Complaint → Property).

Validation Rules:

The screenshot shows the Salesforce Object Manager interface for the 'Property' object. On the left, there is a sidebar with various tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, and Record Types. The 'Validation Rules' tab is selected. The main area displays a table titled 'Validation Rules' with one item listed. The table has columns: RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The single rule listed is 'Amount_GreaterThan_0', located under 'Rent', with the message 'The rent amount should be greater than zero.' The 'ACTIVE' column shows a checked checkbox, and the 'MODIFIED BY' column shows 'Shruti Deshpande, 9/19/2025, 2:53 AM'. A 'New' button is visible in the top right corner of the table header.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Amount_GreaterThan_0	Rent	The rent amount should be greater than zero.	✓	Shruti Deshpande, 9/19/2025, 2:53 AM

3. Booking Object

Purpose: Represents a reservation made by a Guest.

Fields:

- Booking ID – Auto Number (B-{0000}).
- Booking Type (Picklist)
- Guest (Lookup → Guest).
- Guest Rating (Number)
- Number of guests (Number)
- Total Rent (Currency – total rent to be paid)
- Property (Master-Detail → Property).
- Check-in Date (Date).
- Check-out Date (Date).
- Advance amount (Currency – only for students and working professionals)
- Status (Picklist – Confirmed, Cancelled, Completed).

Fields & Relationships 13 Items, Sorted by Field Label				
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD
	Advance Amount	Advance_Amount_c	Currency(16, 2)	
	Booking ID	Name	Auto Number	✓
	Booking Type	Booking_Type_c	Picklist	
	Check In Date	Check_In_Date_c	Date	
	Check Out Date	Check_Out_Date_c	Date	
	Created By	CreatedByld	Lookup(User)	
▼	Guest	Guest_c	Lookup(Guest)	✓

Relationship:

- Lookup with **Guest**.
- Master-Detail with **Property**.
- Master-Detail with **Payment** (Booking → Payment).

Validation Rules:

The screenshot shows the Salesforce Setup interface for the Object Manager. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled 'Validation Rules' under the 'Booking' object. It displays five validation rules, each with a rule name, error location, message, active status, and modified by user. A 'New' button is visible in the top right corner of the list.

Rule Name	Error Location	Error Message	Active	Modified By
Advance_amount_for_longTerm_stay	Top of Page	Advance Amount is mandatory for Student and Working Professional bookings	✓	Shruti Deshpande, 9/20/2025, 12:32 AM
Book_for_longterm_stay	Check Out Date	Check-Out Date is required for Tourist bookings	✓	Shruti Deshpande, 9/19/2025, 3:39 AM
CheckInBeforeCheckOut	Top of Page	Check-In date must be before Check-Out date.	✓	Shruti Deshpande, 9/17/2025, 8:15 AM
Number_of_Guests	Number of Guests	Number of guests should be 1 or more	✓	Shruti Deshpande, 9/19/2025, 2:57 AM
OccupancyLimit	Number of Guests	Number of guests exceeds property maximum occupancy.	✓	Shruti Deshpande, 9/17/2025, 8:16 AM

4. Guest Object

Purpose: Represents the customer who books a property.

Fields:

- Full Name (Text) – Guest's full name.
- Guest ID – Auto Number (G-{0000}).
- Email – Email field type.
- Phone – Phone field type.
- Guest Type – Picklist (Tourist, Student/Working Professional).

The screenshot shows the Salesforce Setup interface for the Object Manager. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled 'Details' under the 'Guest' object. It shows the object's description, API name (Guest__c), and various settings like enable reports, track activities, and deployment status. An 'Edit' and 'Delete' button are located in the top right corner.

Relationship:

- Lookup relationship with **Booking** (Booking → Guest).

Validation Rules:

- None specific at Guest level.

5. Payment

Purpose: Represents payments made for a booking.

Fields:

- Payment ID – Auto Number (Pay-{0000}).
- Booking (Master-Detail → Booking).
- Amount (Currency).
- Transaction Date (Date).
- Payment Status (Picklist – Pending, Completed, Failed).

The screenshot shows the 'Payment' object setup page in the Salesforce Object Manager. The left sidebar lists standard object settings like Details, Fields & Relationships, Page Layouts, etc. The main 'Details' tab is selected, showing the API Name as 'Payment_c', which is custom and checked. Singular and Plural labels are set to 'Payment' and 'Payments' respectively. A description states: 'This object has all the payment details of the bookings.' On the right, there are checkboxes for enabling reports, tracking activities, and field history, all of which are checked. Deployment status is set to 'Deployed'. Help settings point to the standard Salesforce help window.

Relationship:

- Master-Detail with **Booking** (Booking → Payment).

Validation Rules:

The screenshot shows the 'Validation Rules' section for the Payment object. It displays a single rule named 'Transction_Date_Before_Checkin'. The rule specifies that the Transaction Date cannot be after the Check-In Date of the Booking. The rule is active and was modified by Shruti Deshpande on September 18, 2025, at 8:55 AM.

Rule Name	Error Location	Error Message	Active	Modified By
Transction_Date_Before_Checkin	Transaction Date	Transaction Date cannot be after the Check-In Date of the Booking.	✓	Shruti Deshpande, 9/18/2025, 8:55 AM

5. Complaint

Purpose: Represents complaints raised by guests.

Fields:

- Complaint ID – Auto Number (C-{0000}).
- Guest (Lookup → Guest).
- Property (Lookup → Property).
- Complaint Type (Picklist – Maintenance, Cleanliness, Noise, Other).
- Status (Picklist – New, In Progress, Resolved, Closed).
- Description (Long Text Area).

Relationship:

- Lookup with **Guest**.
- Lookup with **Property**.

Validation Rules:

- None specific.

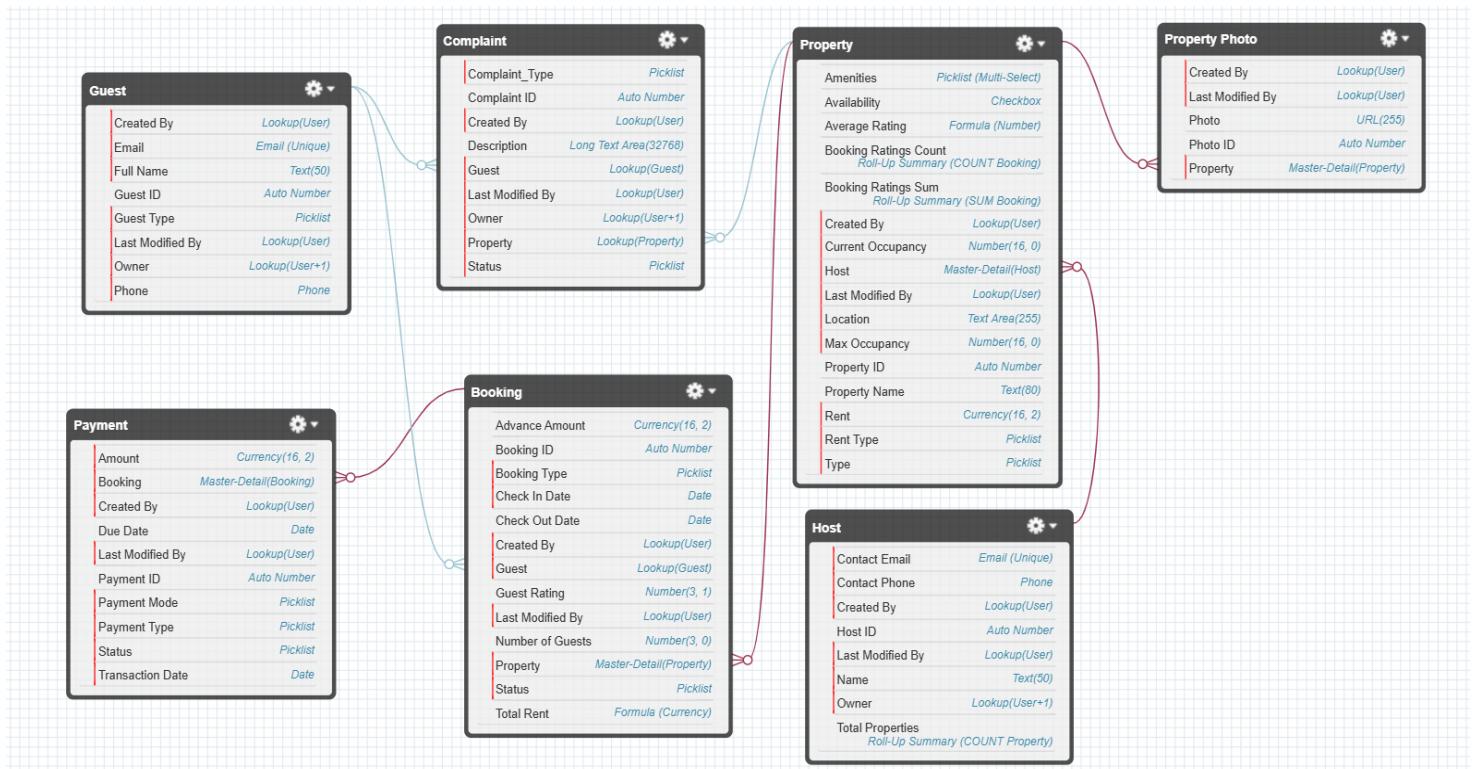


fig : Schema Diagram (Form Schema Builder)

Phase 4 Process Automation (Admin)

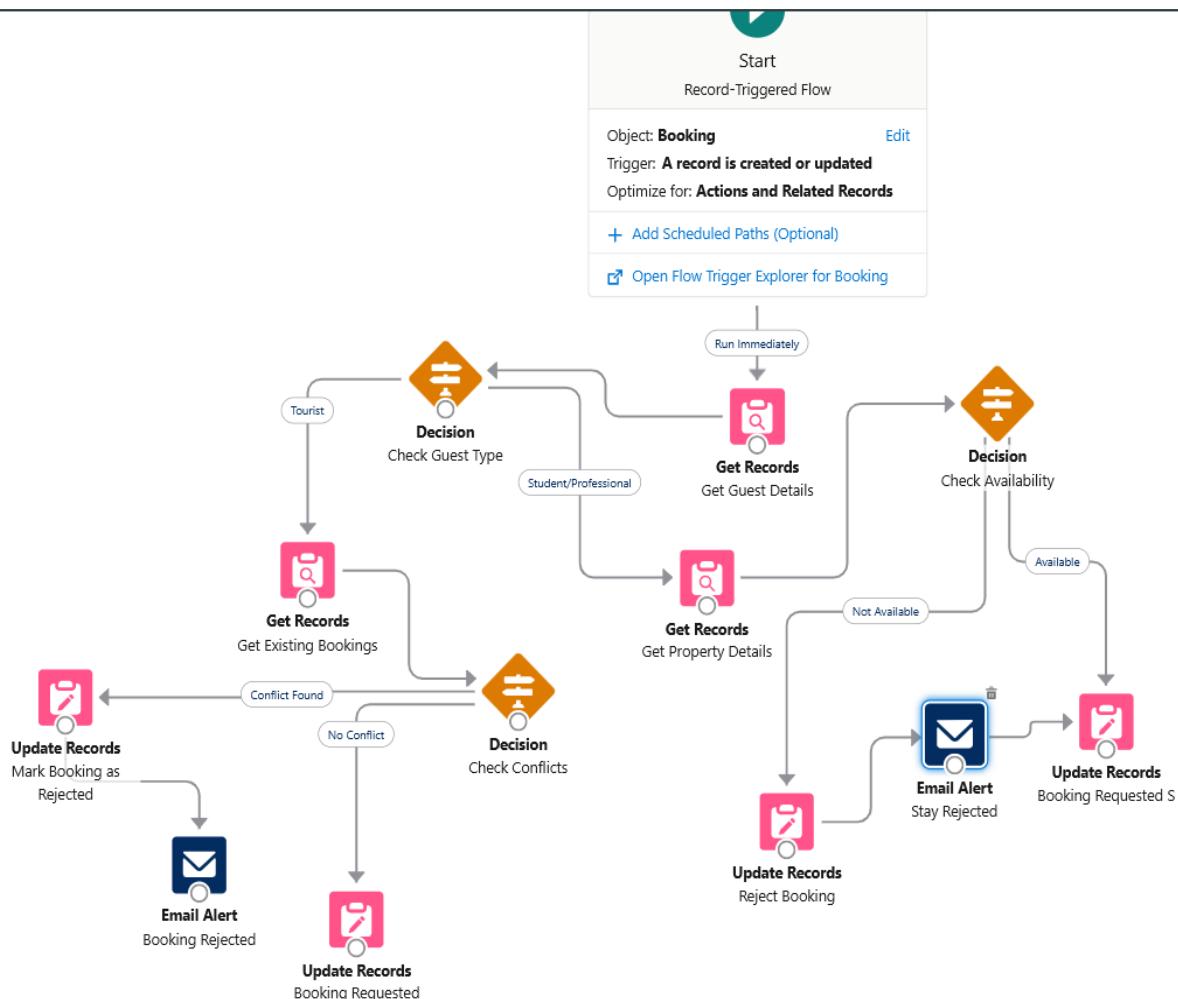
Flow 1: Booking Availability Check

Type: Record-Triggered Flow (on Booking)

Trigger: After insert or update

Logic:

- Get Guest Type (Tourist vs Student/Professional).
- If Guest is **Tourist**:
 - Check if selected property already has overlapping bookings.
 - If overlap → Set Status = *Rejected*.
 - If no overlap → Set Status = *Requested* (*Confirmed only after payment*).
- If Guest is **Student/Professional**:
 - Check Max Occupancy.
 - If Current Occupancy < Max Occupancy → Confirm.
 - Else → Reject.

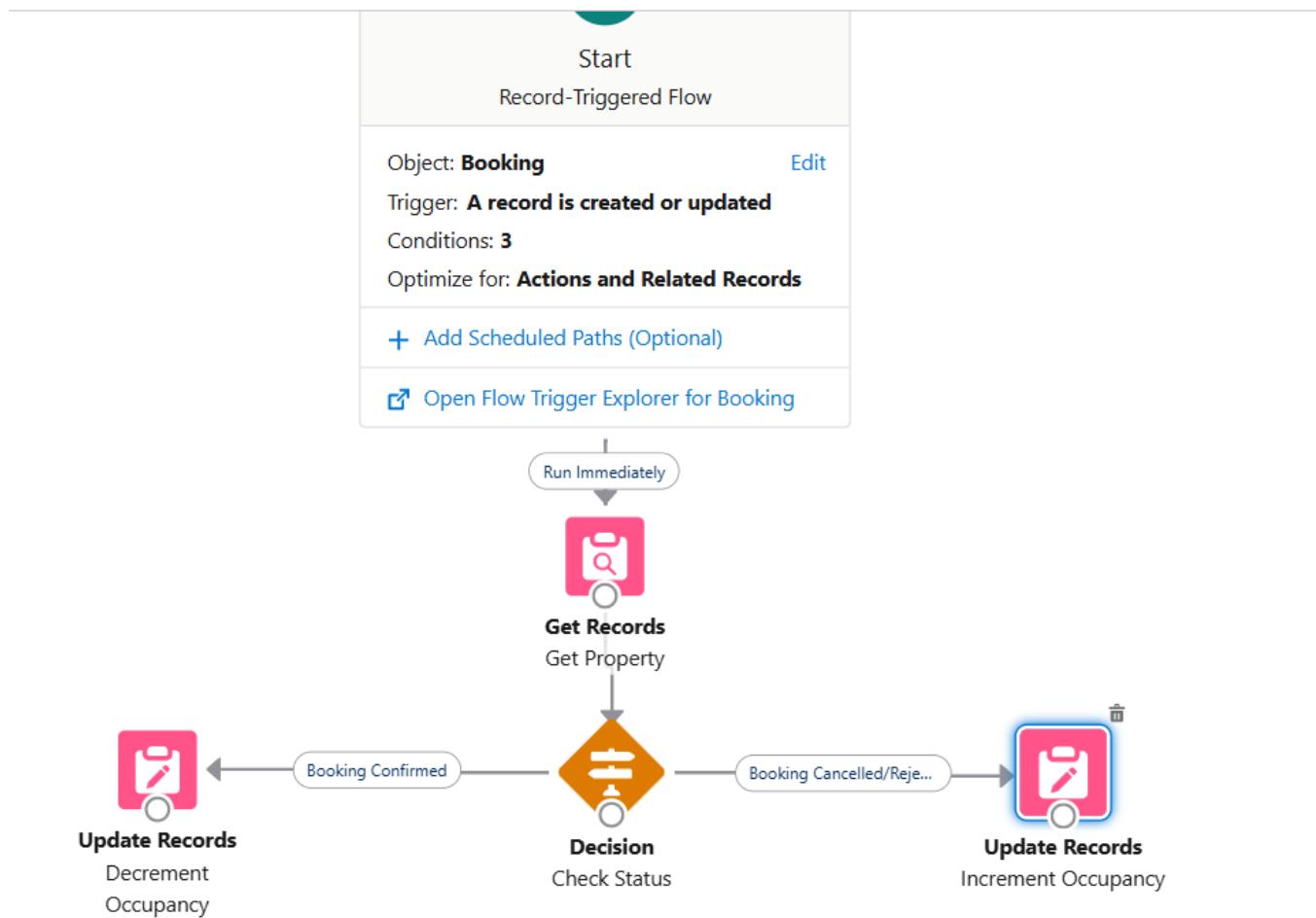


Flow 2: Occupancy Management

Type: Record-Triggered Flow (on Booking, After Save)

Logic:

- On *Confirmed* Booking → Increment Current_Occupancy__c of Property by number of guests.
- On *Cancelled* Booking → Decrement Current_Occupancy__c of Property.

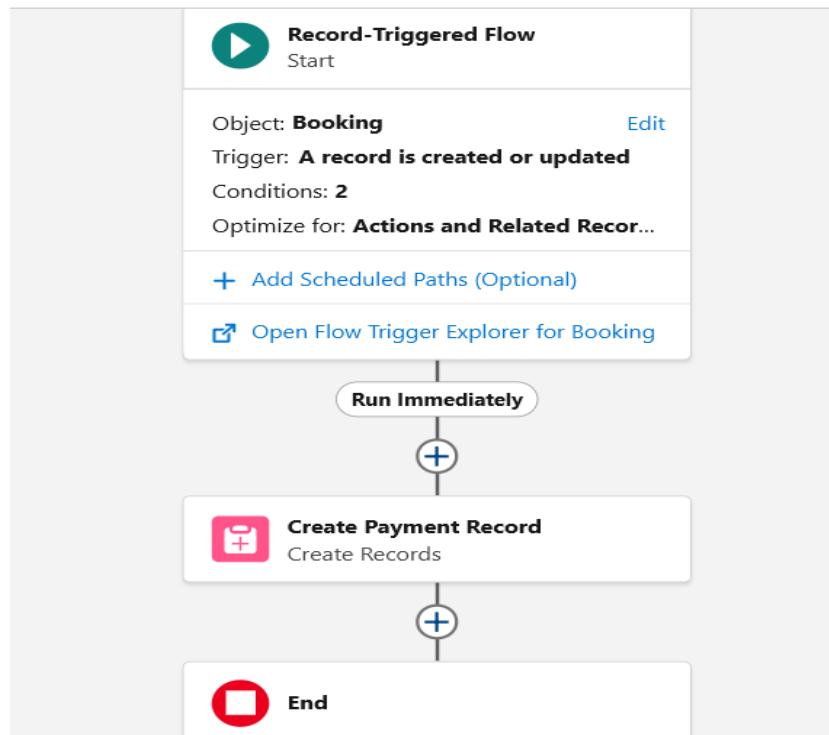


Flow 3: Payment Creation & Notifications

Type: Record-Triggered Flow (on Booking, After Save)

Logic:

- When Booking status = *Confirmed*:
 - Automatically create a **Payment__c** record.
 - Map fields:
 - Booking → Payment.Booking__c
 - Guest → Payment.Guest__c
 - Amount → Booking.Amount__c
 - Transaction Date = TODAY()
- Send **Email Notifications**: (To be implemented)
 - If Confirmed → Send *Confirmation Email Template* (separate templates for Tourist and Student/Professional).
 - If Rejected → Send *Rejection Email Template*.



Summary :

- Phase 3 introduced stricter data model rules, validation for guest types, and default booking behaviors.
- Phase 4 automated key processes: booking conflict handling, occupancy tracking, auto payment creation, and email notifications.
- Together, these ensure smoother property booking, guest experience, and host management.

