Tours and Travels Platform Development Using CRM-Salesforce

Retrieved from *https://3.imimg.com/data3/BE/DV/MY-13222257/tours-and-travels-services.jpg*

A Capstone Project

Presented to

Salesforce and Smartbridge

By

Mark Jerome B. Santos

July 18, 2025

I. Project Overview

The Tours & Travels CRM project is aimed at developing a Salesforce-based Customer Relationship Management system tailored for global travel agencies. It addresses operational challenges such as manual bookings, fragmented communication, and lack of visibility across customer interactions, finance, and employee coordination.

By leveraging Salesforce's automation, data modeling, UI customization, and reporting capabilities, the CRM will enable travel companies to manage bookings, communicate with customers in real time, assign agents and guides, automate follow-ups, and gain insights through dashboards — all within a centralized and secure system.

II. Objectives

The core objective of this project is to design and implement a CRM system that streamlines travel agency operations from inquiry to feedback, reduces manual workload, and enhances customer satisfaction. By aligning Salesforce tools with real-world business challenges, the system will:

- Improve customer relationship management through structured onboarding, booking, and feedback workflows.
- Enable streamlined operations by automating repetitive tasks and enabling collaboration between teams.

- Provide business intelligence through reporting on KPIs such as monthly revenue, retention rates, and service quality.
- Ensure data security and role-based access for users such as agents, guides, finance, and administrators.
- Provide a user interface that is simple and useful for all types of staff.

III. Phase 1:Requirement Analysis & Planning

Objective

This phase focuses on understanding the business domain of the Tours & Travels industry, identifying pain points in existing operations, and clearly defining the functional and non-functional requirements for a CRM system. The goal is to ensure the CRM solution is user-centric, scalable, and aligned with real-world business needs.

1. Understanding Business Requirements

Industry Context

Travel agencies globally face operational complexity due to scattered processes, manual bookings, poor visibility across teams, and lack of automation. These inefficiencies impact customer satisfaction and scalability. A centralized CRM built on Salesforce can help overcome these limitations and digitize core operations.

Stakeholders Analyzed

- Travel Agents: Manage tour bookings, handle inquiries, coordinate with guides.
- Customer Service Managers: Address customer issues, monitor service quality, manage post-trip feedback.
- Finance Teams: Track revenue, manage invoices, oversee payment confirmation.
- System Admins: Configure the Salesforce environment, manage roles and access.
- Customers (End Users): Book trips, make payments, track booking status, and provide feedback.

Points in Existing Systems

Derived from industry analysis, end-user behavior patterns, and stakeholder process studies.

Challenge	**Description**
Manual Booking Processes	Booking via phone, email, or spreadsheets leads to delays, errors, and duplication
Delayed Communication	Customers are not informed in real-time about bookings, payment updates, or changes
Lack of Feedback Loop	Feedback is not consistently captured or tracked, leading to lost insights

Disjointed Collaboration	Travel agents, guides, and finance teams operate in silos without a common platform
No Real-Time Reporting	Agencies struggle to monitor performance metrics like customer retention and monthly revenue

Approach to Requirement Gathering

- Interview-Based Analysis of user roles (simulated via research tools like ChatGPT and domain literature).
- Study of Real-Time Use Cases using publicly available travel platforms and CRM evaluations.
- Reference Materials:
  - Salesforce Official Documentation
  - Trailhead Modules (CRM Basics, Travel Industry Use Cases, Data Modeling)
  - Google Search and Industry Blogs
  - ChatGPT for synthesizing general domain and process knowledge

Key Business Requirements Identified

Category	Requirement
Global Operations	Support customers and travel packages across multiple countries

Booking Flexibility	Allow booking for families, groups, solo, and corporate packages
Automation	Automate booking status updates, payment reminders, feedback collection
Real-Time Communication	Send notifications for booking confirmation, payment status
Task Management	Assign agents and guides for trips using the Salesforce Task object
Reporting & Analytics	Monitor revenue, feedback scores, customer retention metrics
Role-Based Access	Define access by role: agents, guides, admins, finance team

2. Project Scope & Objectives

Scope of the CRM Solution

- Develop a scalable Salesforce CRM tailored for global travel agencies.
- Manage the end-to-end booking lifecycle: inquiry → booking → payment → feedback.
- Implement custom objects, automation, UI, and analytics features.

- Ensure role-based access for different user groups.

3. Gathering & Analyzing User Needs

User Groups Identified

- Customers
- Travel Agent Team
- Tour Guides
- Finance Department
- Admin Team

Key Functional Requirements

Function	Description
Onboarding	Simplified onboarding and package selection for customers
Booking Flow	Transport and accommodation-based booking process
Dynamic Pricing	Adjust pricing based on number of people and membership level
Confirmation	Immediate confirmation via email for booking and payment
Feedback Loop	Follow-up tasks for collecting post-trip feedback

Tools Used During Requirement Gathering

- Google Forms: Simulated method for gathering inputs
- Miro Boards: For visualizing user journeys and identifying process bottlenecks
- User Personas: Applied design thinking principles to guide system design

4. Identified Salesforce Tools & Features

Category	Tools/Features
Custom Objects	Booking__c, TravelPackage__c, Customer__Info__c, Employee__c, BookingPayment__c, Feedback__c, BookingGuest__c
Standard Object	Task (for follow-ups and assignment)
Automation	Flows, Workflow Rules, Process Builders, Approval Processes
Apex	Triggers, Future Methods, Batch Apex, Queueable Apex, Test Classes
UI Components	Lightning App Page, Dynamic Forms, Lightning Web Components
Security	Profiles, Permission Sets, Field-Level Security, Record-Level Sharing
Analytics	Custom Reports and Dashboards

Email Services	Email Alerts and Templates for confirmations and reminders

5. Data & Security Model

Data Model Structure

- Booking is linked to:
    - Customer__c
    - TravelPackage__c
    - Employee__c (as assigned guide)
- TravelPackage__c includes: destinations, transportation, availability, pricing
- BookingPayment__c stores: payment confirmation, amount, and billing details
- Feedback__c and BookingGuest__c support after-trip feedback and guest info

Security Model Highlights

- Role Hierarchy: Travel Manager > Travel Agent > Guide
- Profiles: Separate profiles for Admin, Finance, Agent, Guide, CSR
- Permission Sets: Granular access to reports and object records
- Field-Level Security: Sensitive fields (like billing) restricted to Finance only
- Record-Level Security: Sharing rules allow guides to view only relevant customer records

6. Project Roadmap & Milestones

Phase	Deliverables	Timeline
Phase 1: Requirement Analysis	Stakeholder map, project scope, user needs, Salesforce feature mapping	Week 1
Phase 2: System Design	Data model design, security model (profiles, roles), object relationships	Week 1
Phase 3: Implementation	Object creation, fields, page layouts, validation rules, automations (flows, process builder, approval process), Apex triggers, LWC components	Week 2
Phase 4: Testing & Data Migration	Test cases, unit test classes, bug fixing, duplicate rules, field tracking, data import wizard	Week 3
Phase 5: Deployment & Handover	Reports, dashboards, user training, demo video, documentation	Week 3

IV. Phase 2: Salesforce Development – Backend & Configurations

This phase focused on configuring the backend logic and data model of the Tours and Travels CRM system using Salesforce's declarative and programmatic tools. The development process was aimed at automating business workflows, structuring relational data through custom objects and fields, and enforcing business rules via validation and approval logic.

The following milestones were achieved in this phase:

Milestone 1: Salesforce Account Setup

The project began with creating a Salesforce Developer Edition account. This environment served as the foundation for building, testing, and demonstrating the Tours & Travels CRM solution. This step was crucial to ensure all development and testing happened in a secure and isolated environment.

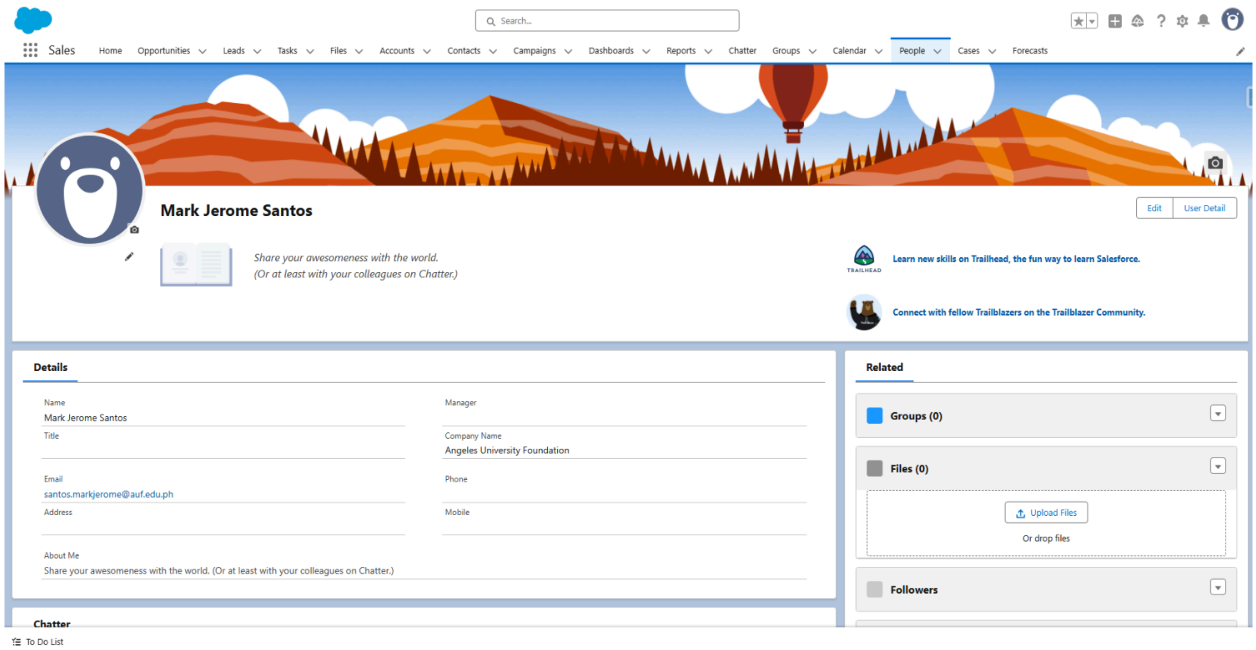


Figure 1. Account Setup

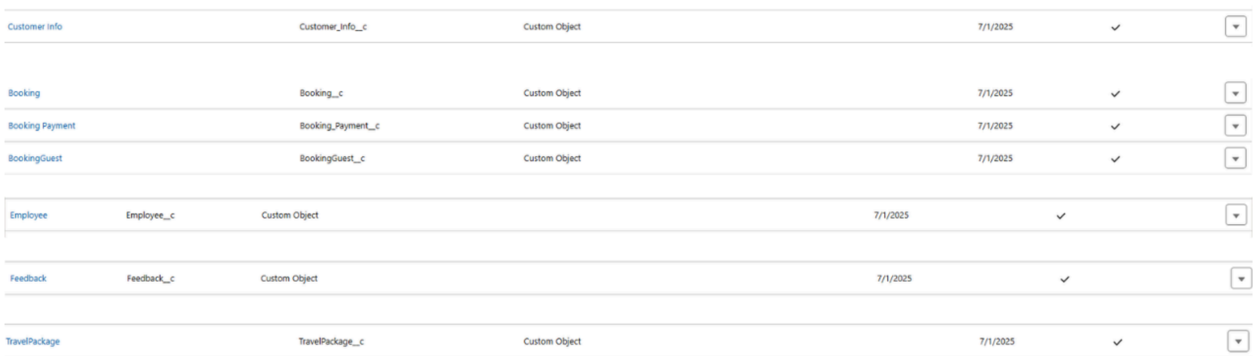
Milestone 2: Custom Object Creation

Custom objects were created to represent the core data entities of the CRM:

- **Customer Info** – stores client details including contact, city, and country.
- **Booking** – captures booking data such as travel details, travelers, cost, and status.
- **BookingGuest** – records individual guest details per booking.
- **TravelPackage** – stores travel package metadata such as destinations, price, and duration.
- **Booking Payment** – manages payment tracking, status, and method for bookings.
- **Employee** – holds internal staff data including role, department, and contact.

- **Feedback** – collects post-trip feedback and ratings from customers.

Each object was configured with record names, custom fields, and relationships for seamless data flow and referential integrity.



Customer Info	Customer_Info__c	Custom Object	7/1/2025	✓
Booking	Booking__c	Custom Object	7/1/2025	✓
Booking Payment	Booking_Payment__c	Custom Object	7/1/2025	✓
BookingGuest	BookingGuest__c	Custom Object	7/1/2025	✓
Employee	Employee__c	Custom Object	7/1/2025	✓
Feedback	Feedback__c	Custom Object	7/1/2025	✓
TravelPackage	TravelPackage__c	Custom Object	7/1/2025	✓

Figure 2. Objects Created

Milestone 3: Tab Creation

Custom tabs were created for each object to enable visibility and access across the Lightning App. Tabs included:

- Customer Info
- Booking
- Booking Guest
- Travel Package
- Booking Payment

- Employee
- Feedback

Each tab was added to a custom Lightning App for easy navigation and use by end-users including travel agents, managers, and admins.

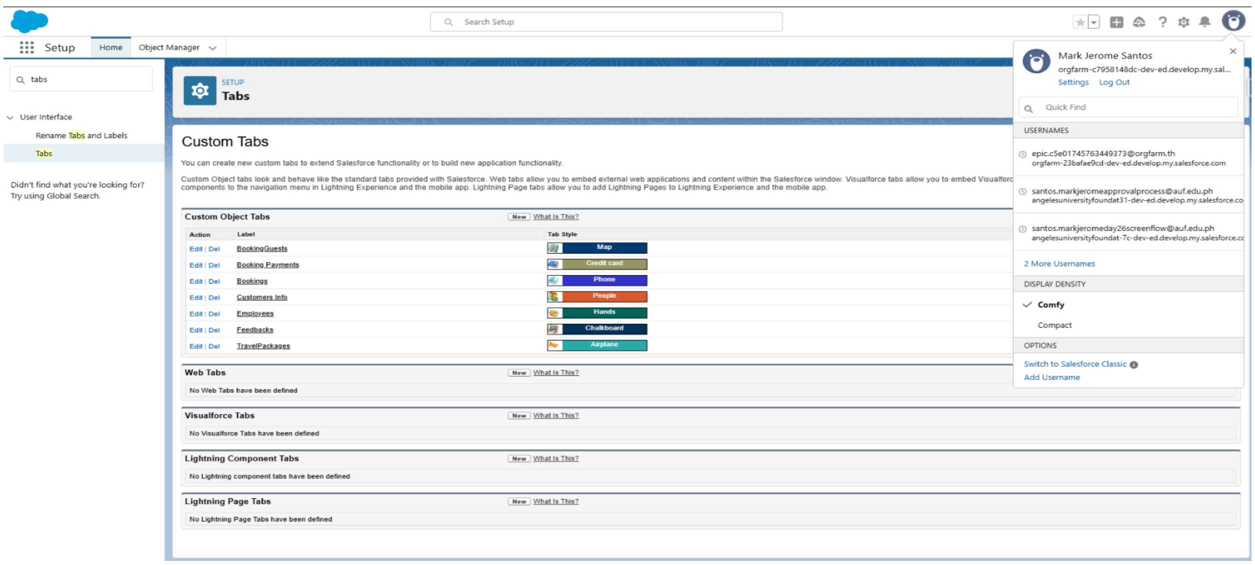


Figure 3. Tabs Created

Milestone 4: Custom Field Creation & Global Picklists

This milestone was one of the most detailed and labor-intensive parts of the development. It involved defining the data structure by creating custom fields, formulas, and picklists for each object.

Customer Info Object:

- Created fields like Email, Phone, Date of Birth, City, and Country.
- Global picklists were used for Country and City.
- A formula field named **Age** was added using the formula:

FLOOR((TODAY() - Date_Of_Birth__c) / 365.25)

Booking Object:

- Lookup Relationships: to Customer Info (Customer) and Employee (Guide_Assigned__c) with lookup filters:
    - Guide Role Assigned = Guide
    - Availability Status = Available
- Formula Fields:
    - Travelling End Date:

      Travelling_Start_Date__c + TravelPackage__r.Duration_in_Days__c

    - Accommodation Cost Per Person Per Day:

      CASE(TEXT(Preferred_Accommodation__c), *"Hotel", 3000,*

"Guest House", 4000,

"Resort", 5000,

"Villa", 6000,

"Cottage", 8000,

"Luxury Hotel", 20000,

0)

- Travel Cost Per Person:

  TravelPackage__r.Price_Per_Person__c

- Total Accommodation Amount:

  *Accomodation_Amount_per_Person_per_Day__c * Number_of_Travelers__c*

- Total Travel Amount:

  *Number_of_Travelers__c * Travel_Cost_Per_Person__c*

- Total Billing Amount:

  Total_Accomodation_Amount__c + Total_Travel_Amount__c

- Picklist Fields:
    - Booking Status
    - Trip Type
    - Preferred Accommodation
    - Membership
    - Preferred Guide Language
- Roll-Up Summary Field:
    - Count of BookingGuest records linked to a Booking

Booking Guest Object:

- Formula Field:
    - Age Category:

IF(Age__c < 12, "Child",

IF(Age__c < 60 , "Adults", "Senior Citizen"))

- **Picklist:** Relation with Customer

Travel Package Object:

- Picklists for: Availability Status, Meals Included
- Multi-Select Picklists: for Membership, Package Type, and Transportation Modes

Employee Object:

- Picklists: Role, Department, Languages Spoken, Country, City, Assigned Region

Booking Payment Object:

- Picklist: Payment Method

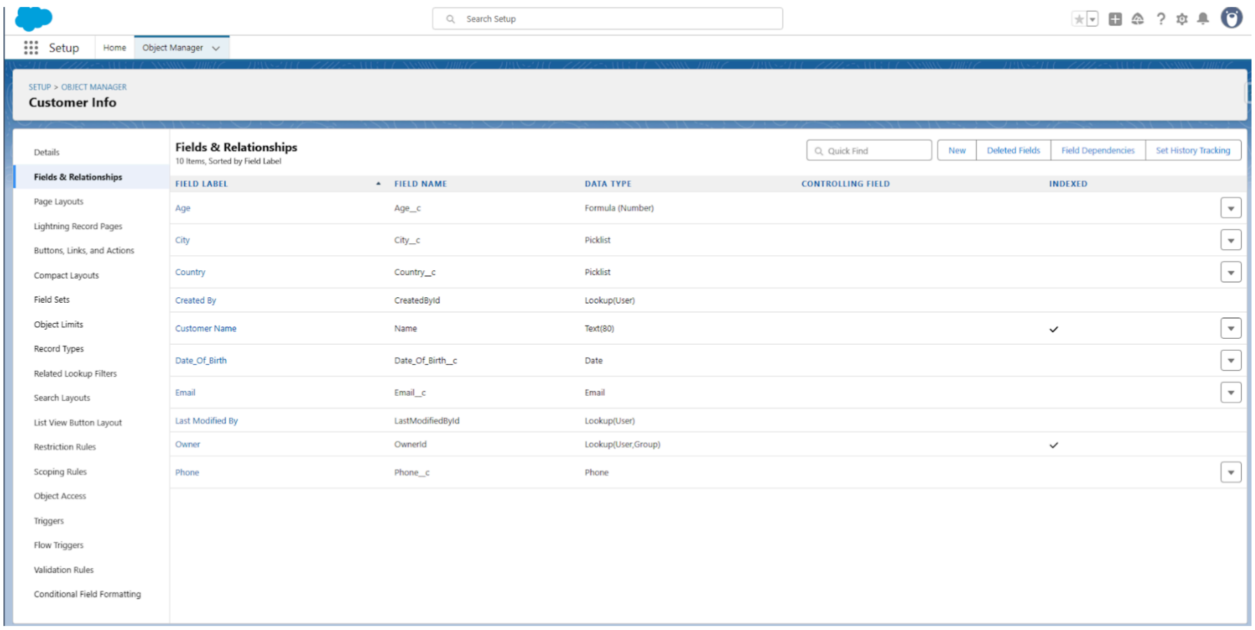


Figure 4. Customer Info Fields & Relationships

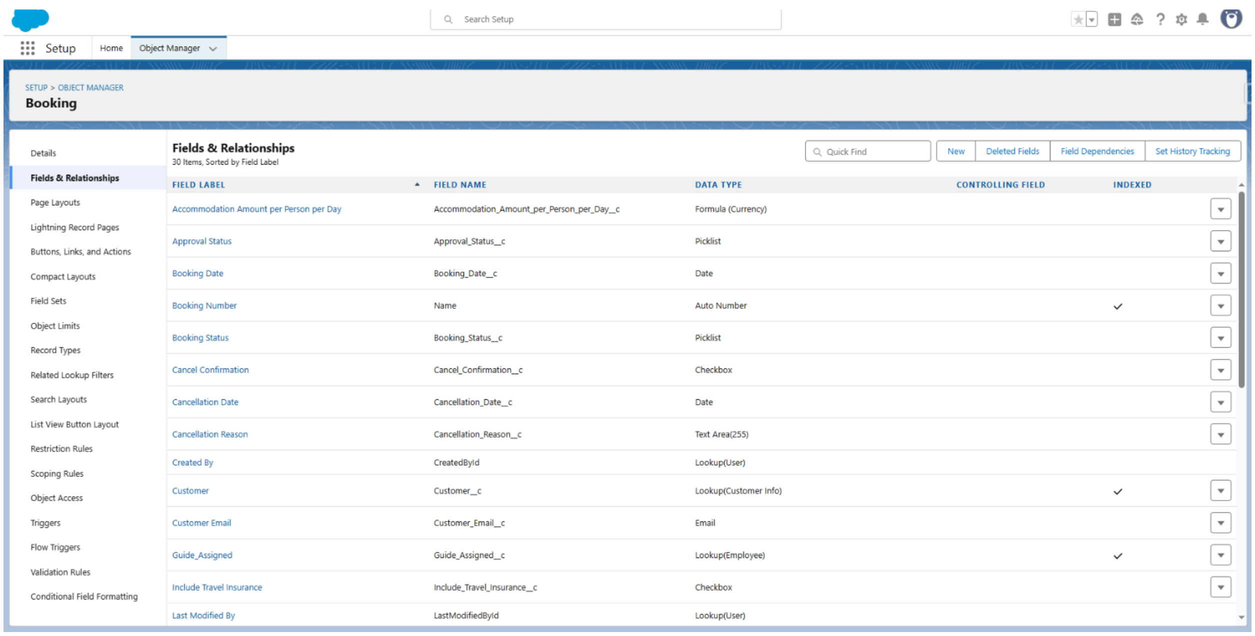


Figure 5. Booking Fields & Relationships

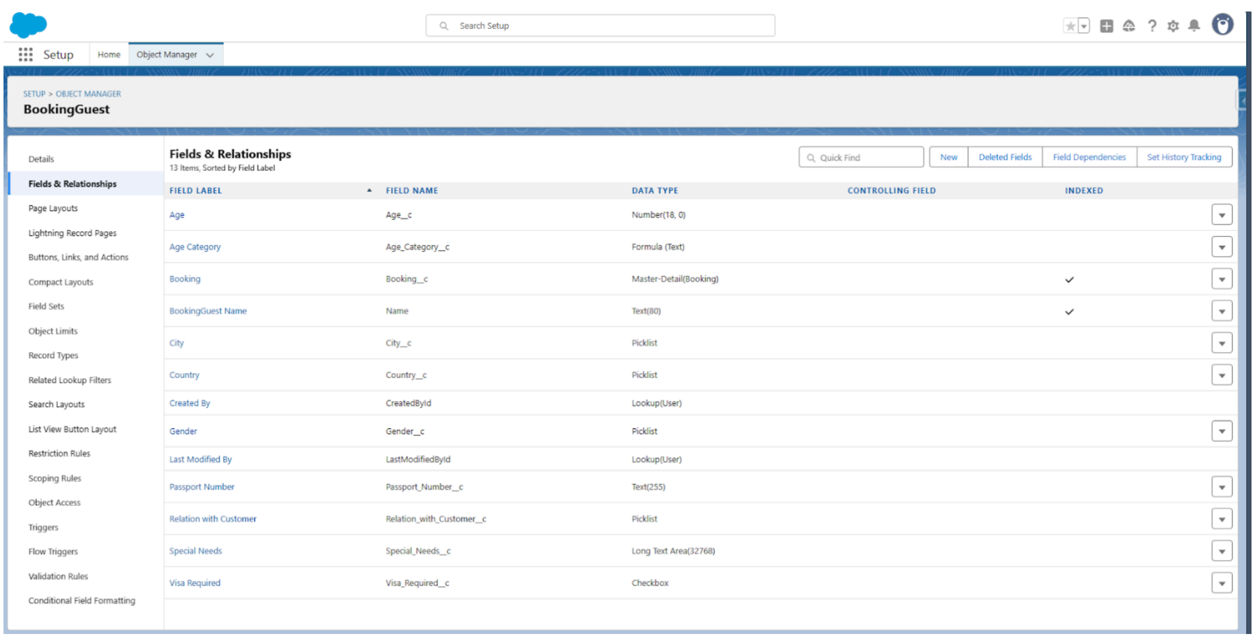


Figure 6. BookingGuest Fields & Relationships

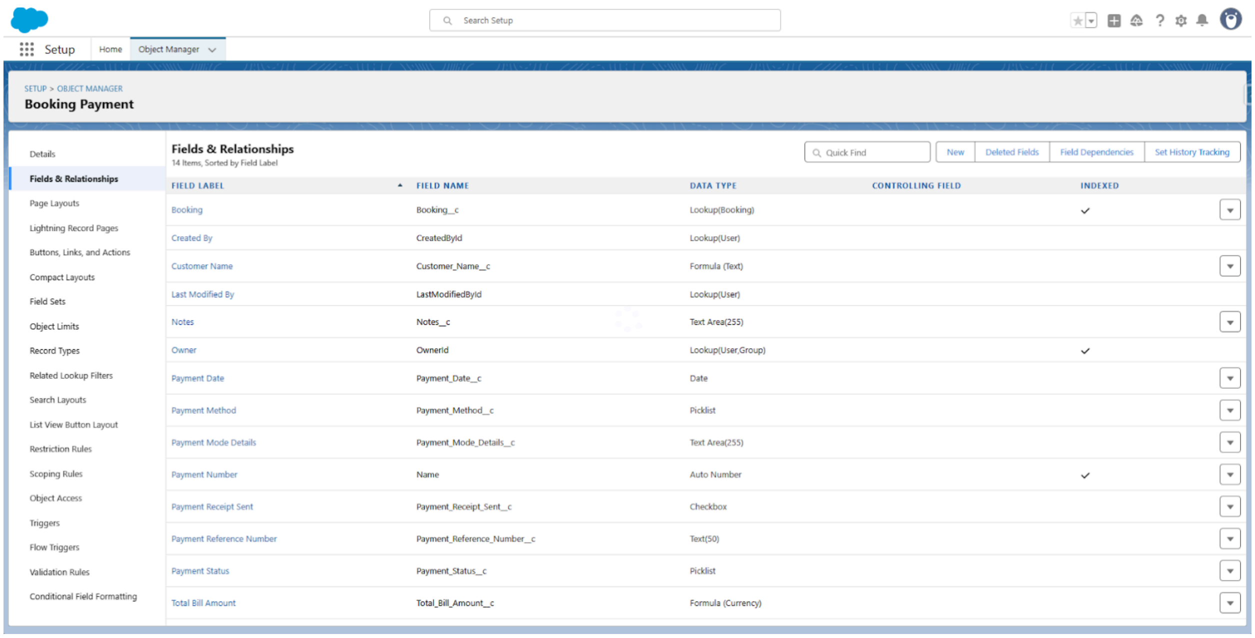


Figure 7. Booking Payment Fields & Relationships

Milestone 5: Field Dependencies

Field dependencies were configured to enhance user experience and ensure logical data selection.

Booking Guest Object:

- Controlling Field: Country
- Dependent Field: City

(e.g., If Country is India, cities like Hyderabad, Delhi become selectable)

Booking Object:

- Controlling Field: Membership

- Dependent Field: Preferred Accommodation

  (e.g., Basic = Hotel, Gold = Resort, VIP = Luxury Hotel)

Employee Object:

- Dependency 1: Department → Role
- Dependency 2: Country → City

These dependencies reduce entry errors and ensure relevance based on other field selections.

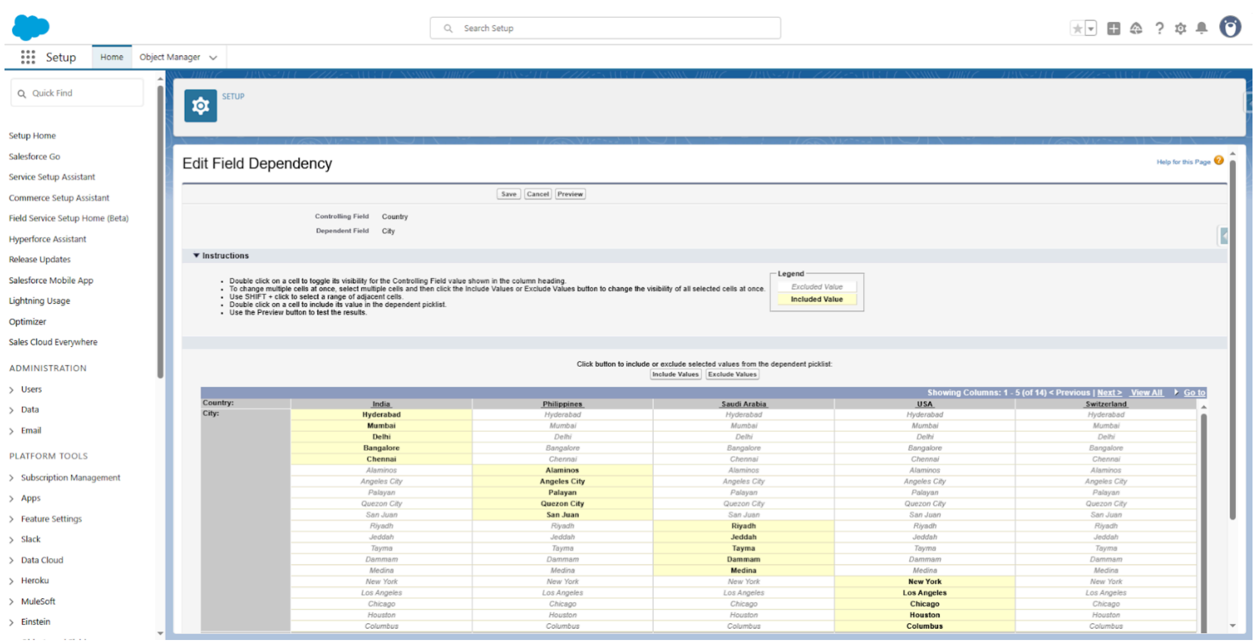


Figure 8. County (City) Field Dependencies

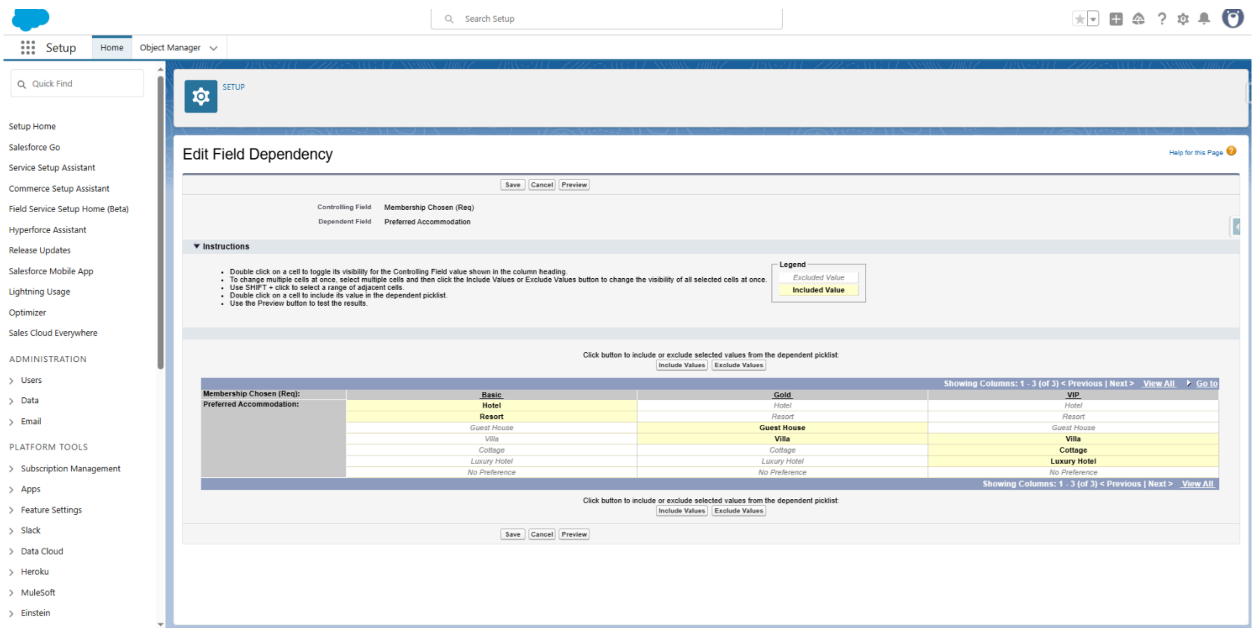


Figure 9. Membership (Preferred Accommodation) Field Dependencies

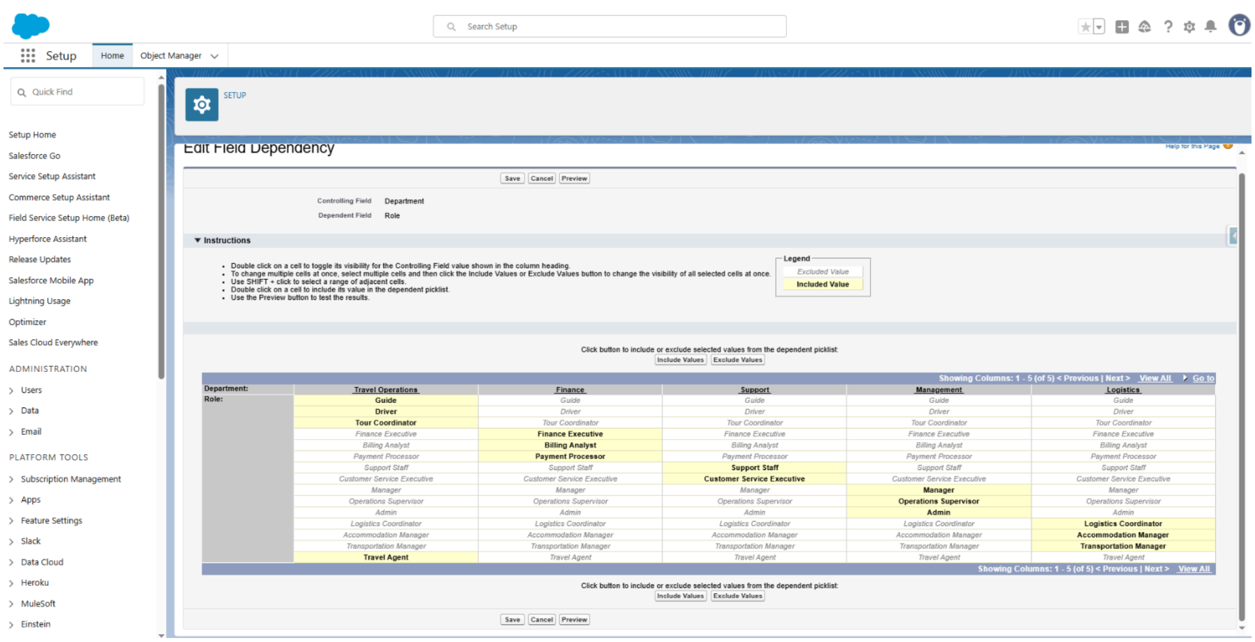


Figure 10. Department (Role) Field Dependencies

Milestone 6: Validation Rules

Validation rules were created to enforce clean, accurate data entry.

- Customer Info:
  - Phone must be 10 digits

    LEN(Phone__c) <> 10

  - Email must be valid

    NOT(REGEX(Email__c,

    "^[a-zA-Z0-9._]+@[a-zA-Z0-9.]+\\.[a-zA-Z]{2,}$"))

  - Date of Birth must not be in the future

    Date_Of_Birth__c > TODAY()

- Booking Guest:
  - Age must be greater than 0

    Age__c <= 0

- Employee:
  - Email is mandatory for Finance Executive and Admin

    AND(OR(ISPICKVAL(Role__c, "Finance Executive"), ISPICKVAL(Role__c, "Admin")), ISBLANK(Email__c))

- Guide must select a language

  AND(ISPICKVAL(Role__c, "Guide"), ISBLANK(Languages_Spoken__c))

- Booking:
  - New bookings must have Booking Status set to "Pending"

    AND(ISNEW(), NOT(ISPICKVAL(Status__c, "Pending")))

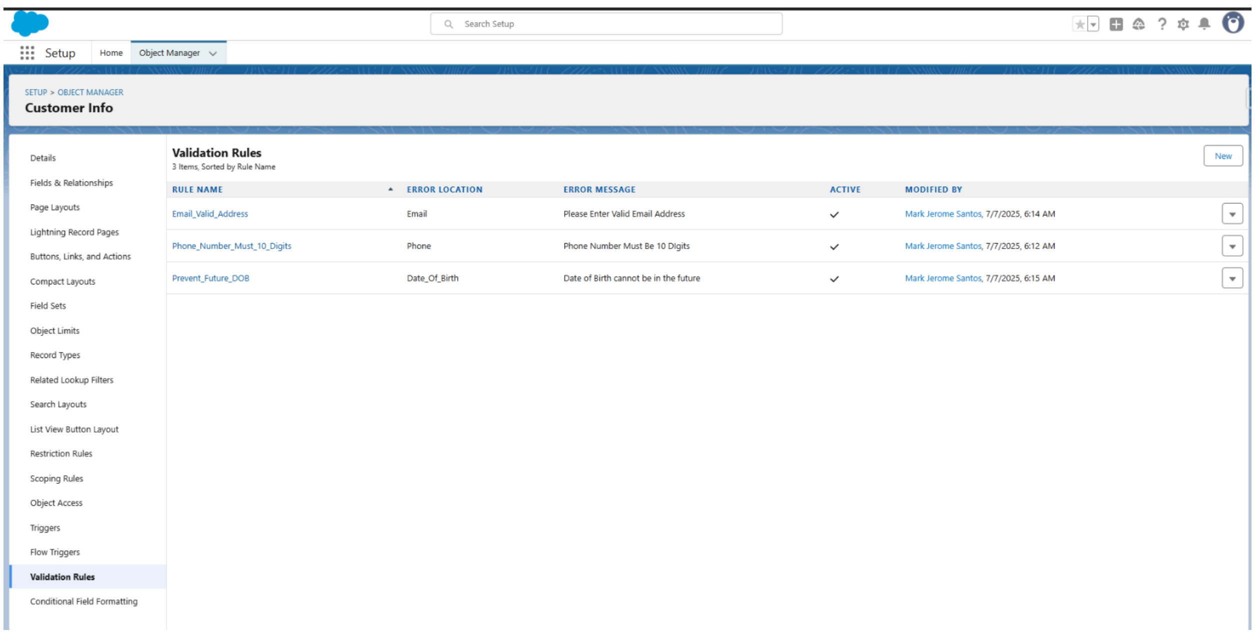


Figure 11. Customer Info Validation Rules

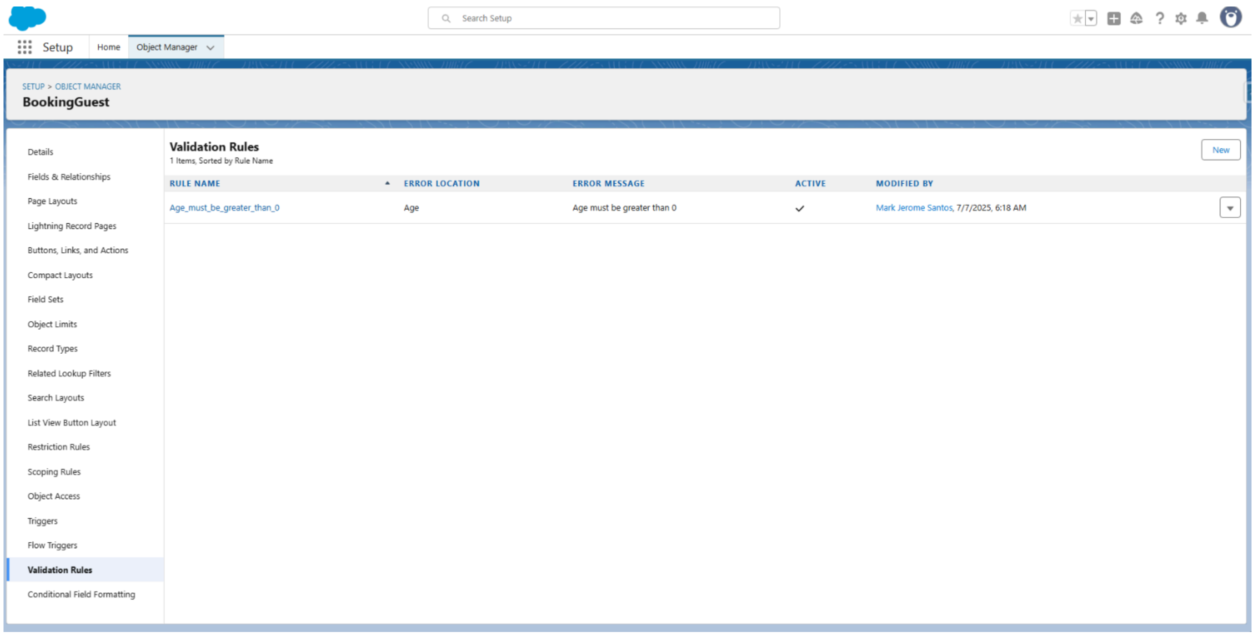


Figure 12. BookingGuest Validation Rules

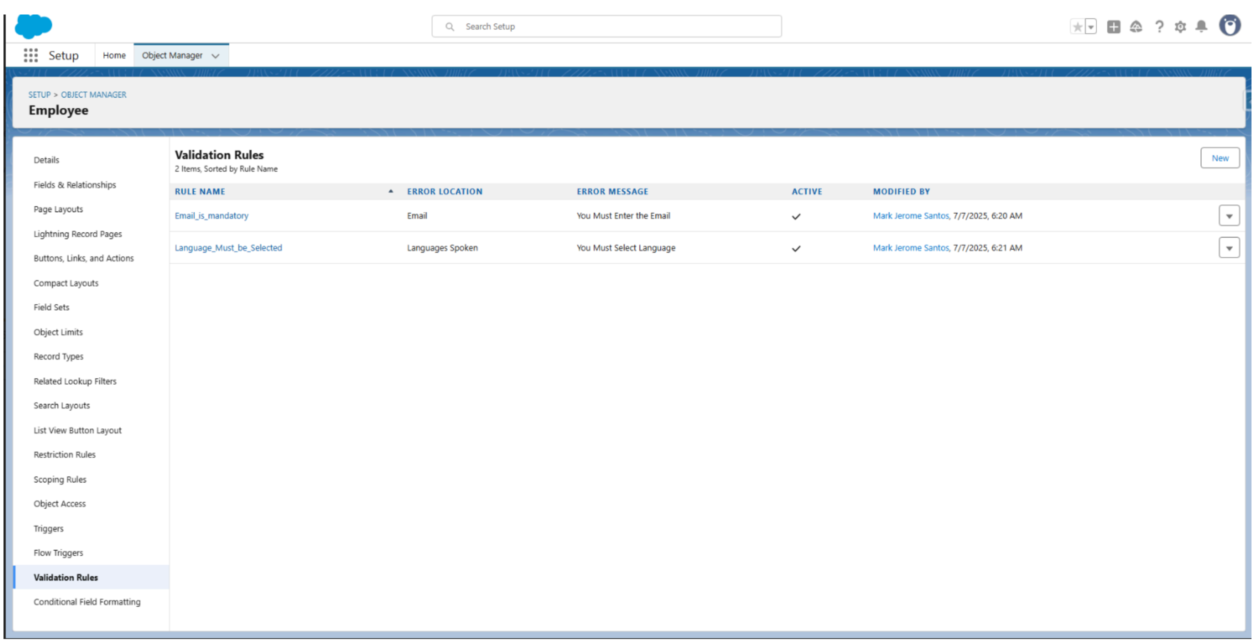


Figure 13. Employee Validation Rules

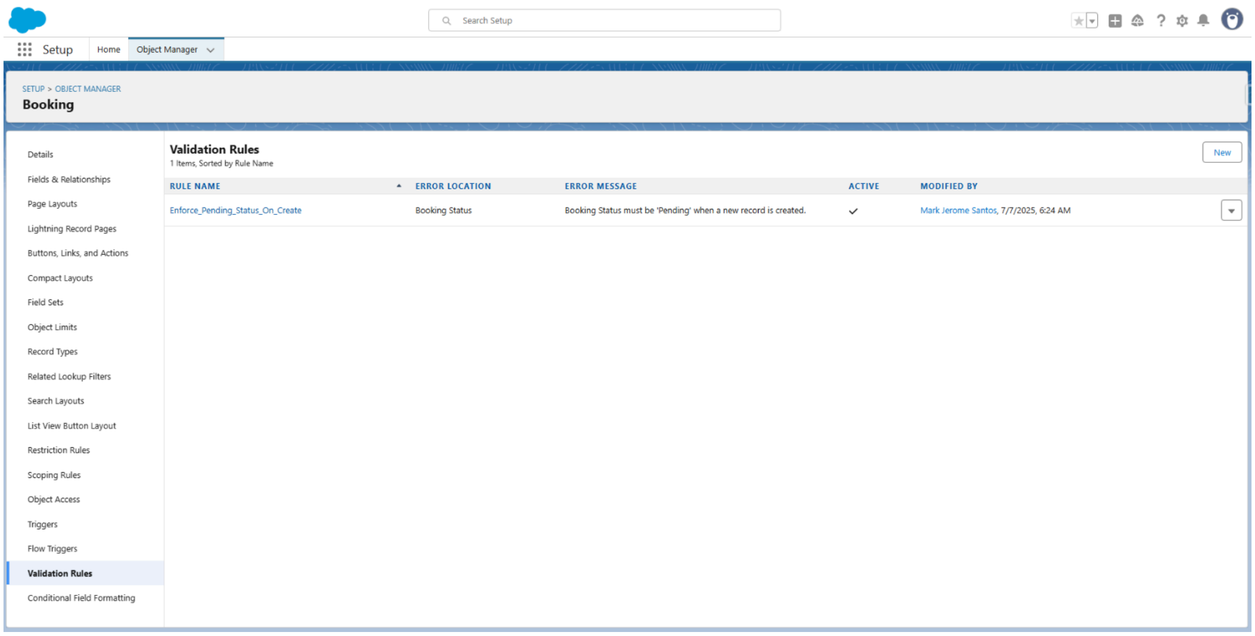


Figure 14. Booking Validation Rules

Milestone 7: Approval Process

A Booking Cancellation Approval Process was created with the following:

- **Entry Criteria:**
  - Booking Status = "Cancelled"
  - Cancel Confirmation = True
- **Initial Submitters:** Booking Owner or Travel Agent
- **Approval Steps:**
  - Assigned to a specific user (Manager or Michael Jackson)
- **Actions:**

- On Submit: Set Booking Status to "Pending"
- **On Approval:**
  - Set Approval Status to "Approved"
  - Email customer using the "Cancellation Approved" template
- **On Rejection:**
  - Set Booking Status to "Confirmed"
  - Set Approval Status to "Rejected"
  - Email customer using the "Cancellation Rejected" template

- **Templates Used:**
  - Booking Cancellation Approval Notification
  - Cancellation Approved
  - Cancellation Rejected

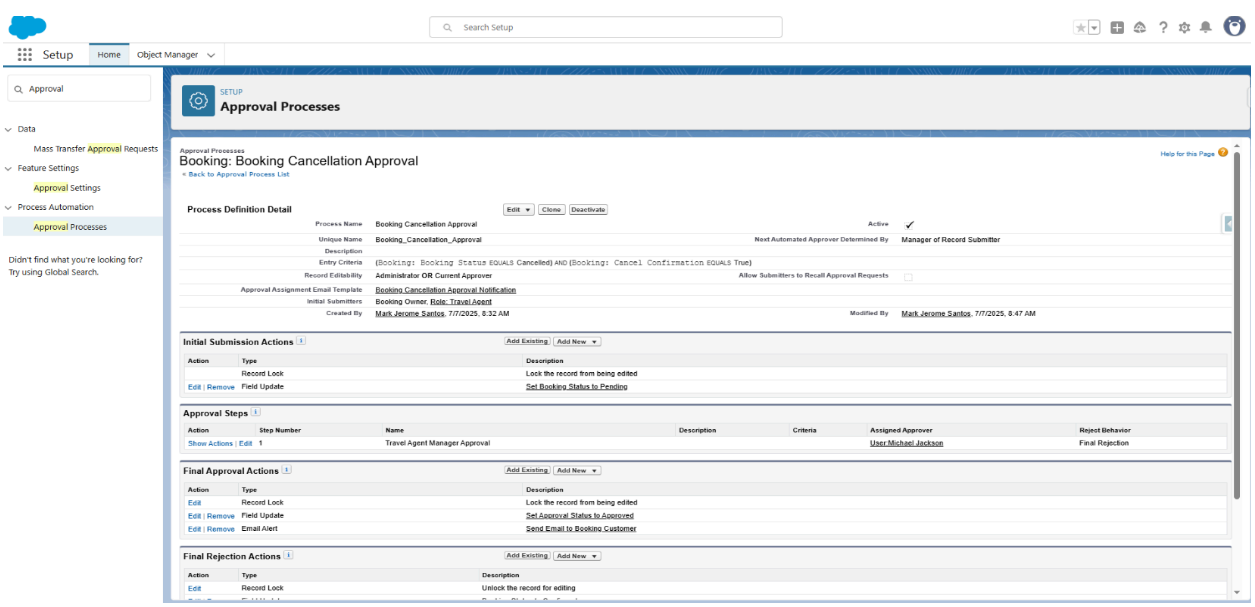


Figure 15. Booking Approval Process

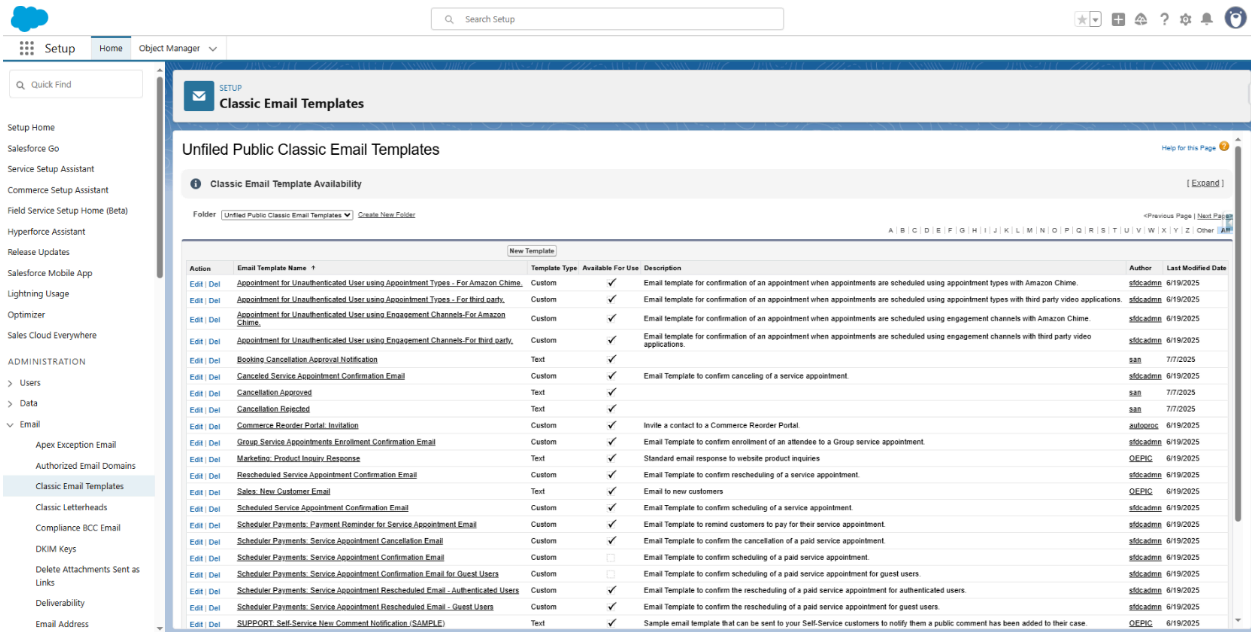


Figure 16. Email Templates

Milestone 8: Flows

A Record-Triggered Flow was built to validate the number of BookingGuest records against the number of travelers in a Booking.

- If the user attempts to create more guests than allowed, the flow prevents the save and displays an error message.
- This maintains data consistency between Booking__c.Number_of_Travelers__c and related BookingGuest__c records.

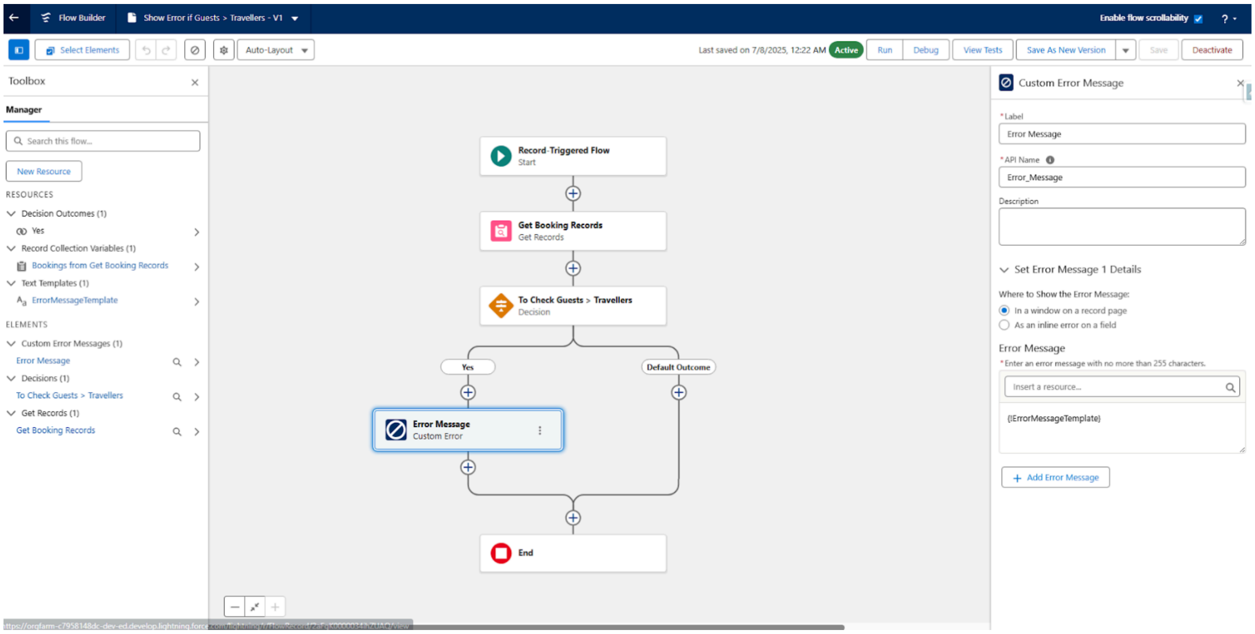


Figure 17. Flow

Milestone 9: Workflows

A Workflow Rule was created for follow-up tasks after trip completion:

- Trigger: Booking Status = "Completed"
- Action: Create Task for the assigned Travel Agent
    - Due Date: 3 days after Travelling_End_Date__c
    - Subject: “Follow up for feedback”
    - Assigned To: Related employee/agent

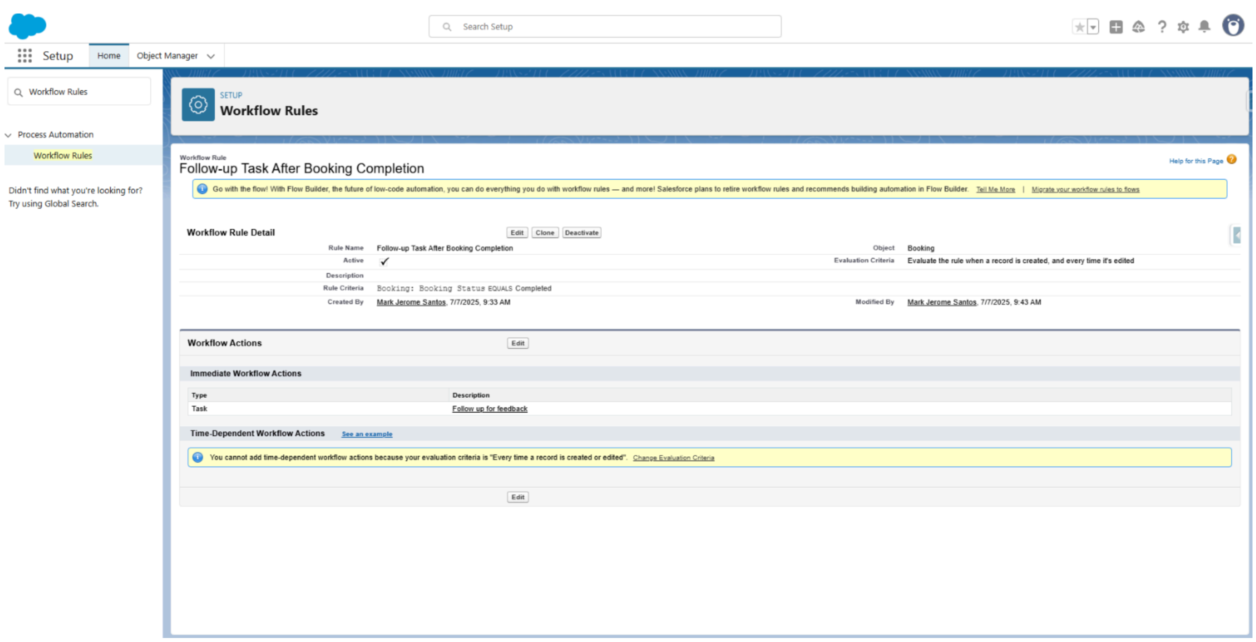


Figure 18. Workflow Rules

Milestone 10: Process Builder

A Process Builder was configured to update Booking status:

- Trigger: When a Booking Payment’s status changes to "Completed"
- Action: Automatically update the related Booking record’s Booking_Status__c to “Confirmed”
- Note: This automation reduces manual coordination between teams

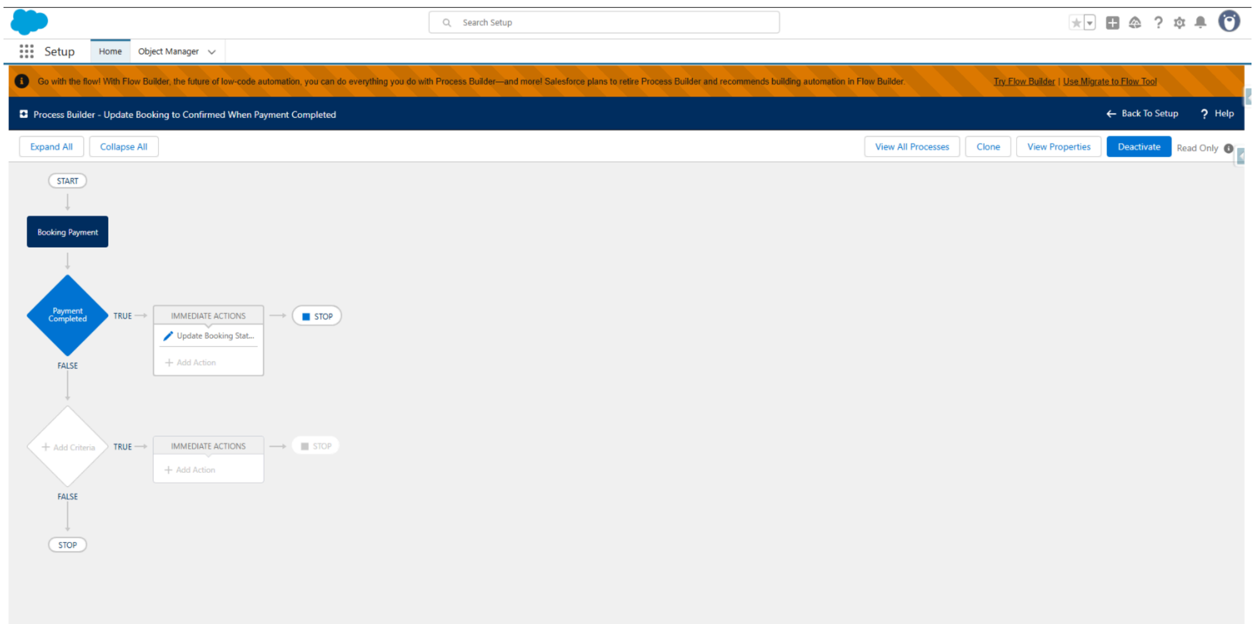


Figure 19. Process Builder

Milestone 11: Triggers (Apex)

Trigger 1: Auto-create Booking Payment

When a Booking is created, a Booking Payment record is automatically inserted with:

payment.Booking__c = booking.Id;

payment.Payment_Status__c = 'Pending';

Trigger 2: Auto-create Booking Guests

The number of BookingGuest records is based on Number_of_Travelers__c. Each guest is inserted with a sequential label:

guest.Name = 'Guest ' + i;

Both functionalities are handled in the same trigger (BookingTrigger) and modularized using a helper class (BookingTriggerHandler).

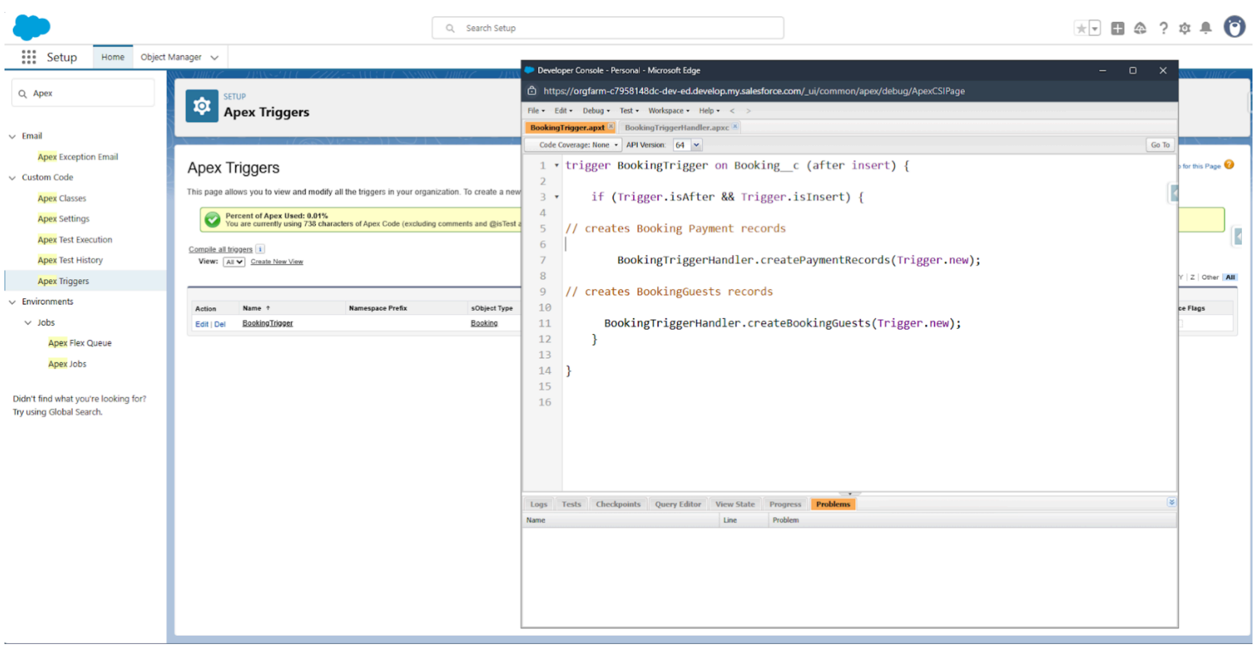


Figure 20. Trigger

Milestone 12: Asynchronous Apex

Asynchronous Apex in Salesforce allows for running operations in the background without blocking the main user interaction. It is essential for processes that are resource-intensive, long-running, or require scheduled execution. This milestone focuses on executing background processes that should not interrupt the user's workflow. In Salesforce, Asynchronous Apex allows long-running tasks like sending emails, complex calculations, and batch operations to be performed in the background. This helps prevent hitting governor limits and ensures a smooth user experience.

Three major types of asynchronous executions were used in this milestone:

- @future methods
- Queueable + Schedulable
- Batchable + Schedulable

Future Apex – Booking Confirmation Email

When a booking is confirmed (status changes to "Confirmed"), the system should automatically send a confirmation email to the customer — asynchronously, using a @future Apex method.

Apex Class: BookingConfirmationEmailer

- Annotation: @future(callout=false) is used to run this method asynchronously.
- Functionality: Loops through a set of Booking records and sends personalized confirmation emails to customers.
- Conditions:

- Booking must have a valid email (Customer_Email__c)
- Booking status must have changed to "Confirmed"

Key Logic:

if (booking.Booking_Status__c == 'Confirmed' && oldBooking.Booking_Status__c != 'Confirmed') {

bookingIdsToSend.add(booking.Id);

}

Trigger Integration:

This logic was added to the existing BookingTrigger under the after update context. It detects status changes and calls the BookingConfirmationEmailer.sendBookingConfirmation() method with the IDs of affected bookings.

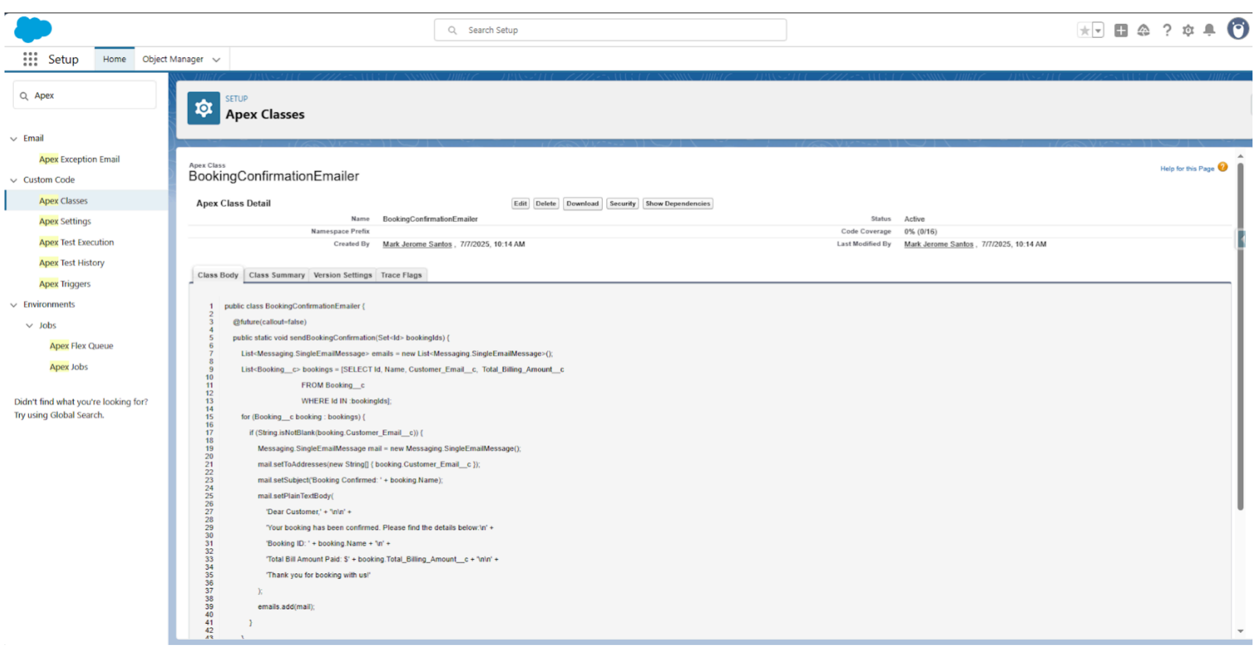


Figure 21. BookingConfirmationEmailer

Queueable + Schedulable Apex – Reminder 3 Days Before Tour

Send automated email reminders to customers 3 days before their tour start date, ensuring they are prepared. This task is handled via Queueable Apex, which is scheduled daily using a Schedulable Apex class.

Queueable Class: BookingReminderQueueable

- Inputs: A list of Booking__c records starting in 3 days.
- Logic: Sends a reminder email with tour start date.

mail.setPlainTextBody(

'Hello,\n\nThis is a friendly reminder that your tour is starting on ' +

booking.Travelling_Start_Date__c.format()

);

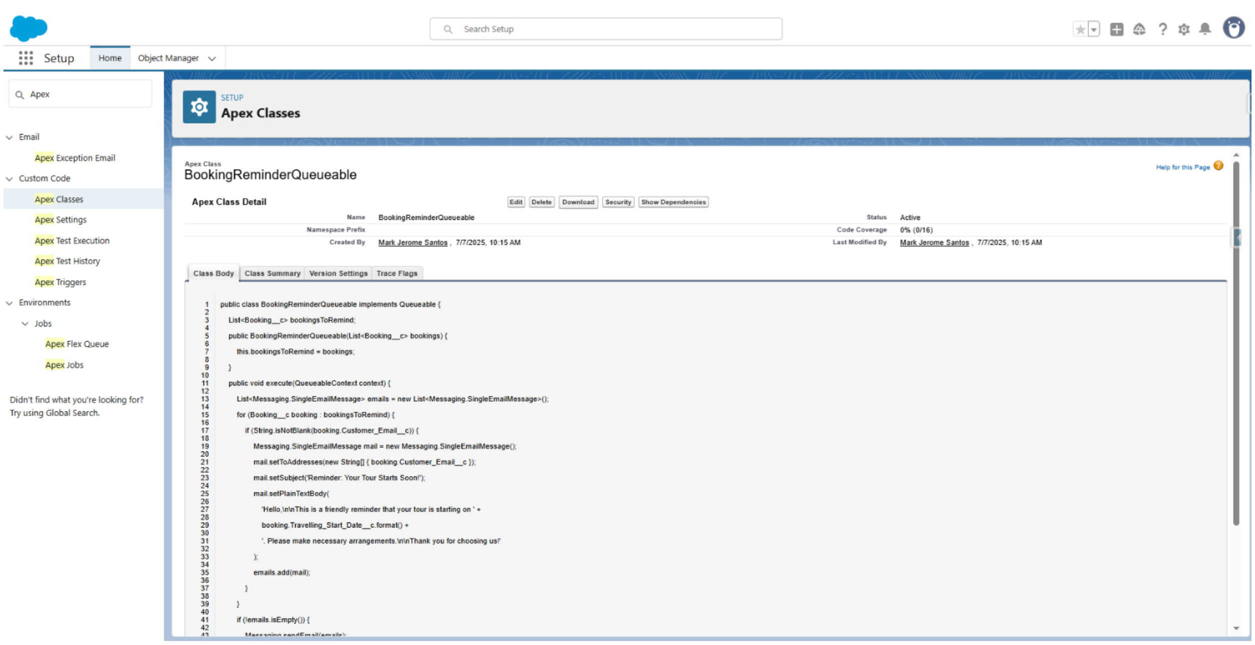


Figure 22. BookingReminderQueueable

Schedulable Class: BookingReminderScheduler

- Executes daily at 6:00 AM (defined via cron expression).
- Queries for bookings with a Travelling_Start_Date__c = 3 days from today and status "Confirmed".
- Passes those bookings to BookingReminderQueueable for email delivery.

Scheduling Logic:

*String cronExp = '0 0 6 * * ?';*

System.schedule('Daily Booking Reminder Job', cronExp, new BookingReminderScheduler());

Batchable + Schedulable Apex – Payment Reminder Post-Booking

Automatically send payment reminders to customers whose bookings were made yesterday and are still in "Pending" status.

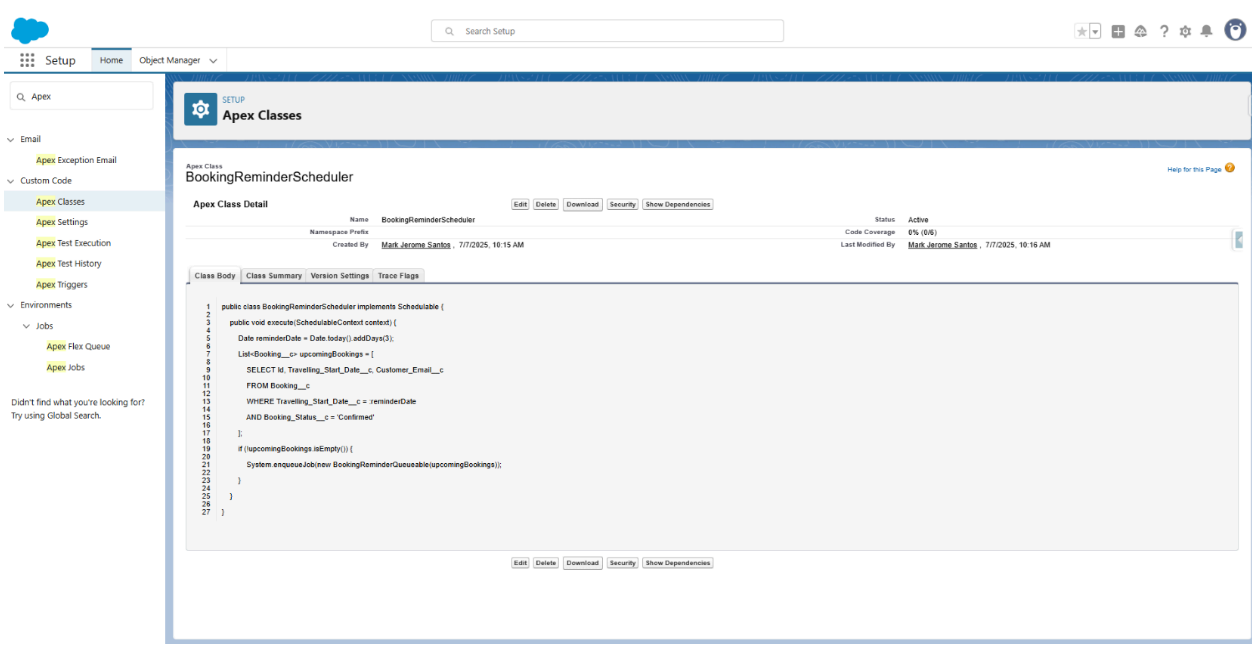


Figure 23. BookingReminderScheduler

Batch Class: PaymentReminderBatch

- start() method: Queries bookings created yesterday with Booking_Status__c = 'Pending'
- execute() method: Sends a payment reminder email to each customer
- finish() method: Sends a summary email to the admin at annapurna@thesmartbridge.com

Key Email Body:

mail.setPlainTextBody('Hi,\n\nThis is a gentle reminder to complete your payment for booking: ' + booking.Name);

Schedulable Class: SchedulePaymentReminderBatch

- Executes the PaymentReminderBatch with a batch size of 200.
- Scheduled to run daily at 5:00 AM.

Scheduling UI Steps:

- Go to Setup → Apex Classes → Schedule Apex
- Job Name: SchedulePaymentReminderBatchJob
- Frequency: Weekly (All days checked)
- Start Time: 5:00 AM

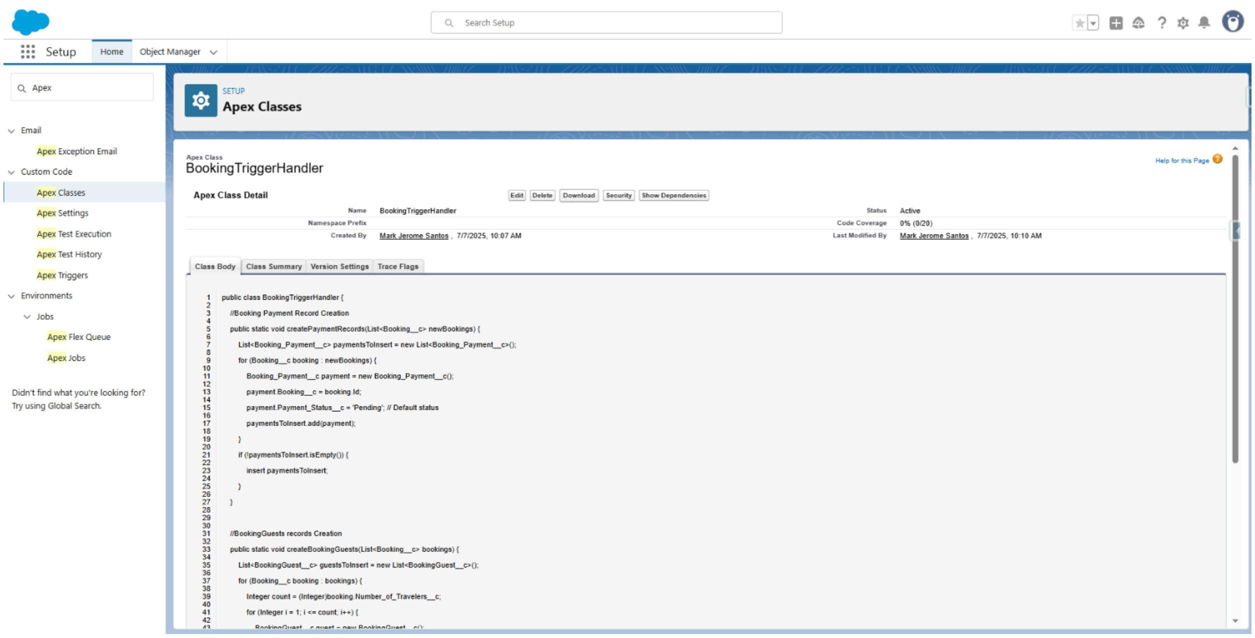


Figure 24. PaymentReminderBatch

This milestone strengthens the CRM's automation capabilities, making customer communication timely and hands-free. The implementation is designed to scale, reduce manual intervention, and maintain compliance with Salesforce governor limits.

V. Phase 3: UI/UX Development & Customization

This phase focuses on user interface setup, layout optimization, dynamic form behavior, role-based user configuration, report creation, dashboard visualizations, and Lightning Web Component (LWC) development. These milestones were implemented to ensure the Salesforce Tours & Travels CRM delivers a user-friendly, dynamic, and data-rich experience for all roles.

Milestone 13: The Lightning App

A custom Lightning App named Tours & Travels CRM was created using the App Manager.

Steps Followed:

- Navigated to App Manager → Clicked New Lightning App.
- Entered App Name: Tours & Travels CRM.
- Uploaded a relevant travel-themed logo.
- Left default settings under App Options and Utility Items.
- Moved the following objects to Selected Items:
    - Customer Info, TravelPackages, Booking, Booking Payments, BookingGuests, Employees, Feedback, Task, Reports, Dashboards.
- Selected System Administrator under Available Profiles.

The app was successfully created and made available to the system admin for full access and visibility.

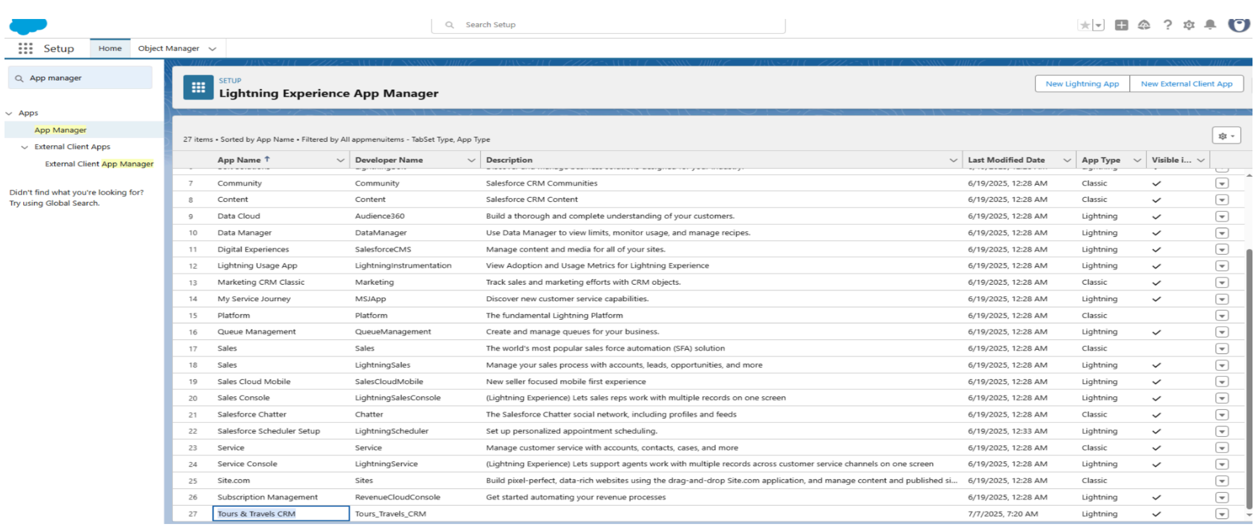


Figure 25. Tours & Travels CRM

Milestone 14: Editing of Page Layouts

Page layouts were modified across all key objects to ensure field sections were logically grouped and user-friendly.

Objects with Edited Layouts:

- Customer Info
- BookingGuest
- TravelPackage
- Employee
- Booking
- Booking Payment
- Feedback

Changes Made:

- Fields were reorganized using drag-and-drop to align similar data (e.g., contact info, booking metadata, travel info).
- Ensured important fields such as Booking Status, Customer, Travel Start Date, etc., are easily accessible on top.
- Saved and updated each layout to reflect across the Lightning interface.

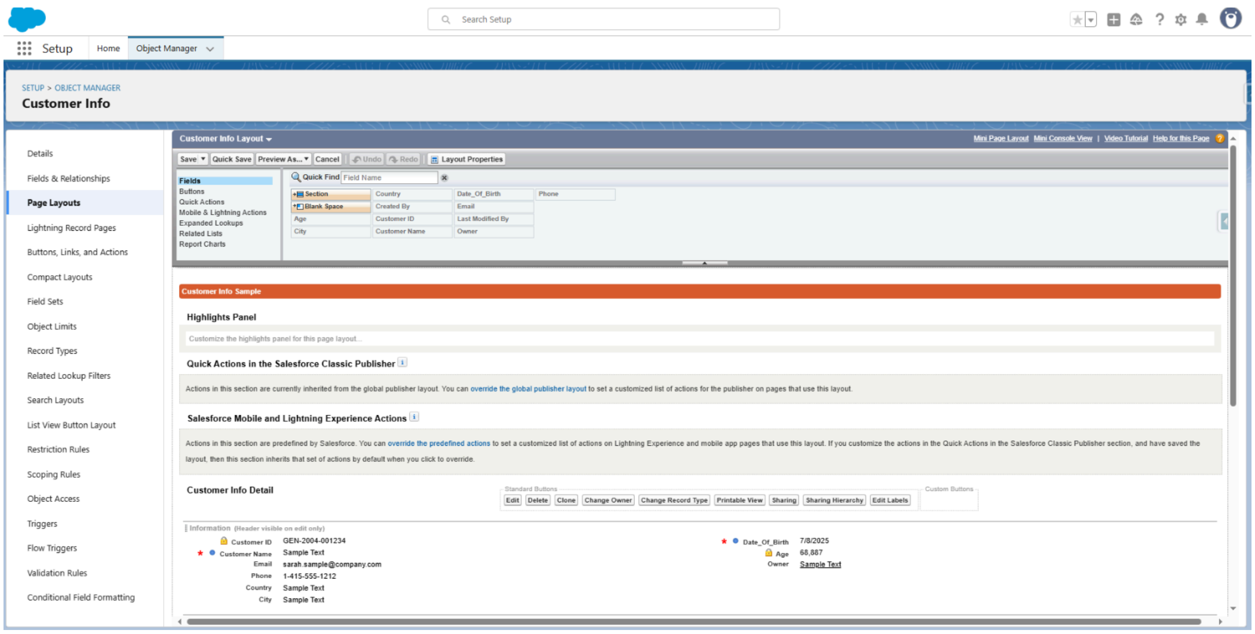


Figure 26. Customer Info Page Layout

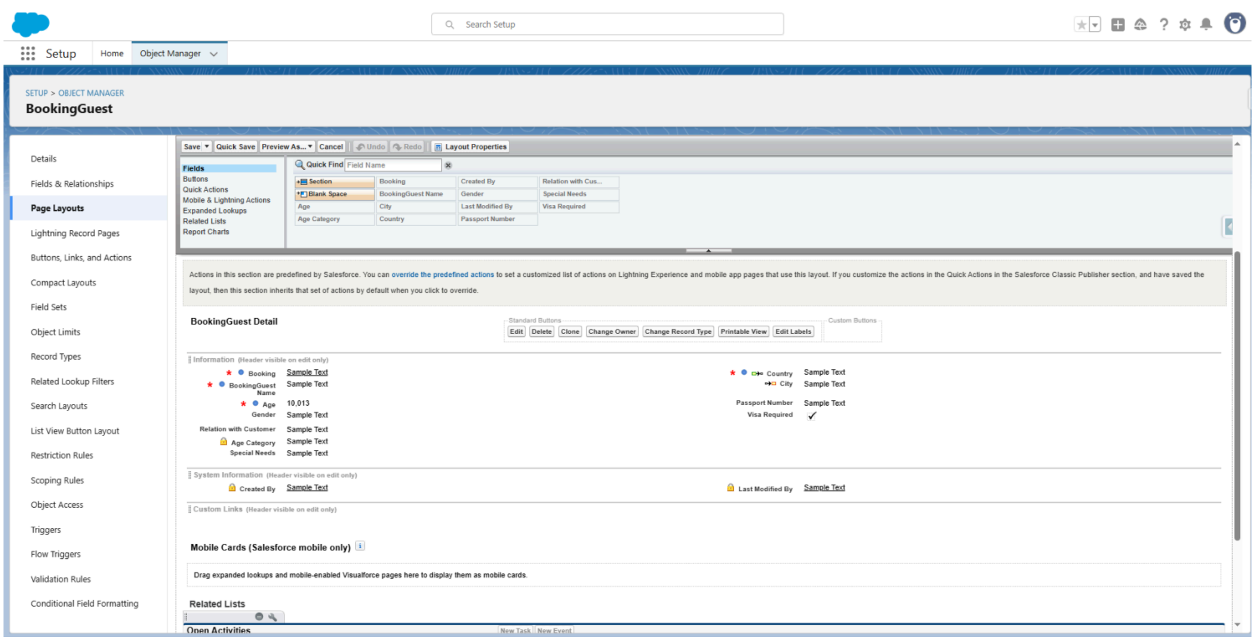


Figure 27. BookingGuest Page Layout

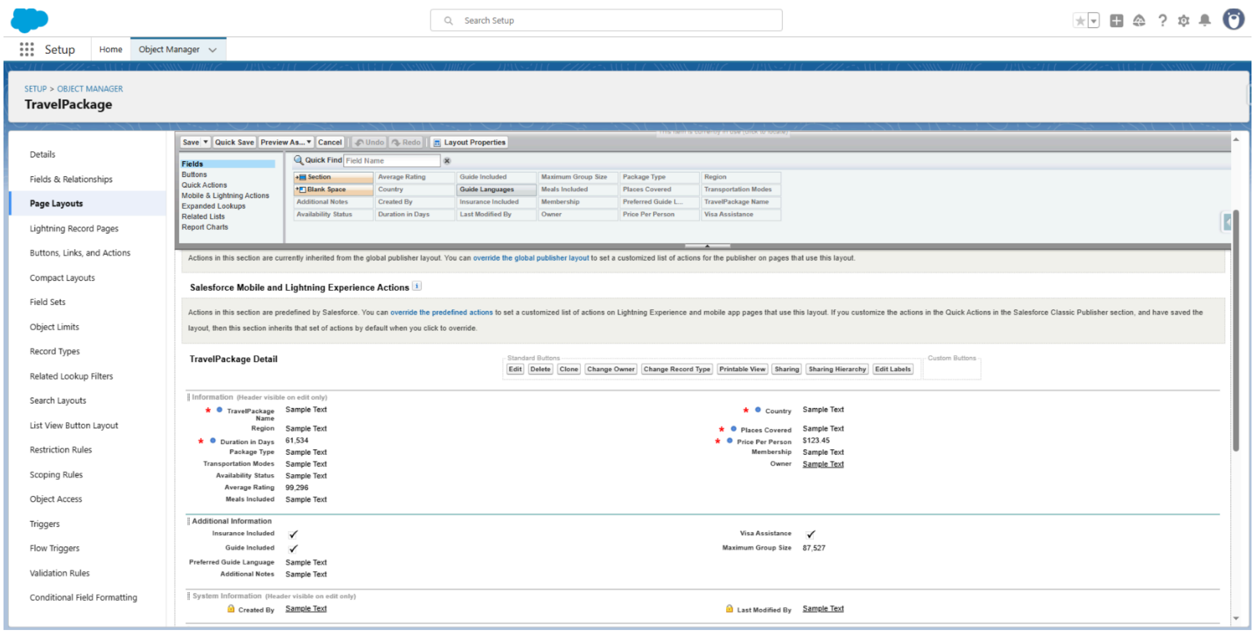


Figure 28. TravelPackage Page Layout

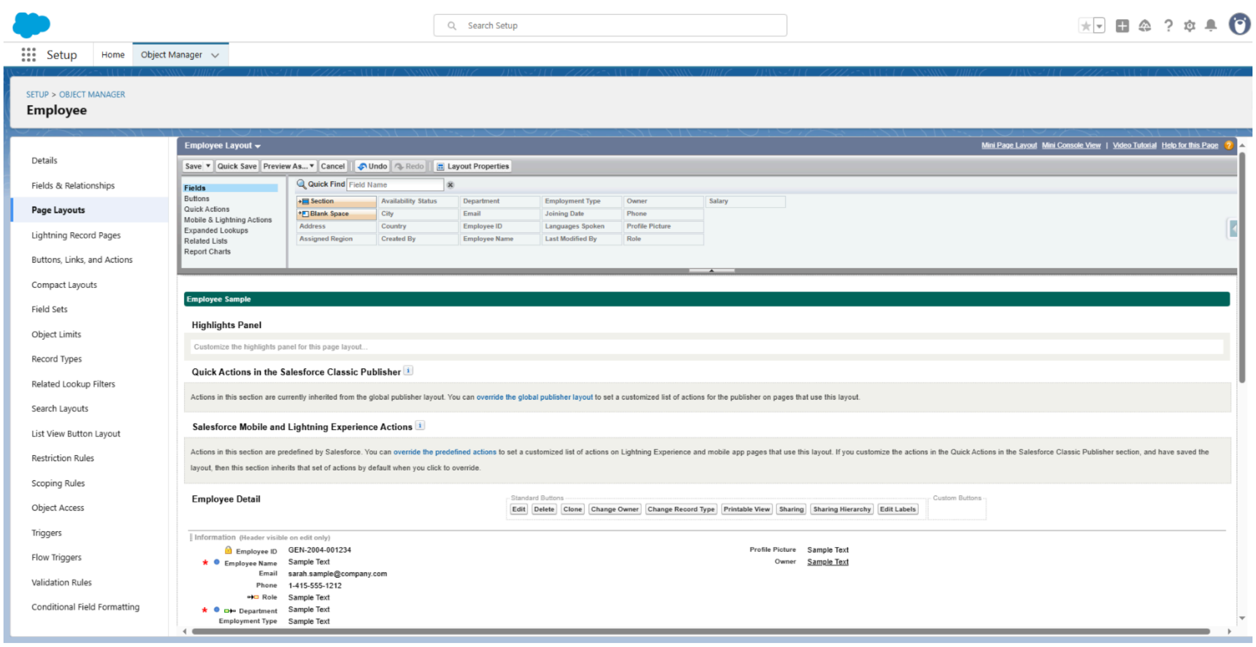


Figure 29. Employee Page Layout

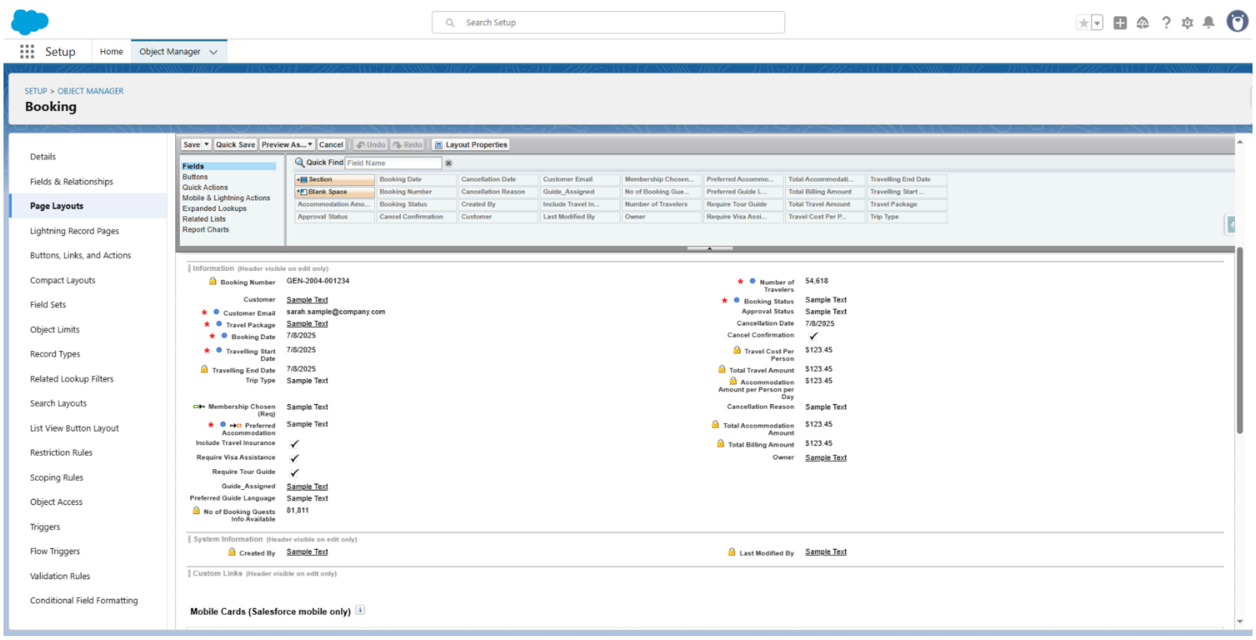


Figure 30. Booking Page Layout

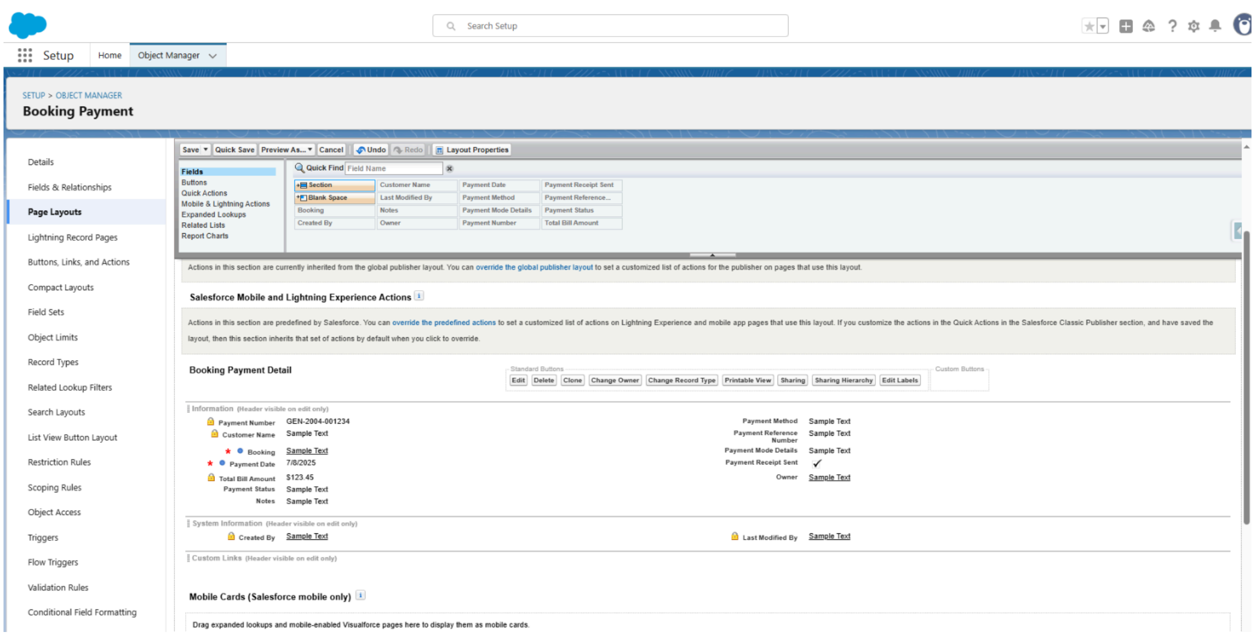


Figure 31. Booking Payment Page Layout

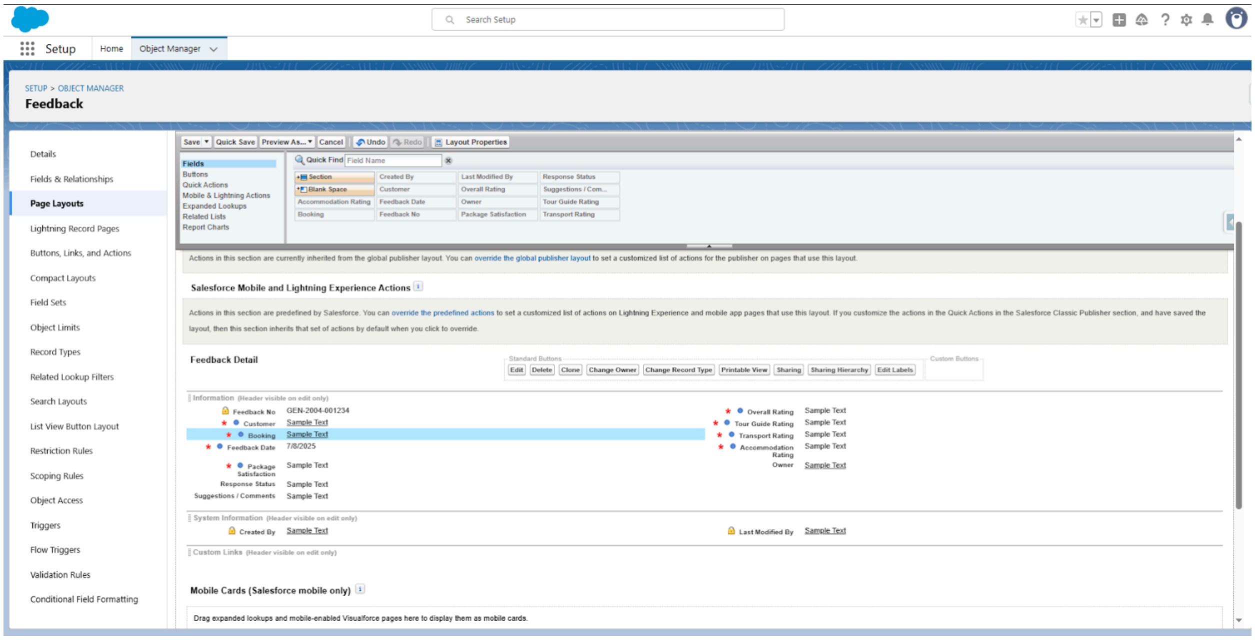


Figure 32. Feedback Page Layout

Milestone 15: Dynamic Forms

Dynamic Forms were enabled for the Booking object to improve page responsiveness and reduce clutter.

Steps Completed:

- Opened a Booking record → Clicked Edit Page (from Gear icon).
- Clicked on Upgrade Now to enable Dynamic Forms.
- Selected Booking PageLayout and finished setup.
- Applied conditional field visibility:
  - Cancellation Date, Cancel Confirmation, Approval Status: Displayed only when Booking Status = Cancelled.
- Saved and activated the page.

- Selected Org Default for Desktop and Phone, ensuring consistent display across devices.

This resulted in a cleaner UI, where fields are shown based on status, improving usability for agents and admins.

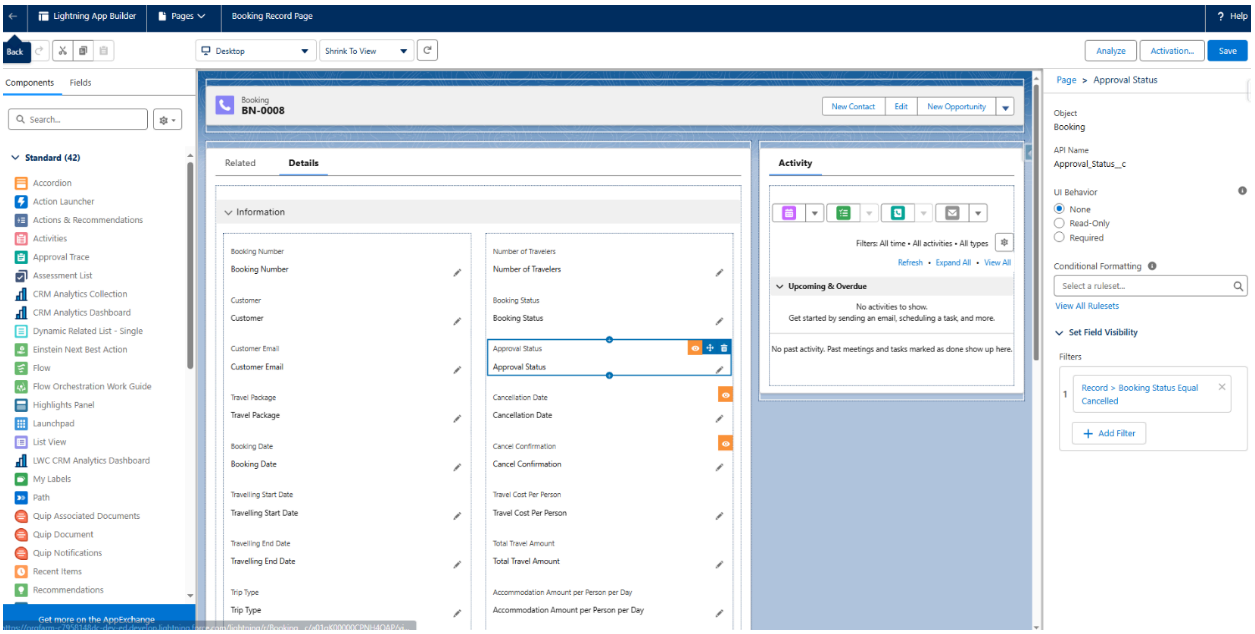


Figure 33. Dynamic Form

Milestone 16: Users

User records were created for the organization following role-based access patterns.

Steps Followed:

- Setup → Users → New User
- Created user:
  - First Name: Michael

- Last Name: Jackson
- Role: Travel Agent Manager
- Profile: Travel Agent Profile
- License: Salesforce Platform

- Repeated for multiple users with:
  - Role: Travel Agent
  - Profile: Travel Agent Profile

Proper user records ensured secure, role-specific access to relevant CRM features like booking, follow-ups, and dashboards.

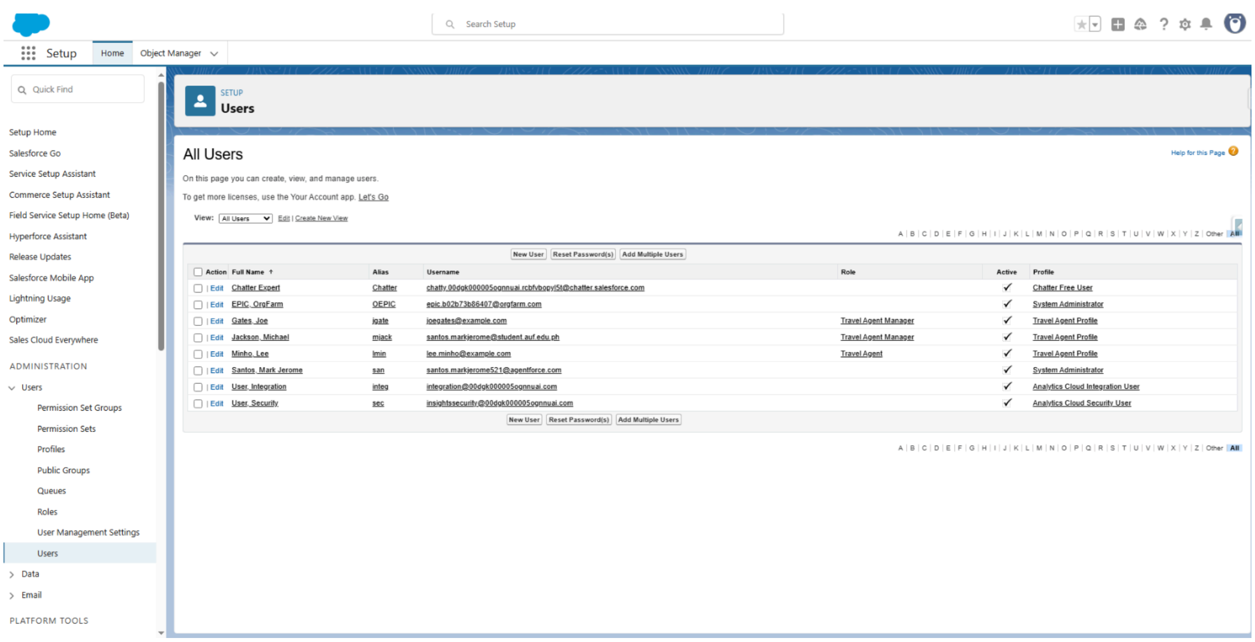


Figure 34. Users

Milestone 17: Reports

Multiple reports were created for tracking revenue, employee roles, and operational metrics.

Key Reports Built:

1. Monthly Revenue Report (Pie Chart):
    - Grouped by: TravelPackage, Revenue Category
    - Bucketed Total Billing Amounts into: Low, Medium, High
    - Filters: Booking Date = This Month, Status = Confirmed/Completed

2. Pending Payments Report
    - Type: Booking Payments with Bookings
    - Filter: Payment_Status__c = 'Pending'

3. Top Travel Packages (By Booking Count)
    - Grouped by: TravelPackage__r.Name
    - Summary: Count of Booking__c

4. Employee Role Distribution
    - Grouped by: Role__c
    - Summary: Count

5. Additional Reports (custom business scenarios):

- Created 5 more to support deeper insights like feedback analysis, customer segments, agent performance.

All reports were saved into appropriate folders (e.g., Revenue Folder) and shared with roles like Travel Agent Manager and Finance Officer with View access.

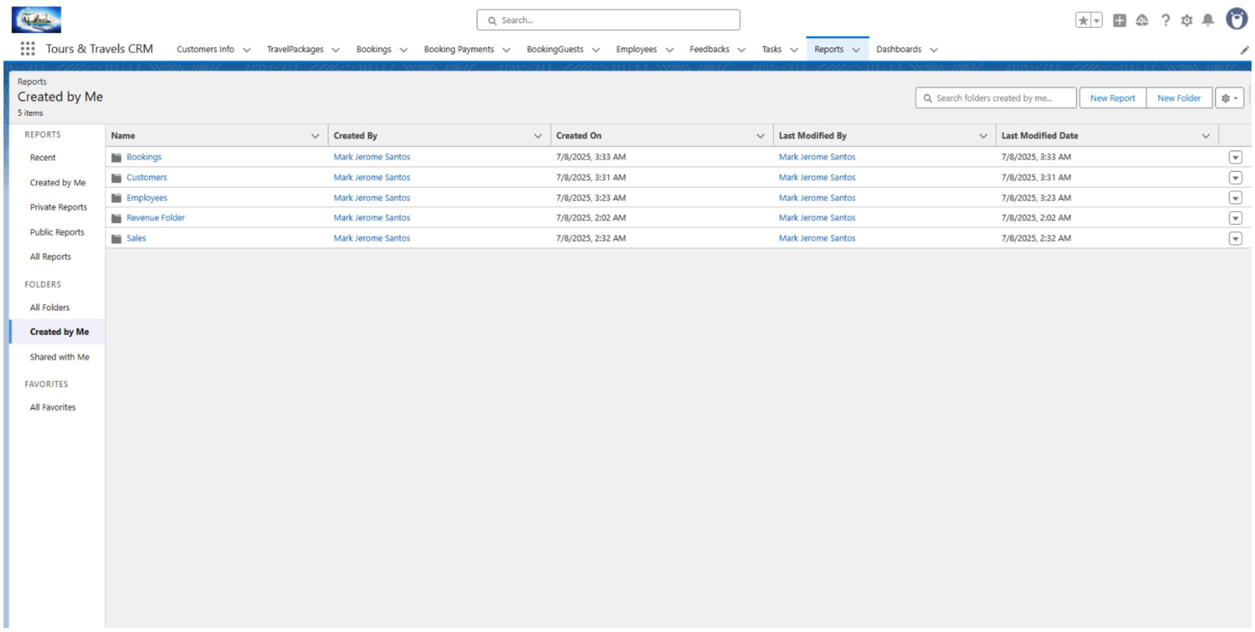


Figure 35. Reports

Milestone 18: Dashboards

Main Dashboard: Tours & Travels Dashboard

Components Added:

1. Monthly Revenue Report – Pie chart with subtitle "Total payments completed Bookings".

2. Additional 3 widgets from other reports:

- Pending Payments
- Top Travel Packages
- Employee Role Breakdown
- Selected appropriate visualizations like bar and table for comparison.

Other Dashboards Created:

- Built 2 more dashboards from remaining reports created in Milestone 17 to support department-specific analysis.

Dashboards were accessible from the Dashboard Tab in the Tours & Travels App, providing real-time metrics.

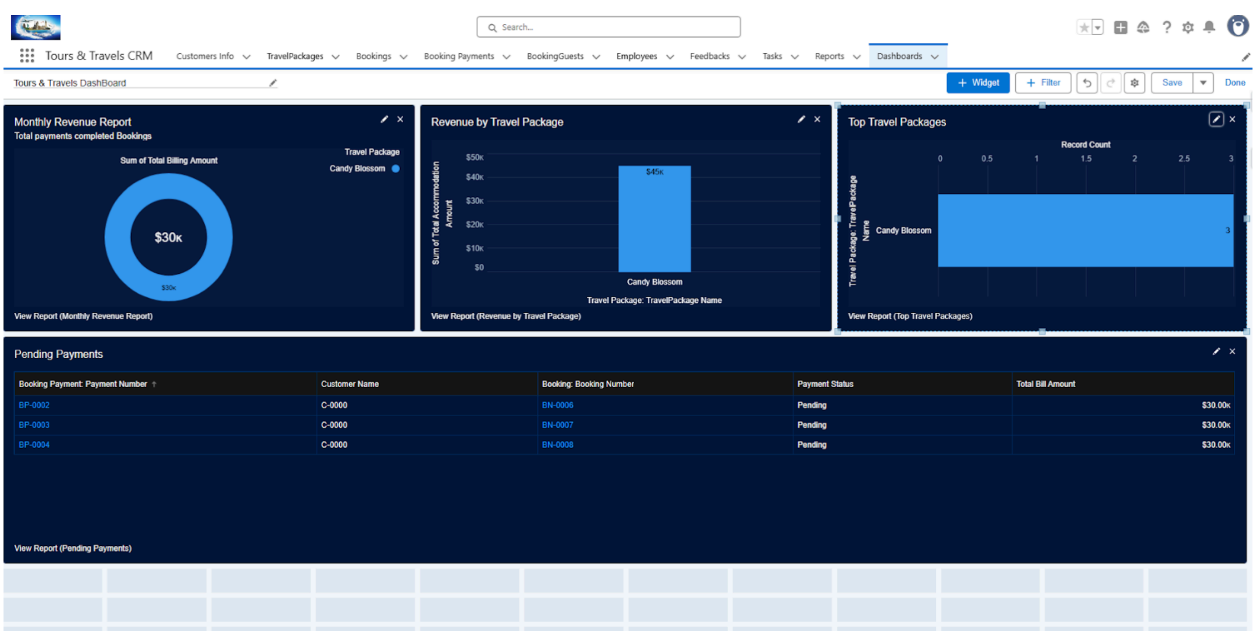


Figure 36. Dashboards

Milestone 19: Lightning Web Component (LWC) Creation

A custom Lightning Web Component was developed to dynamically display available travel packages based on the selected country.

Use Case:

When a customer selects a country during booking, the LWC displays only those Travel Packages that match.

Apex Controller: TravelPackageController

- @AuraEnabled method getPackagesByCountry returns travel packages by country:

```
public static List<TravelPackage__c> getPackagesByCountry(String country) {
        return [SELECT Name, Duration_in_Days__c, ... FROM TravelPackage__c WHERE Country__c = :country];
}
```

LWC Component: travelPackageSelector

Key Features:

- Uses a lightning-combobox to select the country
- Dynamically loads matching packages without page reload
- Displays details: name, duration, guide status, places covered, transport, etc.

Frontend Logic:

```
handleCountryChange(event) {
        this.selectedCountry = event.detail.value;
```

this.packages = [];

this.error = undefined;

getPackagesByCountry({ country: this.selectedCountry })

.then(result => { this.packages = result; })

.catch(error => { this.error = error.body.message; });

}

Targets: Exposed for App Page, Home Page, Record Page

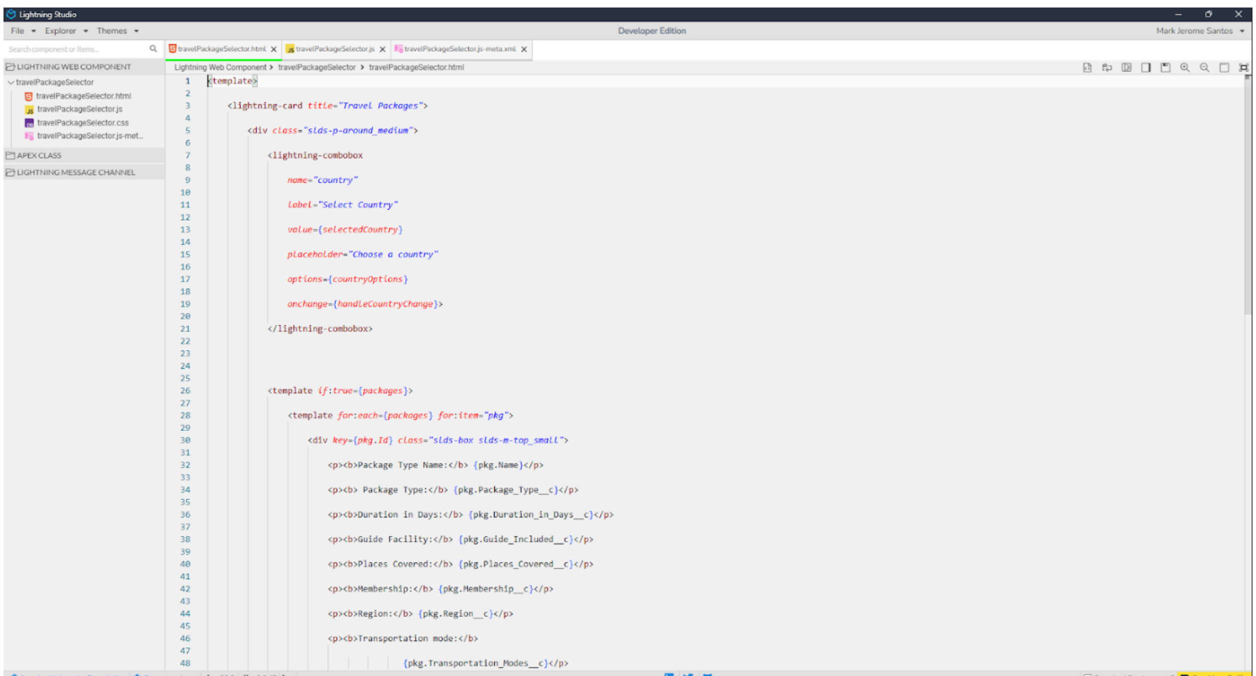


Figure 37. Lightning Studio

Milestone 20: Lightning App Page Creation

The travelPackageSelector LWC was added to a custom App Page for easy user access.

Steps Taken:

- Opened Lightning App Builder → Created App Page: Travel Package Selector
- Selected 1 Region layout
- Dragged the travelPackageSelector component into layout
- Saved and activated it
- Made it visible in the Tours & Travels CRM app

Outcome:

- From App Launcher, both the CRM app and custom LWC app page are accessible.
- Users can now filter and view travel packages interactively by country.

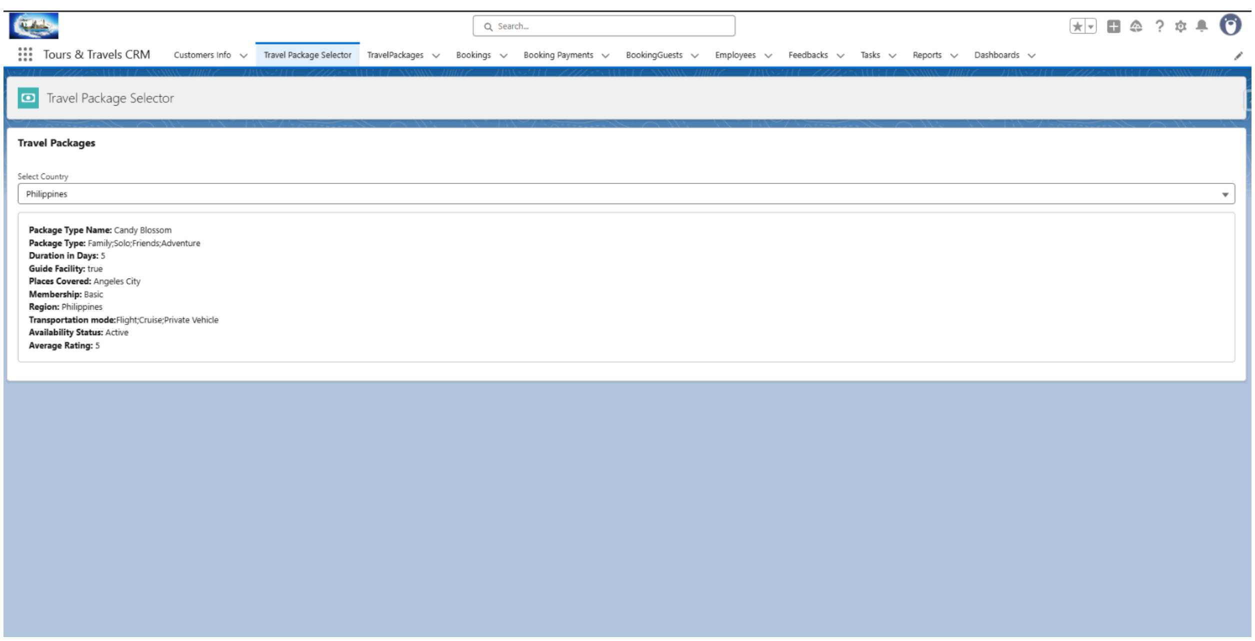


Figure 38. Travel Package Selector

VI. Phase 4: Data Migration, Testing & Security

This phase covers the migration of critical data, the validation of system functionalities through structured testing, and the enforcement of robust security mechanisms using Salesforce tools such as Field History Tracking, Matching Rules, Profiles, Roles, and Permission Sets. Additionally, custom Apex Test Classes were developed to ensure the integrity of automated processes.

Milestone 21: Field History Tracking

Field History Tracking was configured to audit changes to important business fields and display a history of modifications in the record's History related list.

On the Booking Object:

Enabled field history tracking for:

- Number of Travelers – tracks changes to group size.
- Booking Status – tracks status transitions (e.g., Pending → Confirmed).
- Travel Package – tracks any reassignment to a different package.

On the TravelPackage Object:

Enabled tracking for:

- Price Per Person – monitors price changes.

- Availability Status – ensures audit trail for operational visibility (Active, Upcoming, Sold Out).

These configurations ensure data accountability and allow business users and auditors to view historical changes within the record's History section.

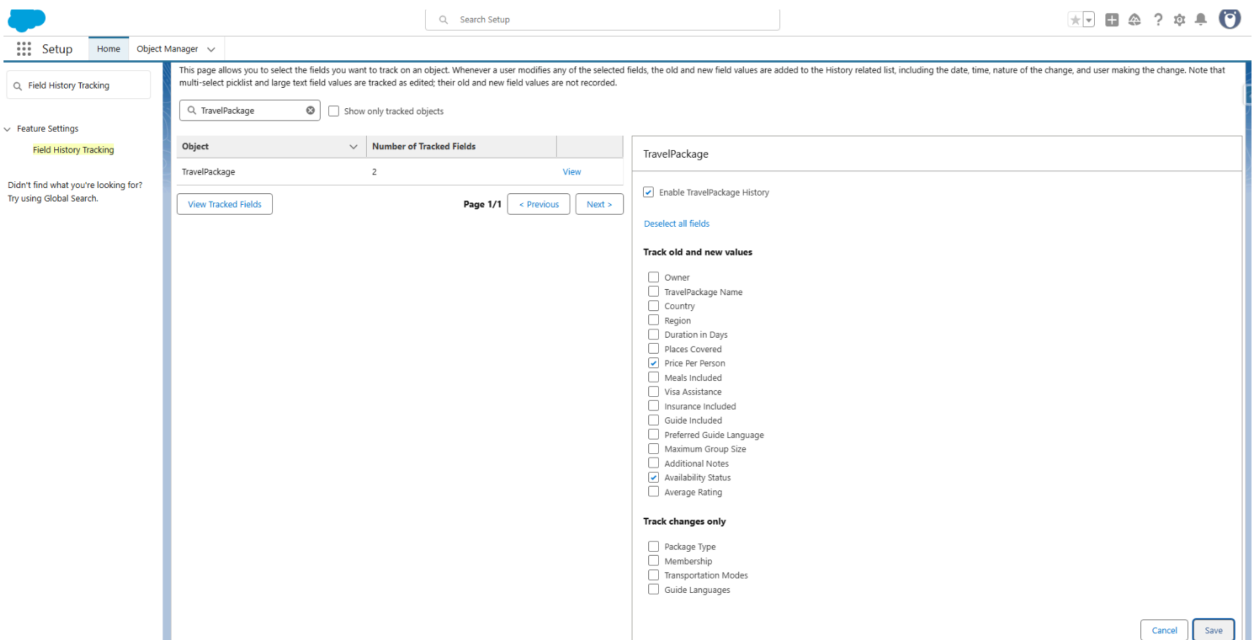


Figure 39. Field History Tracking

Milestone 22: Duplicate & Matching Rules

This milestone ensured customer data uniqueness by preventing duplicate records based on email and phone number combinations.

Matching Rule

- Object: Customer Info
- Rule Name: Unique Email and Phone Number Combination
- Fields Used: Email, Phone
- Matching Method: Exact
- Allow Blank Matches: Checked
- Status: Saved and Activated

Duplicate Rule

- Object: Customer Info
- Rule Name: Unique Email and Phone
- Create/Edit Actions: Allow but report duplicate
- Alert Text: "Email and Phone must be Unique"
- Matching Rule Applied: Unique Email and Phone Number Combination
- Status: Saved and Activated

This combination prevents duplicate customers from being entered, maintaining CRM data hygiene.

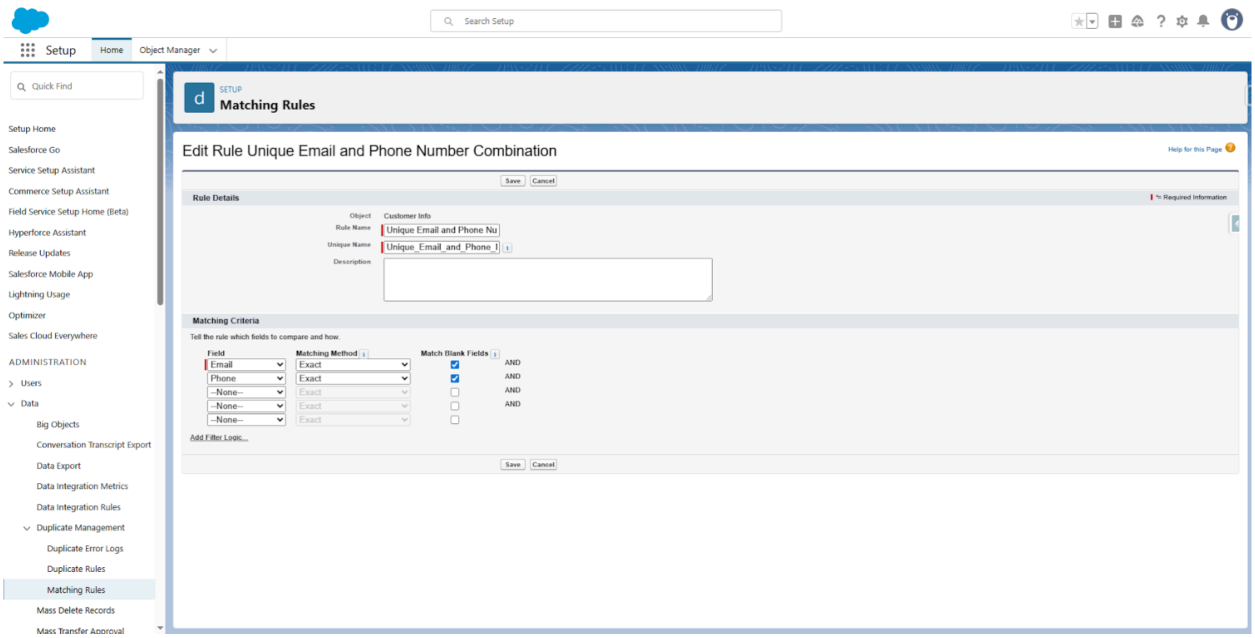


Figure 40. Matching Rules

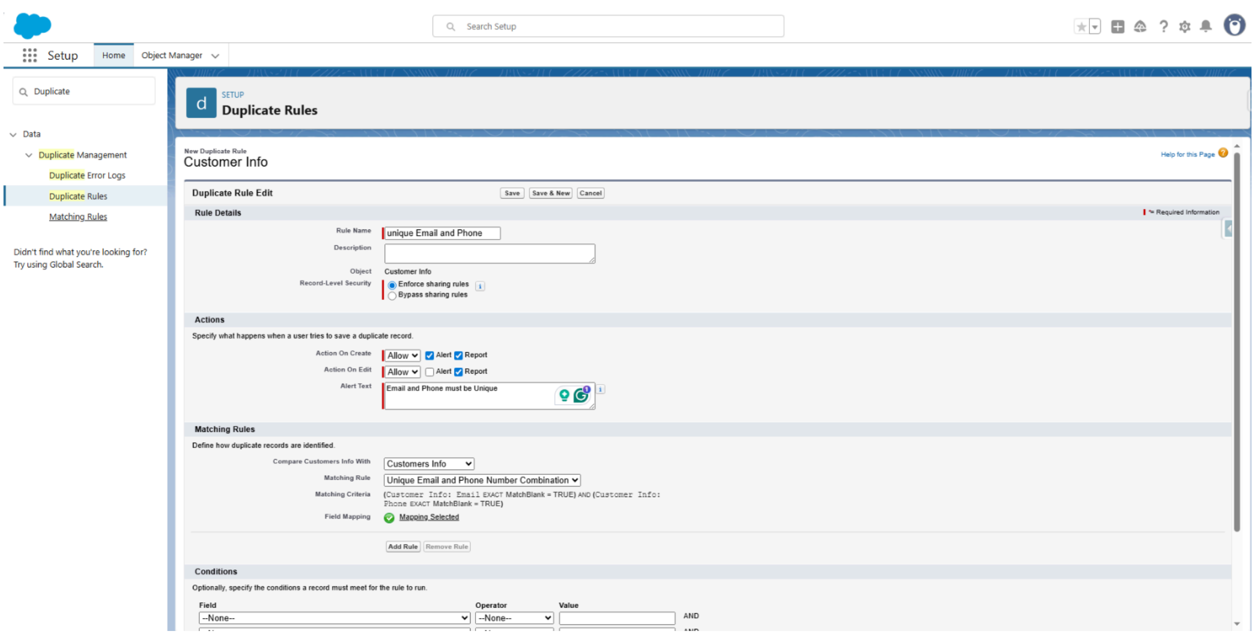


Figure 41. Duplicate Rules

Milestone 23: Profile Management

Profiles define the baseline permissions for users. Two main types were configured:

Standard Profiles (cannot be deleted):

- Examples: System Administrator, Read Only, Standard User

Custom Profiles (created for this project):

- Travel Agent Profile – customized with access to Booking, TravelPackage, Customer Info, etc.
- Travel Agent Manager Profile – granted additional permissions for data review and approval.

Each profile was configured with:

- Object-level permissions (read/create/edit/delete)
- Field-level security
- Tab visibility

This ensured that each user role had only the permissions needed for their function, reducing risk.

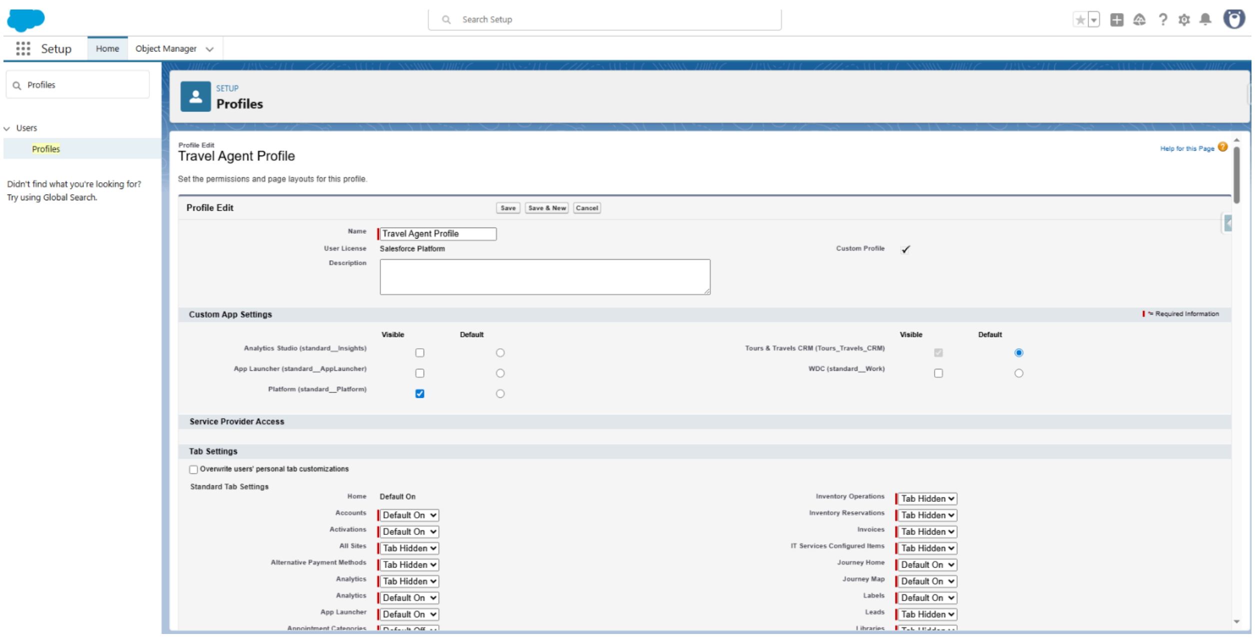


Figure 42. Travel Agent Profile

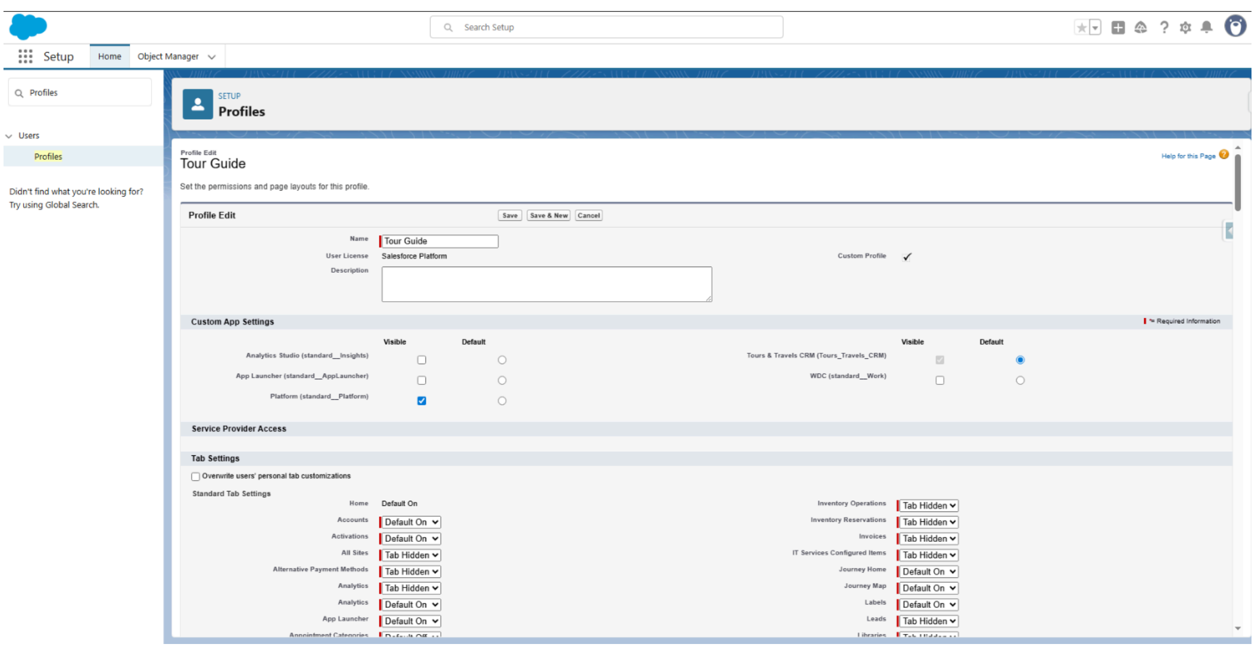


Figure 43. Tour Guide Profile

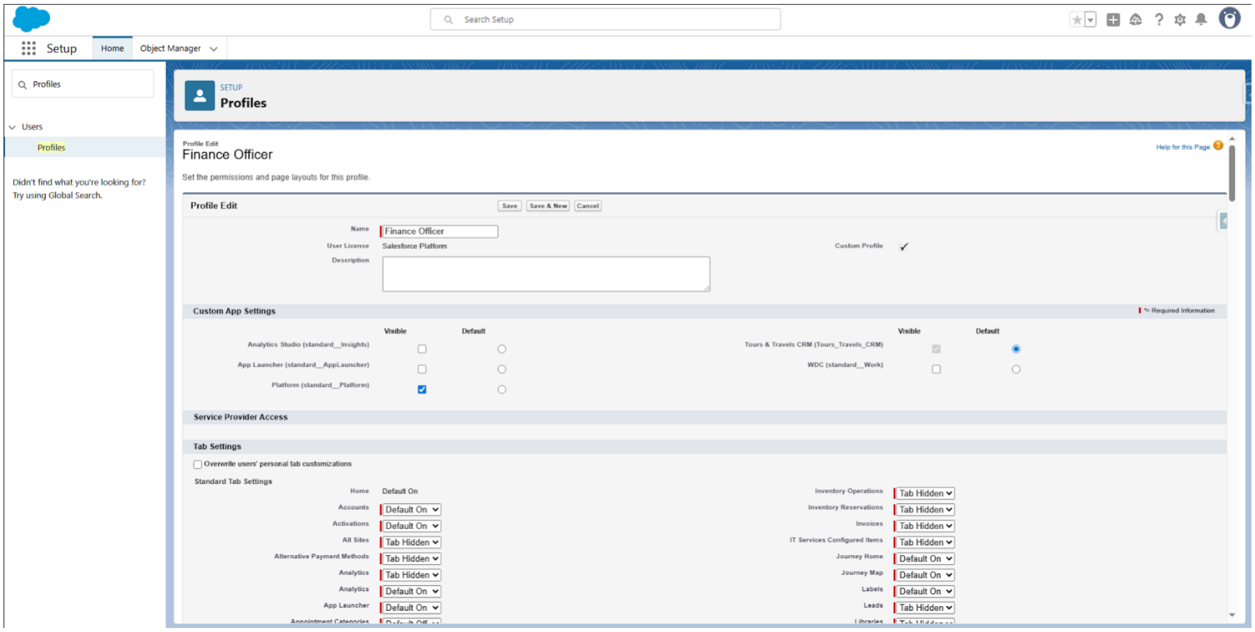


Figure 44. Finance Officer Profile

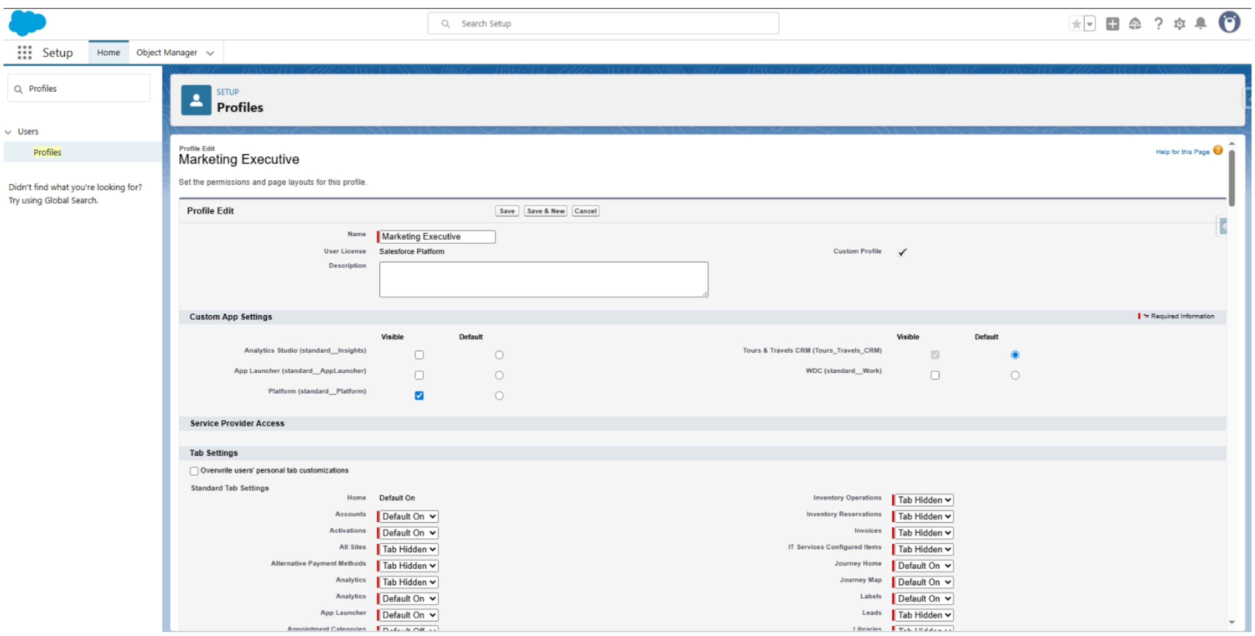


Figure 45. Marketing Executive Profile

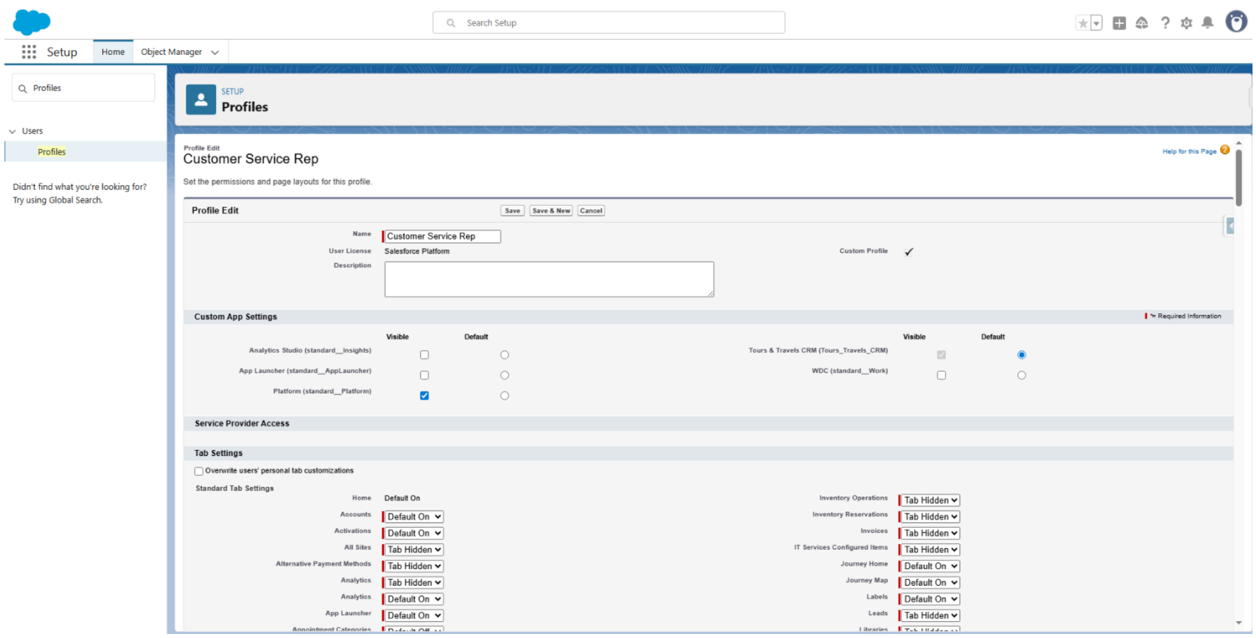


Figure 46. Customer Service Profile

Milestone 24: Roles & Role Hierarchy

Roles determine record-level visibility through the role hierarchy. A top-down structure was created:

- Travel Agent Manager (parent)
    - Travel Agent
        - Tour Guide

This hierarchy ensures:

- Managers can see records owned by agents and guides
- Agents cannot see each other's records unless explicitly shared

- Guides have restricted visibility to only the records shared with them

This structure was critical to enforcing access control while allowing for management oversight.

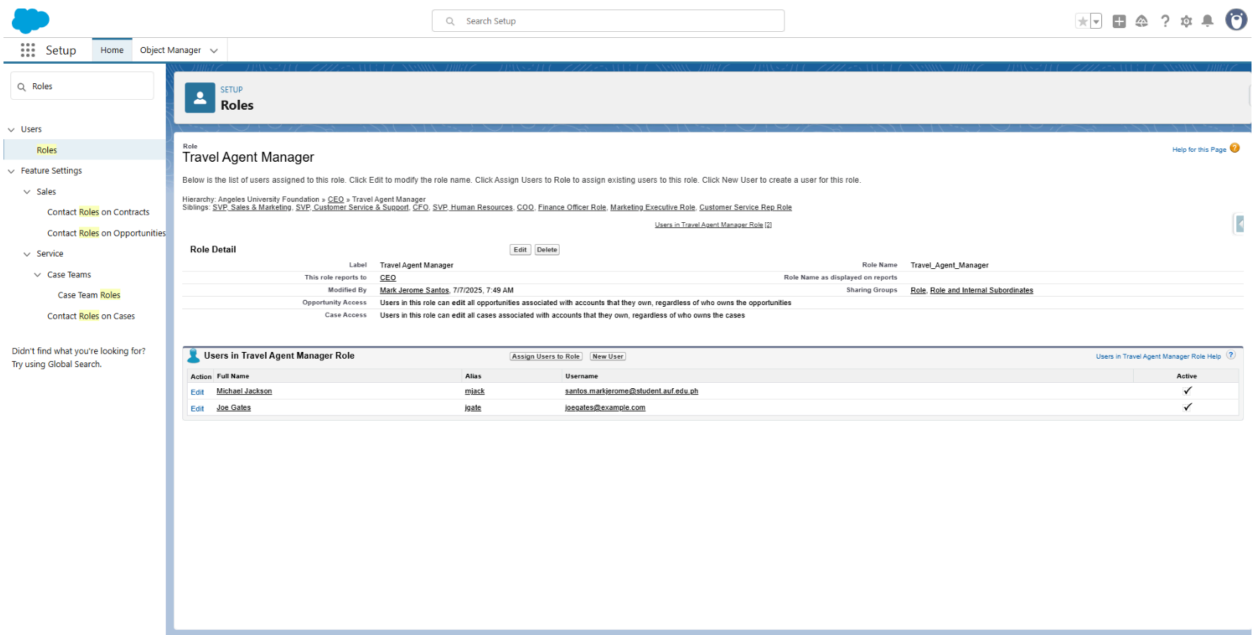


Figure 47. Travel Agent Manager

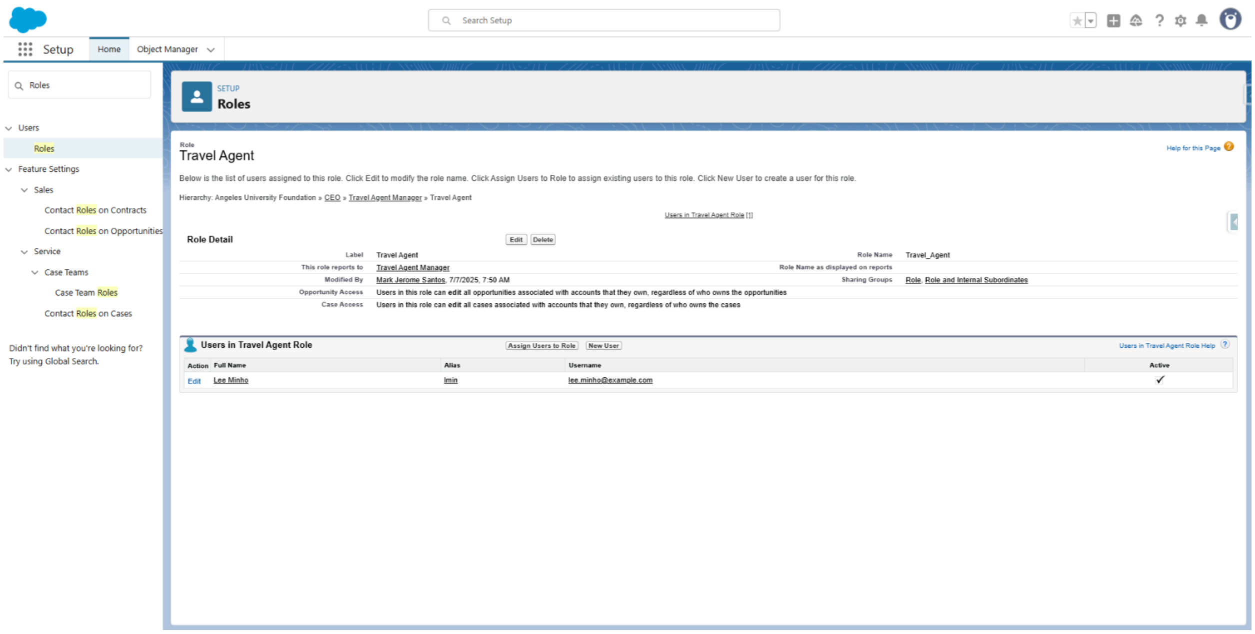


Figure 48. Travel Agent

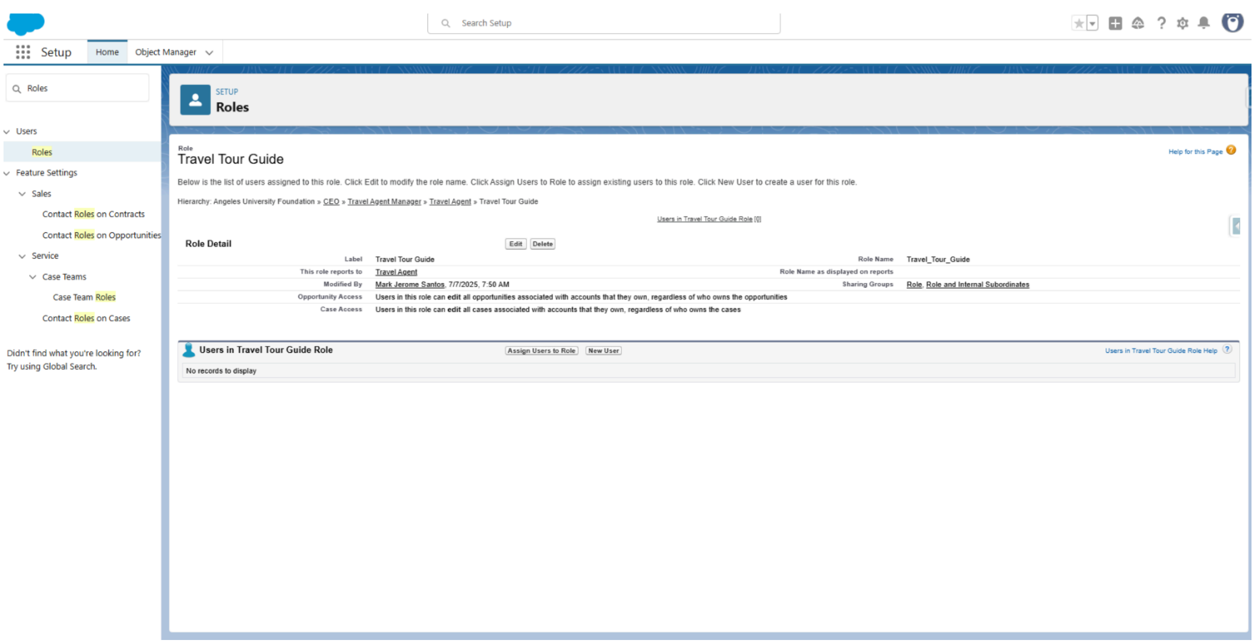


Figure 49. Travel Tour Guide

Milestone 25: Permission Sets

A Permission Set was created to extend access without altering profiles:

- Label: Extra Permission For Travel Agent Manager
- Object: TravelPackage
- Permissions Granted: Read, Edit, Create, Delete
- Assigned To: Users with the Travel Agent Manager Role

This approach ensures flexibility—specific users can be granted access without modifying the broader profile settings.

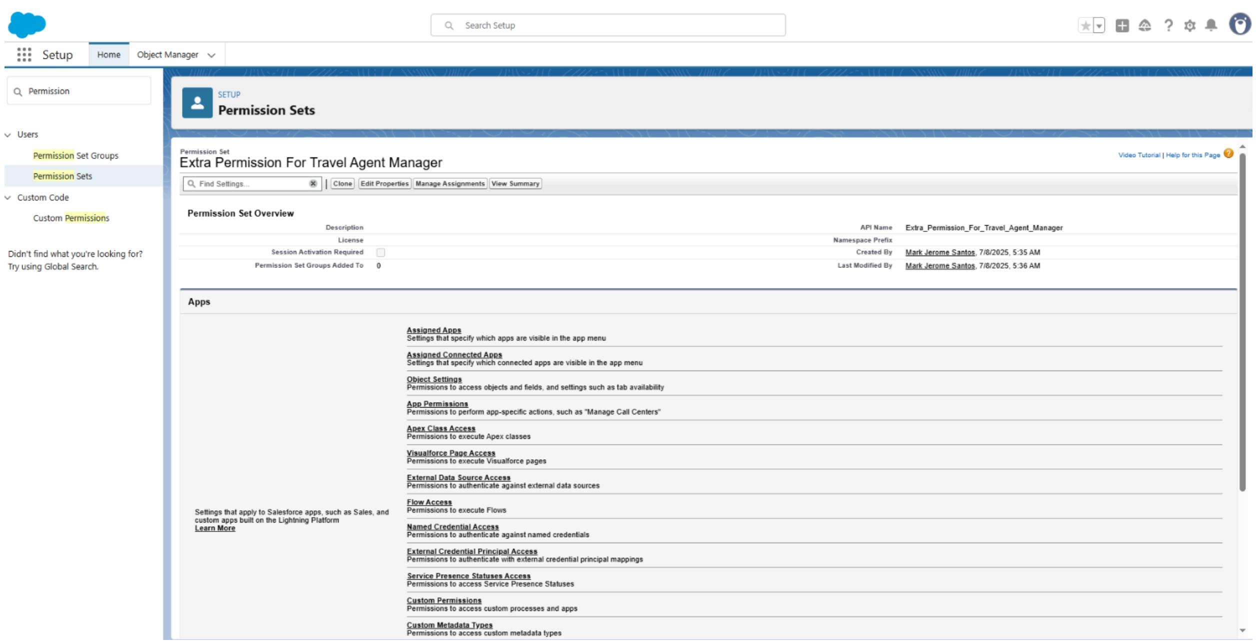


Figure 50. Extra Permission For Travel Agent Manager

Milestone 26: Sharing Settings

Organization-Wide Defaults (OWD):

- Set Customer Info object to Private – ensuring records are only visible to owners and explicitly shared users.

Sharing Rule Configured:

- Label: Customer records auto-shared with Tour Guide Role
- Owner's Role: Travel Agent Role
- Shared With: Tour Guide Role
- Access Level: Read Only

This rule ensures guides can view, but not modify, customer details when assigned to a trip.

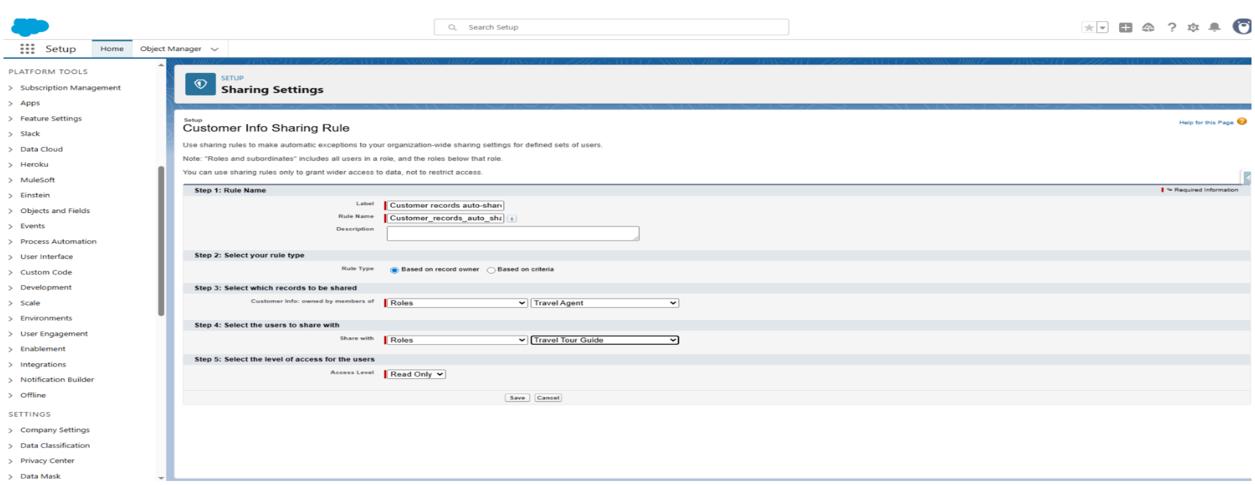


Figure 51. Customer Info Sharing Rule

Milestone 27: Test Classes

Test classes were written to validate that triggers execute correctly. A class named BookingTriggerTest was created.

Highlights:

- Inserts test data for Customer Info, Travel Package, and Booking
- Asserts that a Booking automatically creates:
  - A Booking Payment with status 'Pending'
  - A BookingGuest record per traveler (e.g., 3 guests for 3 travelers)
- Uses Test.startTest() and Test.stopTest() to isolate execution
- Verifies field values with System.assertEquals

This class ensures the trigger logic for booking-related automation works under different input conditions.

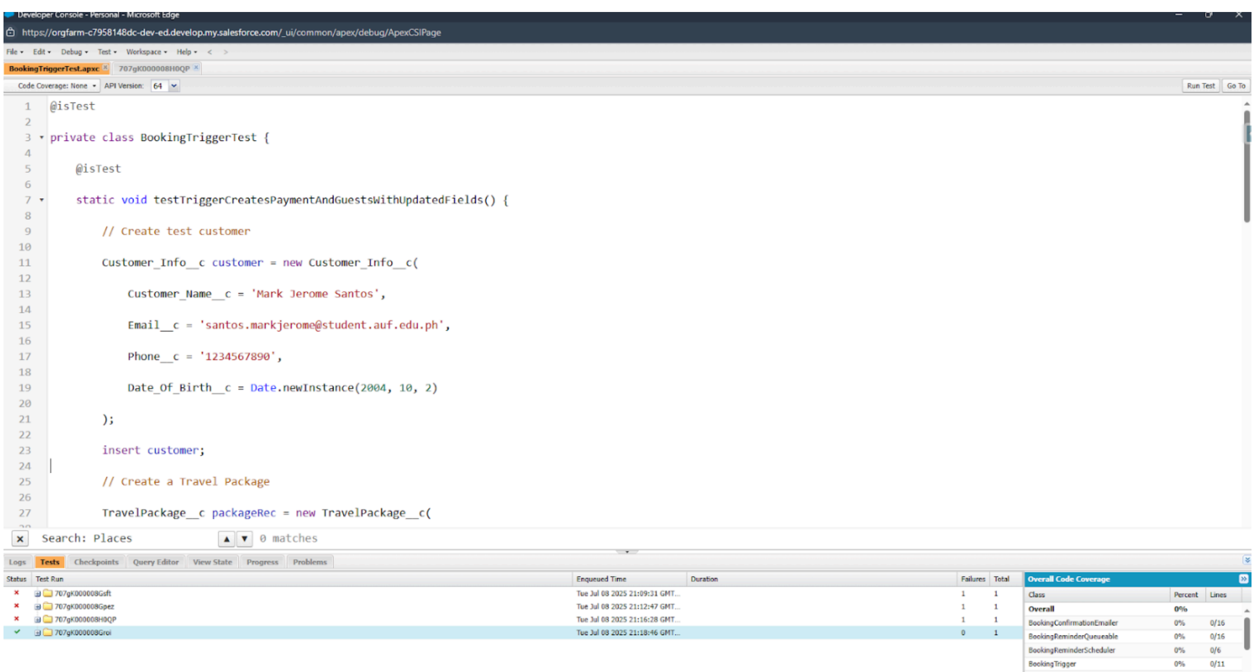


Figure 52. BookingTriggerTest

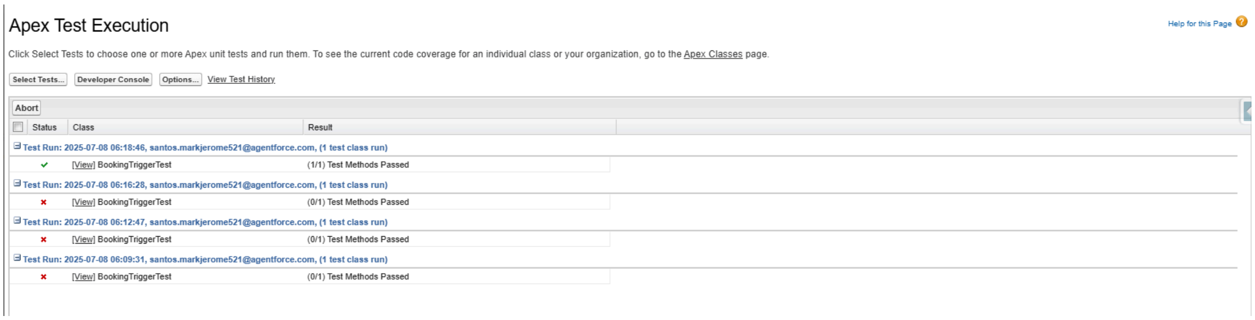


Figure 53. Testing & Troubleshooting

Milestone 28: Preparing Test Cases & Fixing Defects

This milestone focused on systematically verifying that all major functionalities implemented in the Salesforce Tours & Travels CRM application worked as expected. A total of 12 test cases were created, executed, and validated across different modules including Booking, Customer Info, Payment, Automation, and Reporting. Each test case was written based on real-world business scenarios and tested against expected outcomes, ensuring alignment with functional requirements and automation logic.

No.	Test Case Name	Object(s) Involved	Steps	Expected Result	Status
1	Customer Creation	Customer Info	Create and save a customer record	Customer record is created	Passed

				and visible in list view	
2	Booking Creation	Booking, Booking Payment, BookingGuest	Create a booking with required fields	Booking is created, payment and guest records are auto-generated	Passed
3	Payment Status Update	Booking Payment, Booking, Email Trigger	Update Booking Payment status to 'Completed'	Booking status auto-updates to “Confirmed”; confirmation email is sent	Passed
4	Trip Cancellation UI Changes	Booking (Dynamic Forms)	Change Booking Status to 'Cancelled'	Fields (Cancellation Date, Cancel Confirmation, Approval Status) become visible	Passed

5	Customer Record Updates	Customer Info	Edit customer email, phone, or city	Changes are saved with proper validation	Passed
6	Duplicate Prevention	Customer Info	Try creating a new record with existing email and phone	Duplicate warning is displayed; record is not saved	Passed
7	Payment Reminder Notification	Apex (Schedulable + Queueable)	Create a booking 3 days before travel date	Reminder email is sent automatically	Passed
8	Booking Update	Booking	Update fields like number of travelers or travel package	Calculated fields update correctly; record saved	Passed

9	Employee Creation	Employee	Add a new employee with full details	Record is created successfully and visible in tab	Passed
10	Report Creation	Reports	Create a report for bookings with filters	Accurate data shown in report output	Passed
11	Dashboard Creation	Dashboards	Build dashboard using reports on bookings and payments	Visual graphs and stats display correctly	Passed
12	Travel Package Creation	Travel Package	Add a new travel package with pricing and locations	Record is saved and visible in list view	Passed

Milestone 29: Data Import Wizard

In this milestone, the Data Import Wizard tool was used to migrate essential data into the Tours & Travels CRM system. This was a critical step to ensure that key business entities such as customers, travel packages, and employees had real data available in the system for testing and usage.

To bulk-import sample data records for:

- Customer Info
- Travel Package
- Employee

Each object was populated with at least 20 records using well-structured CSV files.

Data Preparation

Three separate .csv files were prepared for the following objects:

- Customer-Info.csv
- Travel-Packages.csv
- Employee.csv

Each file was populated with:

- All required fields for each object
- Standardized formats for picklists (e.g., Country, City, Membership)
- Accurate references for lookups (when applicable)

The data was reviewed for consistency, completeness, and conformance to object-level validations prior to import.

Launching the Data Import Wizard

The Data Import Wizard was accessed via:

Setup → Quick Find → Data Import Wizard → Launch Wizard

From here, each object was imported individually:

Customer Info Object

- Action Selected: Add New Records
- CSV File Used: Customer-Info.csv
- Fields Mapped:
  - Name
  - Email
  - Phone
  - Date of Birth
  - Country
  - City
- Verification: 20 records successfully imported

Travel Package Object

- Action Selected: Add New Records
- CSV File Used: Travel-Packages.csv
- Fields Mapped:
  - Package Name

    - Country
    - Region
    - Places Covered
    - Duration in Days
    - Price Per Person
    - Package Type
    - Membership
    - Transportation Modes
    - Availability Status
    - Guide Languages
    - Meals Included
    - Insurance Included
    - Visa Assistance
    - Guide Included
    - Maximum Group Size
    - Preferred Guide Language
- Verification: 20 records successfully imported

Employee Object

- Action Selected: Add New Records
- CSV File Used: Employee.csv
- Fields Mapped:

  - Name
  - Email
  - Phone
  - Role
  - Department
  - Country
  - City
  - Assigned Region
  - Availability Status
  - Languages Spoken
- Verification: 20 records successfully imported

Field Mapping

In each case, the Data Import Wizard attempted to auto-map the CSV headers to Salesforce fields. Manual verification and adjustments were made to:

- Ensure picklist values matched defined options
- Confirm field-level security allowed access
- Avoid mismatches on similar field names (e.g., Name vs. Full Name)

All mappings were finalized before proceeding.

Monitor Import Jobs

Post-import, the following steps were taken to monitor and verify the jobs:

- Navigated to Setup → Bulk Data Load Jobs
- Confirmed successful completion of all three import jobs
- Verified row counts matched expected import totals
- Validated sample records directly from the respective object tabs (Customer Info, Travel Package, Employee)

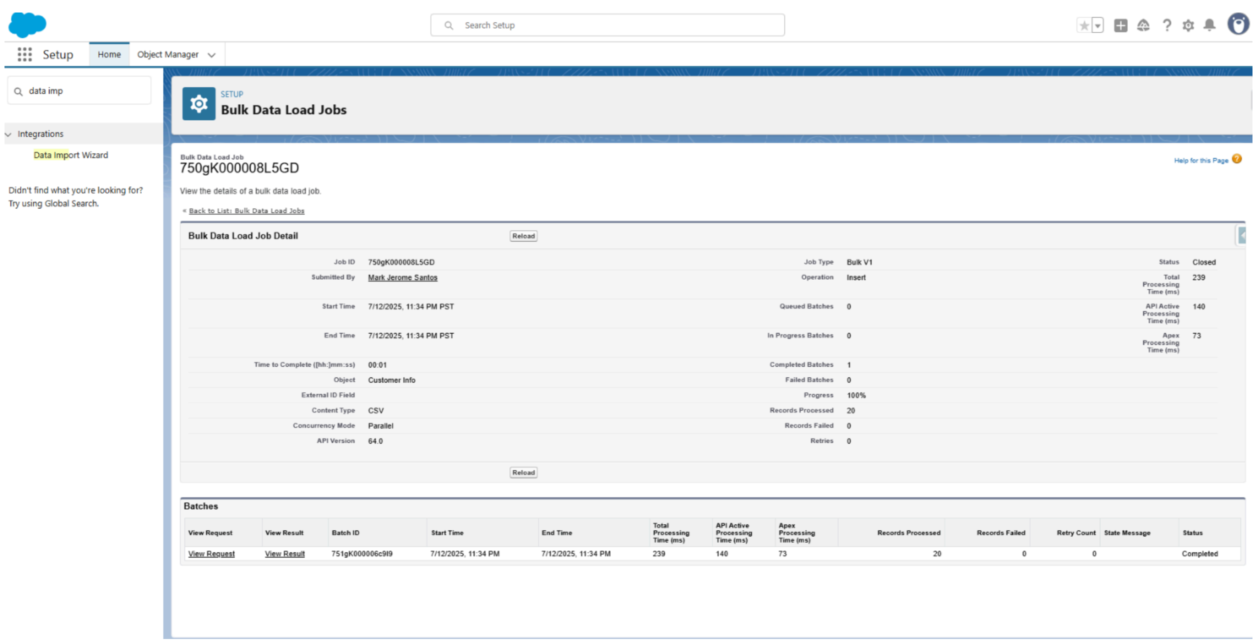


Figure 54. Sample Data Import Result

VII. Phase 5: Deployment, Documentation & Maintenance

This final phase consolidates the delivery of the Tours and Travels CRM system. It covers the deployment strategy, ongoing maintenance, monitoring mechanisms, troubleshooting procedures, testing approach, and forward-looking enhancement ideas to ensure the solution is both reliable and extensible.

Deployment Strategy

The deployment of Salesforce components from the development environment was managed using Change Sets, which is a native Salesforce deployment tool. This method was chosen for its ease of use, metadata bundling capabilities, and support for moving declarative and programmatic components across environments.

Components included in Change Set:

- Custom Objects (e.g., Booking, TravelPackage, Customer Info)
- Custom Fields and Relationships
- Validation Rules, Flows, Workflow Rules, and Process Builders
- Apex Triggers and Classes
- Lightning Pages and Tabs
- Reports and Dashboards
- Profiles, Roles, and Permission Sets

Since the project was deployed in a Salesforce Developer Edition, the deployment was simulated to reflect real-world practices. In a production setup, sandbox to production migration would be handled via Change Sets, Salesforce CLI, or third-party DevOps tools.

Maintenance, Monitoring & Troubleshooting

Maintenance Activities Include:

- Regular review of automation logic (Flows, Process Builders, Triggers).
- Updating TravelPackage or Booking data as per business needs.
- Managing user roles and permissions with organizational changes.
- Reviewing audit logs and field history tracking to ensure data integrity.

Monitoring Mechanisms:

- Monitor Scheduled Jobs via Apex Jobs screen for failures or delays.
- Use Debug Logs to track and investigate automation failures.
- Review Workflow & Process Builder Execution Logs for audit purposes.
- Validate Test Coverage Reports before deploying any code changes.

Troubleshooting Approach:

- Use Debug Logs and Developer Console to trace and fix Apex errors.
- Revalidate Flow logic and decision trees using Flow Debug Mode.
- In case of user-reported issues, impersonate via login access to reproduce errors.
- Resolve formula or automation failures by simulating edge case scenarios.

Testing Approach

Testing was executed throughout the system development cycle using manual validation, unit test classes, and test cases tied to business workflows.

Feature	**Test Type**	**Method**
Object Record Creation	Manual	Form submission validation
Booking Triggers	Automated (Apex Test)	BookingTriggerTest class with assert checks

Flows (Guest Count Check)	Manual & Flow Debug	Simulated traveler mismatch scenarios
Approval Process	Manual	Status change scenario walkthrough
Workflows & Process Builder	Manual	Triggering based on field changes
Reports and Dashboards	Manual	Filter-based data review
Field Visibility (Dynamic)	Manual	Booking Status = Cancelled scenarios
Data Import	Manual	Salesforce Data Import Wizard used

Screenshots of Input and Output were collected and included in the Testing Section of the deliverable for each Salesforce feature implemented.

Documentation Highlights

Documentation was compiled during each phase of the project. It includes:

- Object models, schema diagrams, and picklist definitions
- Automation logic: Flow diagrams, validation logic, formula breakdowns
- Apex code: Triggers, Classes, Test Methods (with coverage results)
- Role hierarchy and permission assignment mappings
- Reports and dashboards visualization summaries
- Input and output screenshots for verification

Each section of the documentation also provides concise annotations and explanations to facilitate understanding for non-technical stakeholders.

Future Enhancements

To extend the CRM's value and usability, the following future enhancements are proposed:

Feature	Description
Chatbot Integration	Connect Einstein Bot to assist customers in real-time with bookings or inquiries.

AI-Based Suggestions	Leverage Salesforce AI to recommend travel packages based on user profile.
Mobile Optimization	Develop a mobile-first layout for customer-facing components.
Real-Time Travel API Integration	Sync with third-party APIs for live transport and weather updates.
Customer Satisfaction Analysis	Add sentiment tracking from Feedback object via AI analysis.

Final Summary & Conclusion

The Tours and Travels CRM capstone project demonstrated the full end-to-end lifecycle of implementing a customized CRM application on the Salesforce platform. Starting from domain research and requirement gathering, the project progressed through structured development, dynamic UI/UX customization, rigorous testing, and deployment simulation. Each phase not only added technical depth but also reflected real-world business workflows and stakeholder needs.

From designing data models and setting up security roles to building complex automation with Apex and Flow, the project incorporated nearly every core feature of Salesforce — including objects, validation rules, approval processes, reports, dashboards, and Lightning Web Components (LWC). Every implementation choice was made with a focus on user experience, scalability, and maintainability.

Beyond the technical deliverables, a major emphasis was placed on documentation, testing, and future planning. The project included:

- Detailed testing scenarios with screenshots
- Debugging strategies using Salesforce tools
- Deployment plans using Change Sets
- Role-specific access control mechanisms
- Real-world automation using asynchronous Apex

The CRM successfully simulated a functioning travel agency ecosystem — managing bookings, travel packages, agents, customers, and payments — all while ensuring data integrity, business automation, and user accessibility.

What I Learned

This capstone pushed me to apply both declarative and programmatic features of Salesforce in a cohesive way. I learned how to:

- Translate user and business requirements into technical specifications
- Use tools like Data Import Wizard, Flow Builder, and Apex Scheduler effectively
- Handle debugging, exception handling, and test class creation in Apex
- Maintain user-centric design through Lightning App pages, dynamic forms, and reports
- Adopt real-world deployment strategies and best practices for CRM maintenance

The entire project serves as both a portfolio showcase and a learning archive that reflects hands-on Salesforce development, end-user empathy, and enterprise-grade solution architecture.

With the foundation now in place, this CRM system can be enhanced with features such as AI recommendations, chatbot integration, and API connectivity to elevate it into a more intelligent, connected platform for global travel businesses.

References

- Salesforce Data Modeling Guide
- Salesforce Official Documentation
- Travel Industry Reports and Use Cases via Google
- ChatGPT