Tours & Travels Platform Development Using CRM-Salesforce

A Capstone Project
Presented to
SmartBridge x Salesforce

By
Cariño, Marcus Jeremy M.

July 18, 2025

Project Overview

The Tours and Travels CRM is a custom Salesforce-based solution designed to address the operational needs of global travel agencies. It centralizes booking management, customer engagement, employee collaboration, and financial tracking into a unified platform. The system streamlines core processes such as trip bookings, payment tracking, customer onboarding, and feedback handling. Through tailored automation and role-based access controls, the CRM enhances service efficiency, reduces manual tasks, and enables data-driven decision-making for tour operators and managers.

Objectives

The primary objective of the Tours and Travels CRM is to build a scalable and intelligent system capable of managing the complete lifecycle of travel bookings and customer interactions. The project aims to:

- Automate booking processes and payment confirmations to minimize delays and human errors.
- Improve customer satisfaction through personalized communication and timely updates.
- Enable employee-specific access and functionality via roles, profiles, and permissions.
- Collect and analyze business insights through customized reports and dashboards.
- Maintain data security and integrity through validation rules, sharing settings, and audit trails.
- Provide a user-friendly interface tailored to the responsibilities of agents, guides, finance officers, and service representatives.

Phase 1: Requirement Analysis & Planning

Understanding Business Requirements

The project began with a comprehensive study of how tours and travel agencies operate globally. The analysis revealed several recurring challenges:

- Manual booking processes often result in slow responses and data inconsistencies.
- Fragmented communication between customers, agents, and finance staff hinders coordination.
- Lack of systematic tracking for guest information, payments, and tour feedback.
- Difficulty managing packages based on geography, group size, and membership levels.

Stakeholder interviews and secondary research were conducted to understand the roles and expectations of customers, travel agents, finance officers, and tour guides. Key user needs identified include real-time status updates, customizable tour options, and simplified task handling for internal staff.

Defining Project Scope and Objectives

The scope of this CRM includes designing a complete digital system for:

- Customer onboarding and data management
- Tour package listing and filtering
- Booking creation and guest tracking

- Payment processing and monitoring
- Automated follow-ups and feedback collection
- Employee assignment and task tracking

Project objectives were shaped to address these needs by leveraging Salesforce's declarative and programmatic capabilities. The CRM is built to serve not only individual travelers but also groups and corporate clients, offering flexible pricing and service structures.

Designing the Data Model and Security Model

The system's data model was designed around several key entities and their relationships:

- `Booking__c` is the core object linked to `Customer__c`, `TravelPackage__c`, and `Employee__c`.
- `BookingGuest__c` serves as a master-detail child of `Booking__c`, tracking individual travelers per booking.
- `BookingPayment__c` ensures traceability of transaction status for each booking.
- `Feedback__c` collects structured ratings and remarks linked to past trips.

Security was enforced through a combination of:

- **Role hierarchy**: enabling managers to view subordinate records.
- **Profiles**: defining object and field-level access based on job function.
- **Permission sets**: granting extra access for specific users as needed.
- **Field-level security**: hiding sensitive financial or contact data from unauthorized roles.

- **Sharing rules**: allowing controlled access to records across different user groups.

Creating the Project Roadmap and Milestones

A milestone-based roadmap was outlined to structure the development process. Each milestone addressed a functional layer of the system—from foundational object creation to advanced automation and testing. Milestones were grouped into five key phases:

- Phase 1: Requirement Analysis & Planning
- Phase 2: Salesforce Backend Development
- Phase 3: UI/UX Customization
- Phase 4: Data Migration, Testing & Security
- Phase 5: Deployment, Maintenance & Documentation

Phase 2: Salesforce Development – Backend & Configurations

Setup Environment and DevOps Workflow

The development environment was initialized through the Salesforce Developer Edition platform. This served as the base instance for configuring objects, automations, and Apex logic. Initial setup activities included:

- Activating the Developer Edition and accessing Setup.
- Verifying object creation permissions.
- Familiarizing with Salesforce CLI and Developer Console for Apex code deployment.
- Establishing naming conventions for objects, fields, and automation processes to maintain consistency throughout the application lifecycle.

Though this project did not require full DevOps integration or CI/CD pipelines, care was taken to separate configuration tasks logically to simulate modular development best practices.

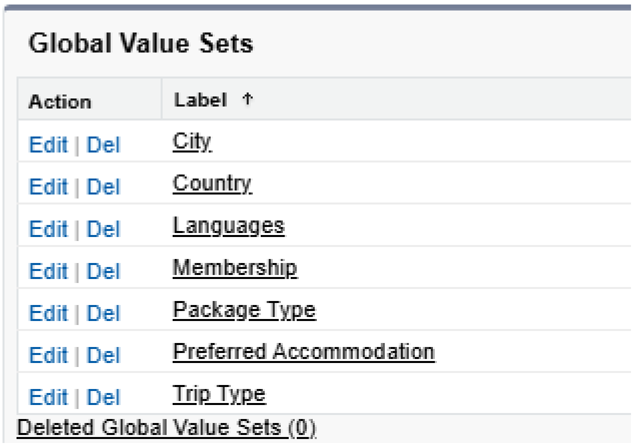
Customization of Objects, Fields, and Validation Rules

Several custom objects were created to support core business processes:

- `Customer_Info__c`: Manages customer contact and demographic data.
- `Booking__c`: Tracks reservation details and is linked to customer and tour information.
- `TravelPackage__c`: Stores tour-related details such as country, price, duration, and availability.
- `BookingGuest__c`: Captures guest-specific data linked to a Booking record.
- `BookingPayment__c`: Handles payment status, type, and amount for bookings.

- `Employee__c`: Manages internal team members, including their department and role.
- `Feedback__c`: Collects post-travel evaluations from customers.

Each object was enhanced with the necessary custom fields, including picklists, formulas, checkboxes, and roll-up summary fields. Global value sets were implemented for fields like Country, City, Membership Type, and Department to standardize inputs. Below are related screenshots of object creation, custom fields, and formulas.

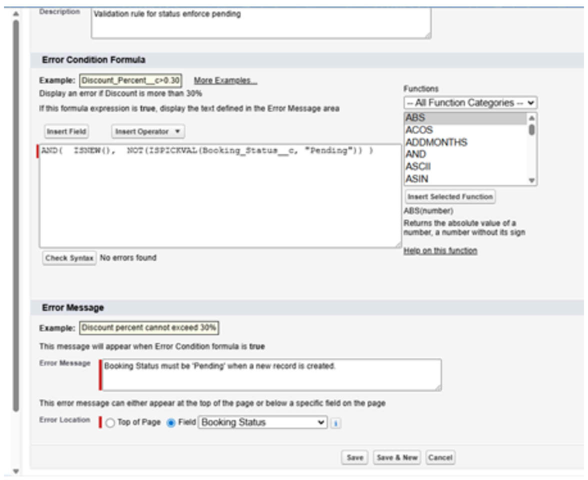


Global Value Sets

Action	Label ↑
Edit | Del	City
Edit | Del	Country
Edit | Del	Languages
Edit | Del	Membership
Edit | Del	Package Type
Edit | Del	Preferred Accommodation
Edit | Del	Trip Type

Deleted Global Value Sets (0)

Figure 1. Global Value Sets



Description: Validation rule for status enforce pending

Error Condition Formula

Example: Discount_Percent__c>0.30 More Examples...

Display an error if Discount is more than 30%

If this formula expression is true, display the text defined in the Error Message area

```
AND(  ISNEW(),  NOT(ISPICKVAL(Booking_Status__c, "Pending")) )
```

No errors found

Functions: -- All Function Categories --

ABS(number)

Returns the absolute value of a number, a number without its sign

Error Message

Example: Discount percent cannot exceed 30%

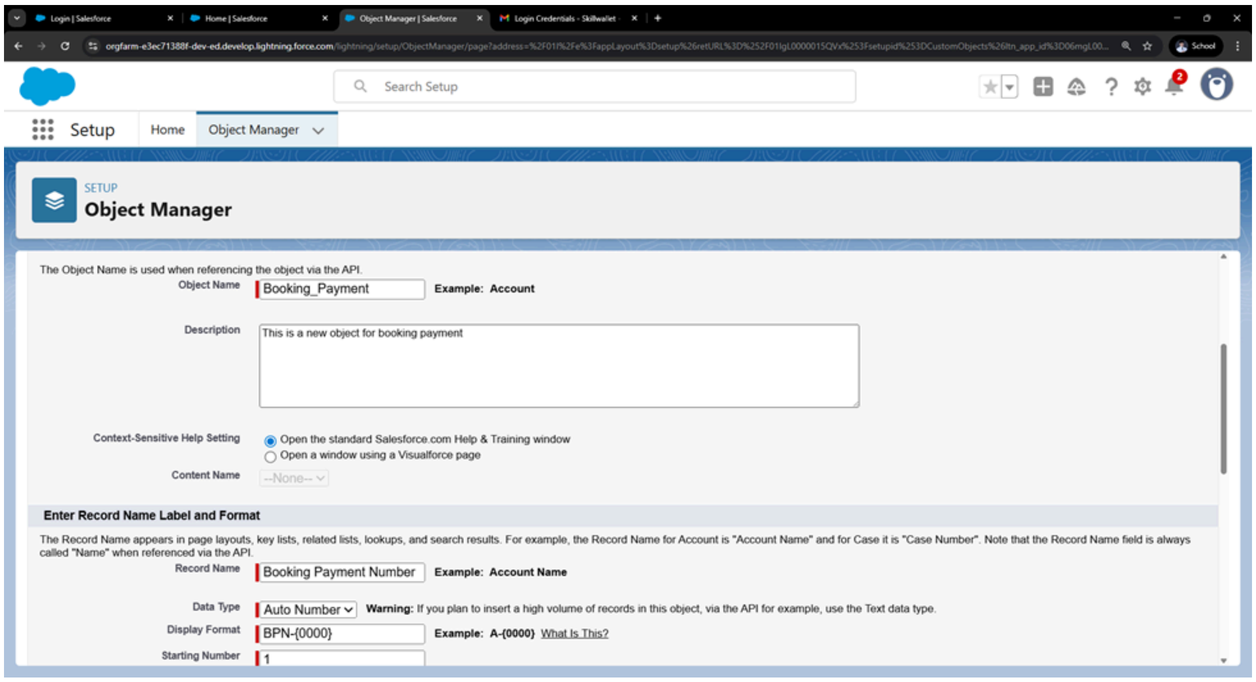
This message will appear when Error Condition formula is true

Error Message: Booking Status must be 'Pending' when a new record is created.

This error message can either appear at the top of the page or below a specific field on the page

Error Location: ○ Top of Page [✓ Field] Booking Status

Figure 2. Code for Booking Status as default "Pending"



SETUP

Object Manager

The Object Name is used when referencing the object via the API.

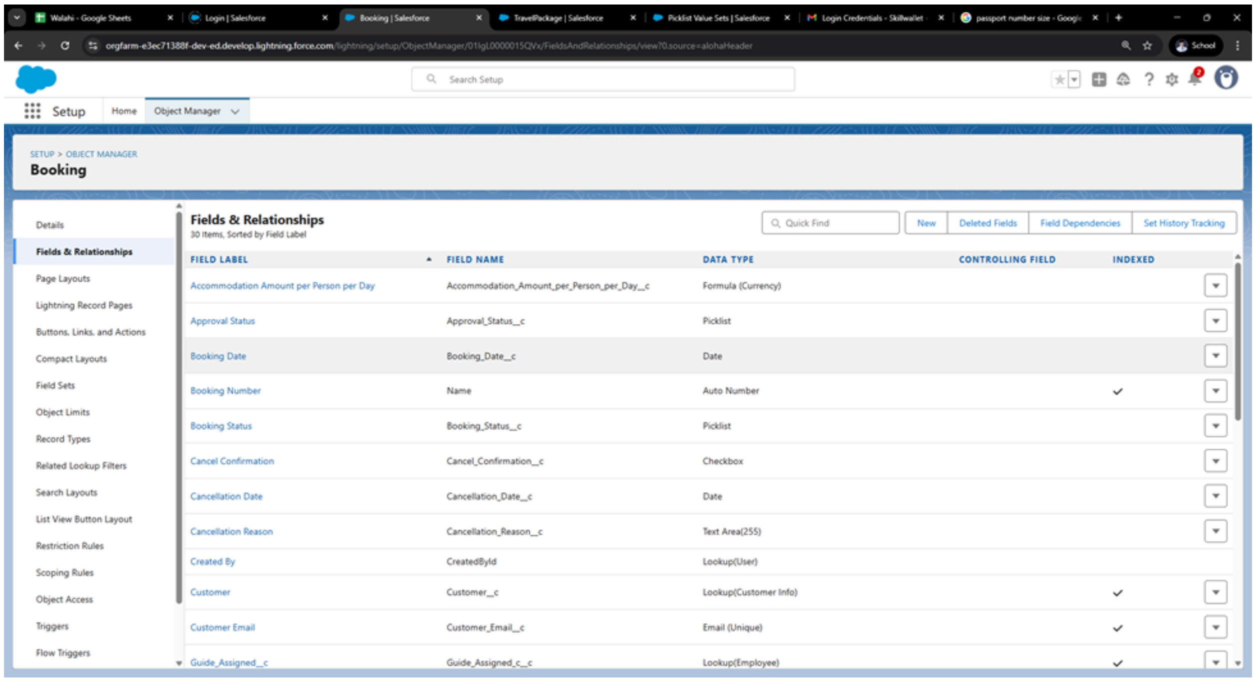
Object Name	Booking_Payment	**Example: Account**
Description	This is a new object for booking payment	
Context-Sensitive Help Setting	[✓ Open the standard Salesforce.com Help & Training window] Open a window using a Visualforce page	
Content Name	--None--	

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name	Booking Payment Number	**Example: Account Name**
Data Type	Auto Number	**Warning:** If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.
Display Format	BPN-{0000}	**Example: A-{0000}** What Is This?
Starting Number	1	

Figure 3. Sample Object Creation



SETUP > OBJECT MANAGER

Booking

Fields & Relationships
30 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Accommodation Amount per Person per Day	Accommodation_Amount_per_Person_per_Day__c	Formula (Currency)		
Approval Status	Approval_Status__c	Picklist		
Booking Date	Booking_Date__c	Date		
Booking Number	Name	Auto Number		✓
Booking Status	Booking_Status__c	Picklist		
Cancel Confirmation	Cancel_Confirmation__c	Checkbox		
Cancellation Date	Cancellation_Date__c	Date		
Cancellation Reason	Cancellation_Reason__c	Text Area(255)		
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Lookup(Customer Info)		✓
Customer Email	Customer_Email__c	Email (Unique)		✓
Guide_Assigned__c	Guide_Assigned_c__c	Lookup(Employee)		✓

Figure 4. Sample Fields and Relationships

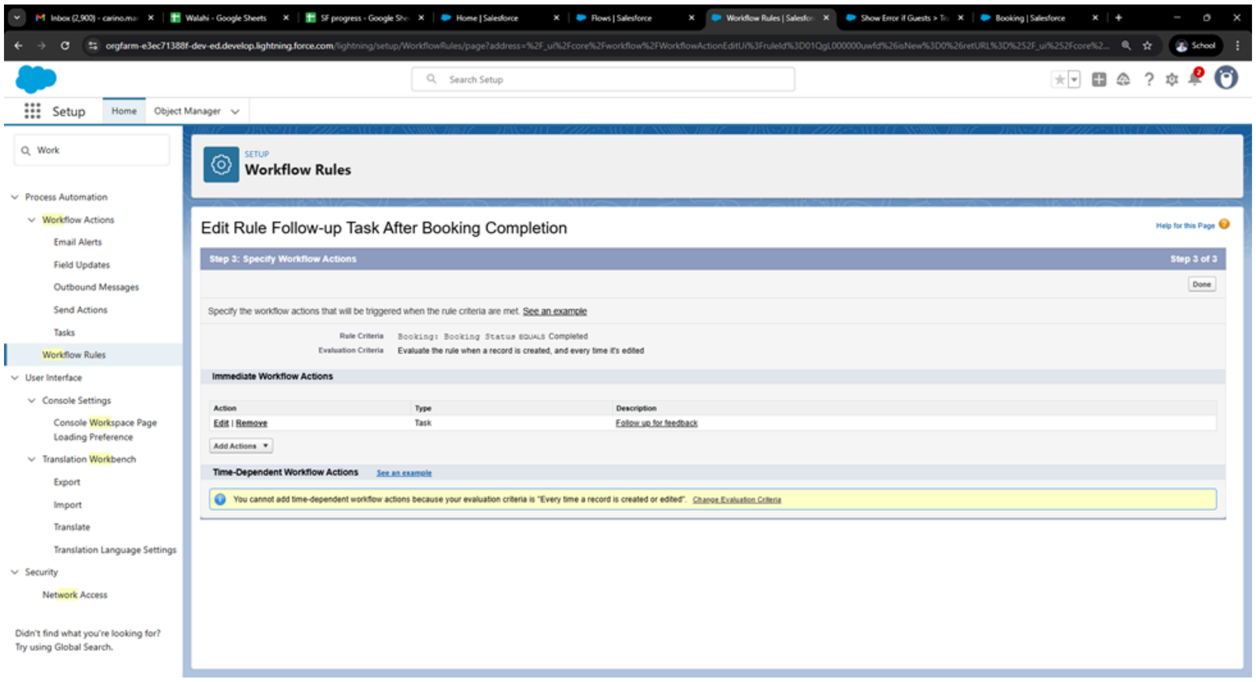
Validation rules were added to enforce data integrity:

- Phone numbers must be exactly 10 digits.
- Emails must follow a valid format.
- Guests cannot have future-dated birthdays.
- Mandatory fields are conditional based on role selection (e.g., Language for Guides).
- Age must be greater than zero.
- Booking Status must default to "Pending" upon record creation.

Automation: Workflow Rules, Process Builder, Flows, and Approval Process

A layered automation strategy was applied using Salesforce's declarative tools:

- **Workflow Rule**: Automatically creates a Task record for the Booking Owner when a Booking is marked as Completed. The task reminds the owner to collect feedback, with a due date set three days after the travel end date.



Edit Rule Follow-up Task After Booking Completion

Step 3: Specify Workflow Actions — Step 3 of 3

Specify the workflow actions that will be triggered when the rule criteria are met. See an example

Rule Criteria	Booking: Booking Status EQUALS Completed
Evaluation Criteria	Evaluate the rule when a record is created, and every time it's edited

Immediate Workflow Actions

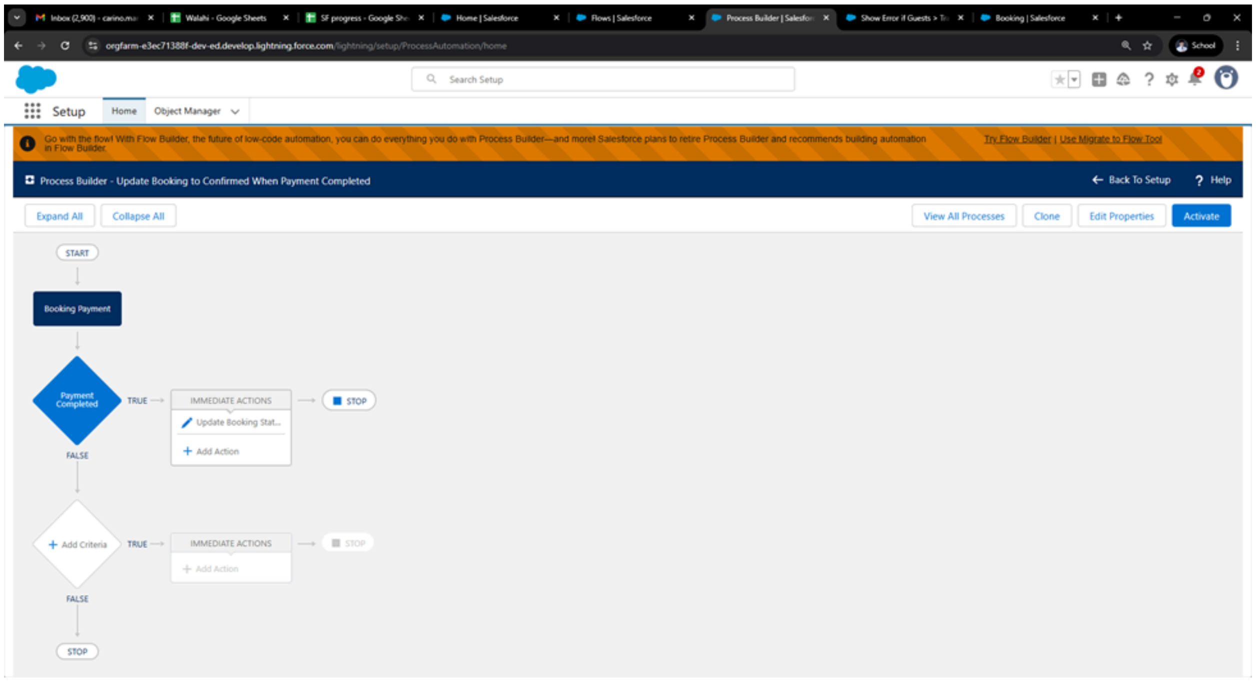
Action	Type	Description
Edit | Remove	Task	Follow up for feedback

Time-Dependent Workflow Actions See an example

You cannot add time-dependent workflow actions because your evaluation criteria is "Every time a record is created or edited". Change Evaluation Criteria

Figure 5. Sample Workflow Rule

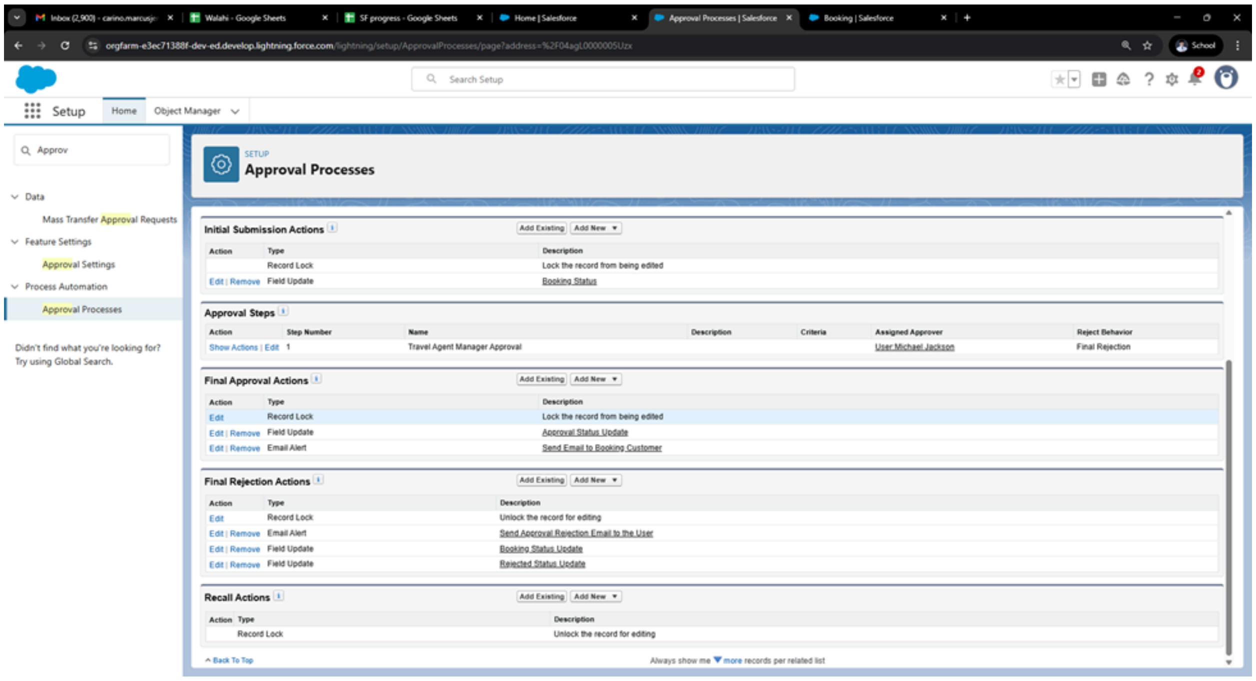
- **Process Builder**: Updates the `Booking_Status__c` field to "Confirmed" when the related `BookingPayment__c` record is marked as "Completed".


Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder.

Process Builder - Update Booking to Confirmed When Payment Completed

Figure 6. Sample Process Builder

- **Approval Process**: Enables multi-step review and approval of Booking cancellation requests. Notifications and field updates were configured to occur based on approval outcomes, with custom email templates sent to involved parties.



Approval Processes

Initial Submission Actions

Action	Type	Description
	Record Lock	Lock the record from being edited
Edit | Remove	Field Update	Booking Status

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions | Edit	1	Travel Agent Manager Approval			User Michael Jackson	Final Rejection

Final Approval Actions

Action	Type	Description
Edit	Record Lock	Lock the record from being edited
Edit | Remove	Field Update	Approval Status Update
Edit | Remove	Email Alert	Send Email to Booking Customer

Final Rejection Actions

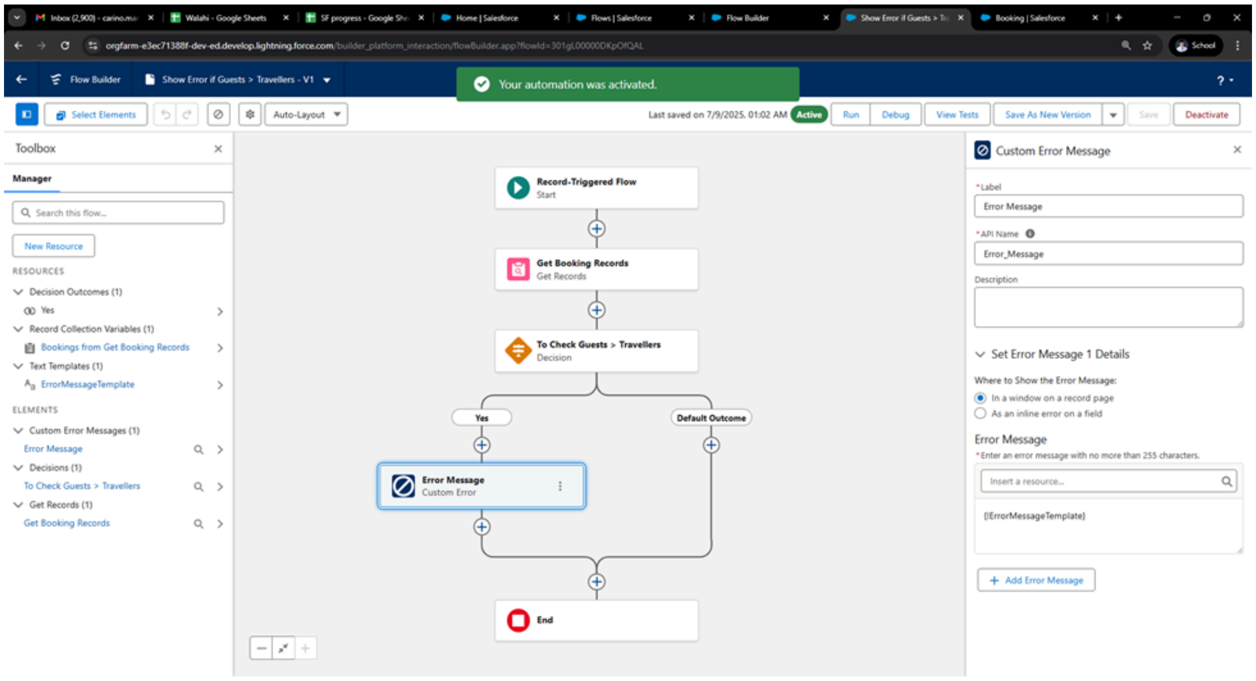
Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit | Remove	Email Alert	Send Approval Rejection Email to the User
Edit | Remove	Field Update	Booking Status Update
Edit | Remove	Field Update	Rejected Status Update

Recall Actions

Action	Type	Description
	Record Lock	Unlock the record for editing

Figure 7. Sample Approval Process

- **Flow (Before Save)**: A record-triggered Flow was created on the `BookingGuest__c` object. It ensures that the number of guest records does not exceed the `Number_of_Travelers__c` value in the parent Booking. This helps enforce guest count consistency without Apex intervention.



*Label: Error Message

*API Name: Error_Message

Description:

Set Error Message 1 Details

Where to Show the Error Message:
[✓ In a window on a record page]
As an inline error on a field

Error Message
*Enter an error message with no more than 255 characters.

{!ErrorMessageTemplate}

Figure 8. Sample Flow

Apex Classes, Triggers, and Asynchronous Apex

Where declarative tools were insufficient, Apex code was implemented:

- **Trigger on Booking**: When a Booking is created, it automatically generates:
  - One `BookingPayment__c` record with default status "Pending".
  - Multiple `BookingGuest__c` records matching the number of travelers.

- **Trigger Handler Pattern**: A trigger handler was used to centralize and maintain logic readability.
- **Future Apex**: An asynchronous email confirmation is triggered when a Booking is updated and its status becomes "Confirmed". This reduces blocking operations and ensures smooth user experience.

- **Queueable Apex + Scheduler**: A reminder email is sent to customers three days before their travel start date. This job is scheduled to run daily, ensuring timely communication.
- **Batch Apex + Scheduler**: Another scheduled batch job sends payment reminders for bookings created the previous day with status still marked as "Pending". A summary email is sent to the administrator after batch execution completes.

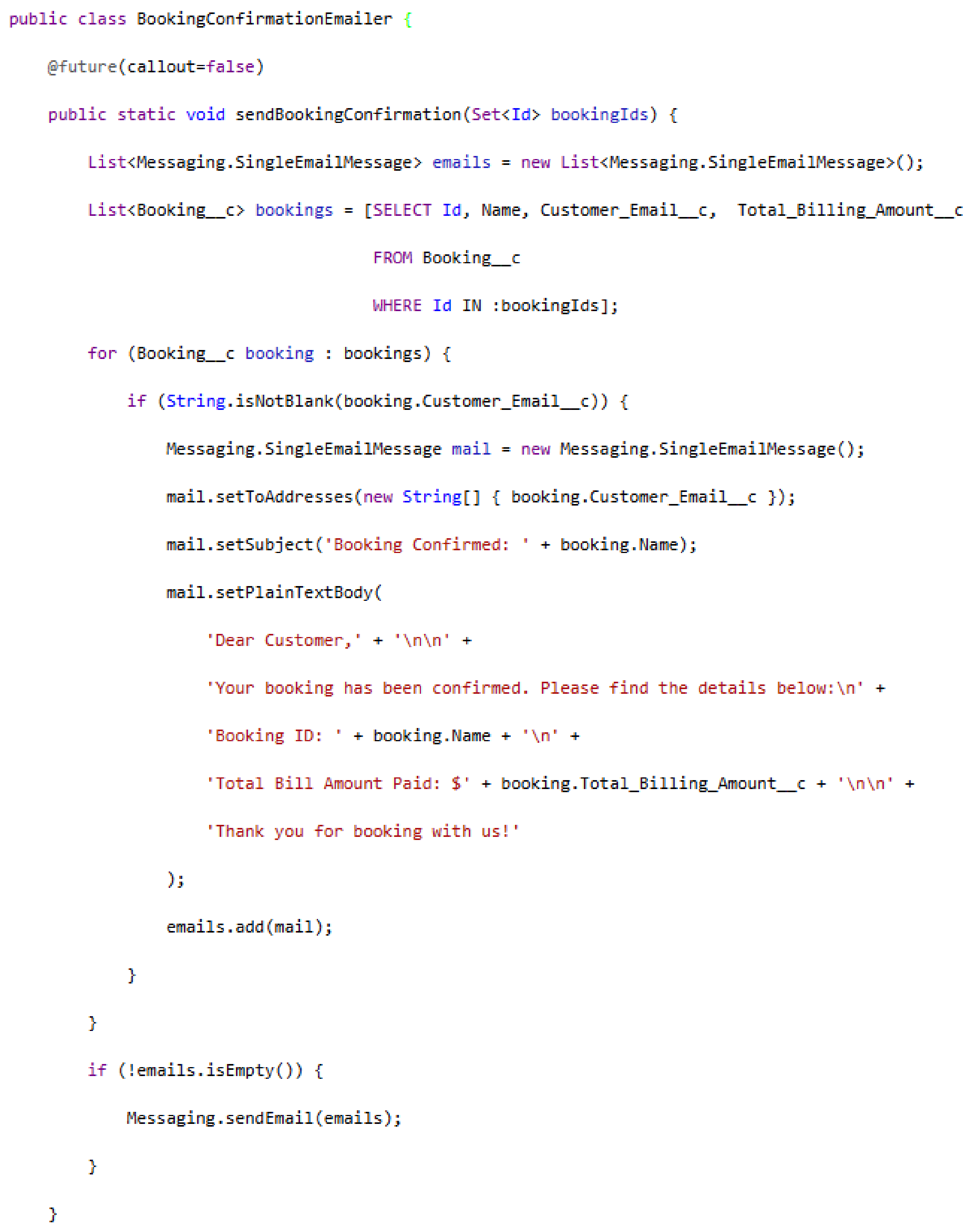
All custom logic was written following best practices for governor limits, bulkification, and test coverage.

To automate email notifications upon confirmation of a booking, a custom Apex Trigger was implemented. This logic ensures that an email is only sent when the **Booking Status** changes from any other value to `"Confirmed"`.

```
for (Booking__c booking : Trigger.new) {
    Booking__c oldBooking = Trigger.oldMap.get(booking.Id);
    // Check for status change to "Confirmed"
    if (booking.Booking_Status__c == 'Confirmed' && oldBooking.Booking_Status__c != 'Confirmed') {
        bookingIdsToSend.add(booking.Id);
    }
}
 if (!bookingIdsToSend.isEmpty()) {
    // Call the Future Method
    BookingConfirmationEmailer.sendBookingConfirmation(bookingIdsToSend);
}
```

Figure 9. Sample Snippet of Code

To enhance system responsiveness and avoid timeouts during record updates, a **Future Apex class** was implemented to handle outbound email processing after a booking is confirmed.



```
public class BookingConfirmationEmailer {
    @future(callout=false)
    public static void sendBookingConfirmation(Set<Id> bookingIds) {
        List<Messaging.SingleEmailMessage> emails = new List<Messaging.SingleEmailMessage>();
        List<Booking__c> bookings = [SELECT Id, Name, Customer_Email__c,  Total_Billing_Amount__c
                                     FROM Booking__c
                                     WHERE Id IN :bookingIds];
        for (Booking__c booking : bookings) {
            if (String.isNotBlank(booking.Customer_Email__c)) {
                Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
                mail.setToAddresses(new String[] { booking.Customer_Email__c });
                mail.setSubject('Booking Confirmed: ' + booking.Name);
                mail.setPlainTextBody(
                    'Dear Customer,' + '\n\n' +
                    'Your booking has been confirmed. Please find the details below:\n' +
                    'Booking ID: ' + booking.Name + '\n' +
                    'Total Bill Amount Paid: $' + booking.Total_Billing_Amount__c + '\n\n' +
                    'Thank you for booking with us!'
                );
                emails.add(mail);
            }
        }
        if (!emails.isEmpty()) {
            Messaging.sendEmail(emails);
        }
    }
```

Figure 10. Sample Snippet of Code

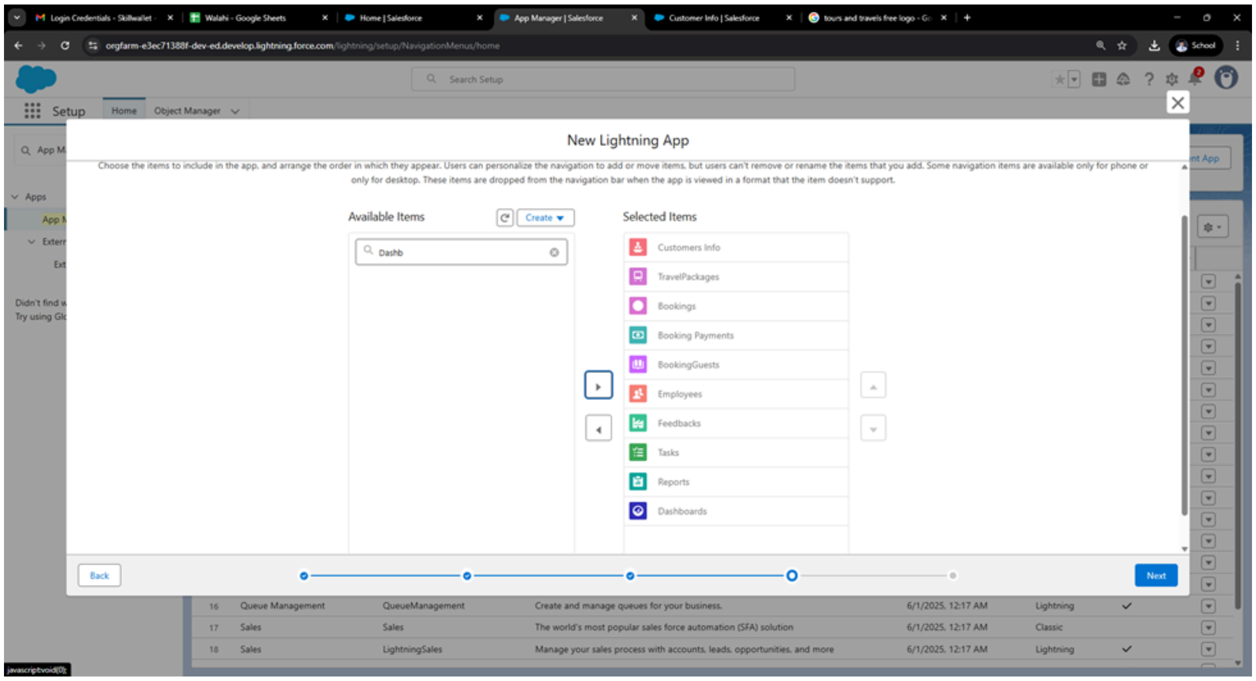
Phase 3: UI/UX Development & Customization

This phase of development was centered on enhancing user experience by customizing the interface of the Tours and Travels CRM. Through the use of Lightning components, page layout configurations, user account setup, and reporting tools, the application was optimized for role-based usage, clarity, and real-time data presentation. The goal was to provide a seamless and intuitive environment that supports day-to-day travel operations while adhering to data visibility requirements across roles.

Lightning App Configuration

A dedicated Lightning App named **"Tours & Travels CRM"** was created using the App Manager in Salesforce Setup. The app serves as the primary interface for users to access various objects and functionalities. It includes custom tabs for:

- Customer Info
- Travel Packages
- Bookings
- Booking Guests
- Employees
- Booking Payments
- Feedback
- Reports and Dashboards



Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Figure 11. Lightning App for Tours & Travels CRM

This setup enables centralized navigation and modular access based on user roles.

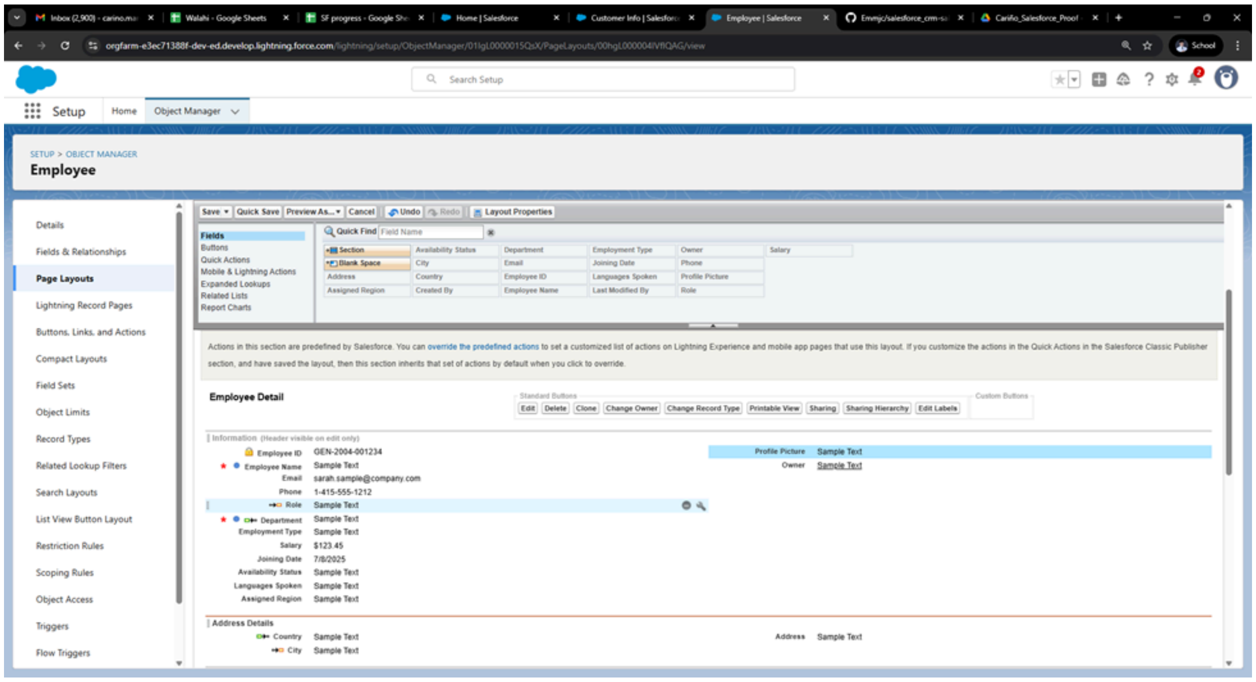
Page Layouts and Dynamic Forms

Each major object was customized with an appropriate page layout to improve data entry and visibility. Fields were logically grouped and reordered to follow business process flows.

Dynamic Forms were enabled particularly for the **Booking** object to achieve conditional field visibility. For example:

- The **Cancellation Date** and **Approval Status** fields are only displayed when the **Booking Status** is set to "Cancelled".
- Conditional rendering was applied for both desktop and mobile platforms, ensuring consistency across devices.

This approach reduced form clutter and guided users based on the current context of the record.



Actions in this section are predefined by Salesforce. You can override the predefined actions to set a customized list of actions on Lightning Experience and mobile app pages that use this layout. If you customize the actions in the Quick Actions in the Salesforce Classic Publisher section, and have saved the layout, then this section inherits that set of actions by default when you click to override.

Employee Detail

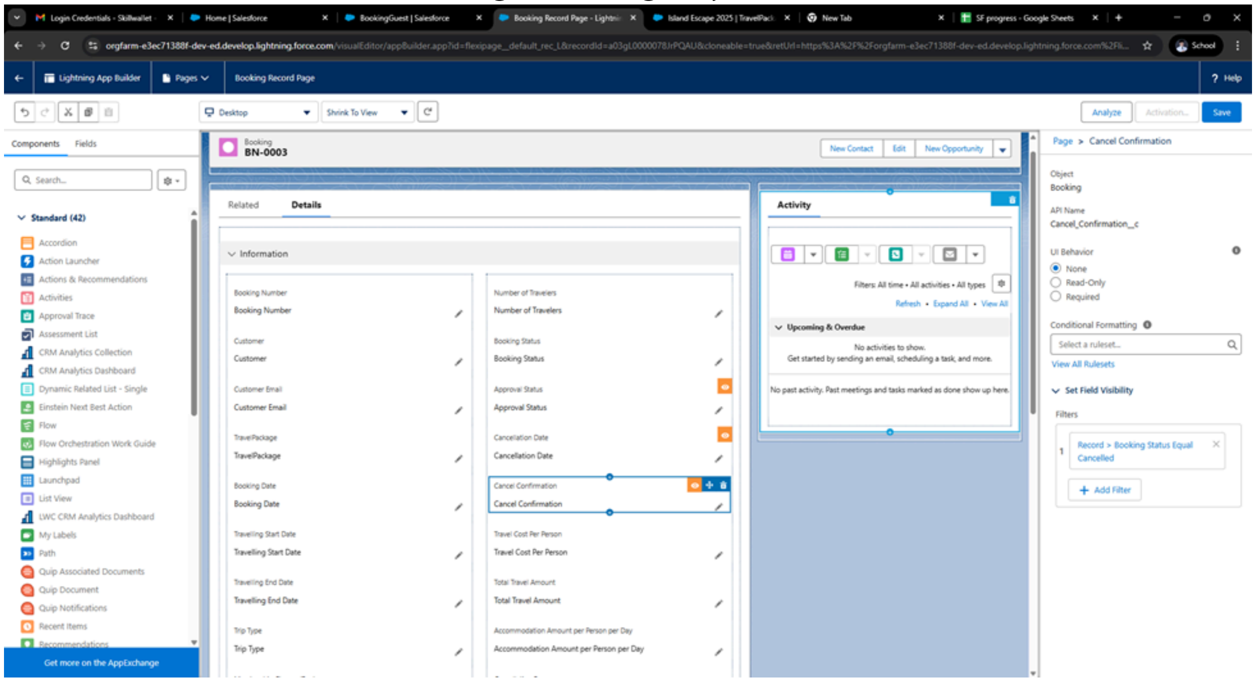
Information (Header visible on edit only)

Employee ID	GEN-2004-001234
Employee Name	Sample Text
Email	sarah.sample@company.com
Phone	1-415-555-1212
Role	Sample Text
Department	Sample Text
Employment Type	Sample Text
Salary	$123.45
Joining Date	7/8/2025
Availability Status	Sample Text
Languages Spoken	Sample Text
Assigned Region	Sample Text
Profile Picture	Sample Text
Owner	Sample Text

Address Details

Country	Sample Text
City	Sample Text
Address	Sample Text

Figure 12. Page Layouts



Page > Cancel Confirmation

Object: Booking

API Name: Cancel_Confirmation__c

UI Behavior: [✓ None] Read-Only Required

Conditional Formatting: Select a ruleset...

View All Rulesets

Set Field Visibility

Filters

1 Record > Booking Status Equal Cancelled

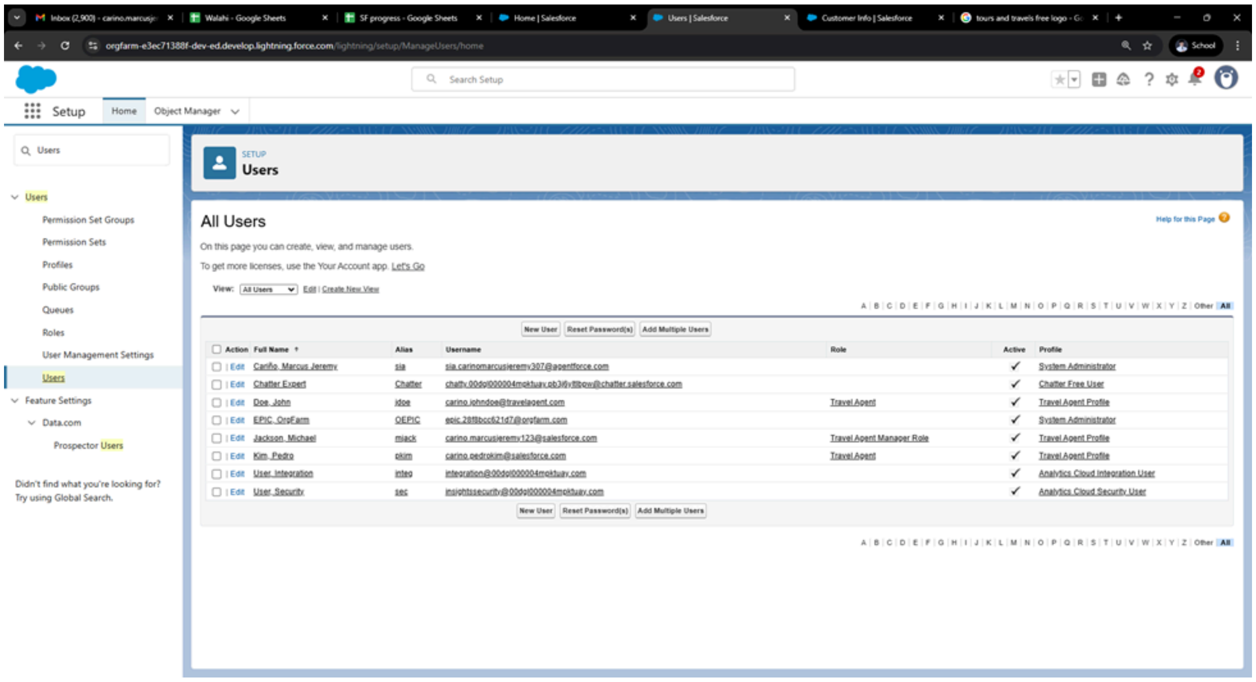
Add Filter

Figure 13. Dynamic Forms

User Management

Multiple user accounts were created and assigned appropriate roles and custom profiles established during Phase 2. Users included representatives for the following job functions:

- Travel Agent
- Travel Agent Manager
- Tour Guide
- Finance Officer
- Marketing Executive



All Users

On this page you can create, view, and manage users.

To get more licenses, use the Your Account app. Let's Go

Action	Full Name	Alias	Username	Role	Active	Profile
Edit	Cariño, Marcus Jeremy	sia	sia.carinomarcusjeremy307@agentforce.com		✓	System Administrator
Edit	Chatter Expert	Chatter	chatty.00dgl000004mpktuay.pb3i6yf8bow@chatter.salesforce.com		✓	Chatter Free User
Edit	Doe, John	jdoe	carino.johndoe@travelagent.com	Travel Agent	✓	Travel Agent Profile
Edit	EPIC, OrgFarm	OEPIC	epic.28f8bcc621d7@orgfarm.com		✓	System Administrator
Edit	Jackson, Michael	mjack	carino.marcusjeremy123@salesforce.com	Travel Agent Manager Role	✓	Travel Agent Profile
Edit	Kim, Pedro	pkim	carino.pedrokim@salesforce.com	Travel Agent	✓	Travel Agent Profile
Edit	User, Integration	integ	integration@00dgl000004mpktuay.com		✓	Analytics Cloud Integration User
Edit	User, Security	sec	insightssecurity@00dgl000004mpktuay.com		✓	Analytics Cloud Security User

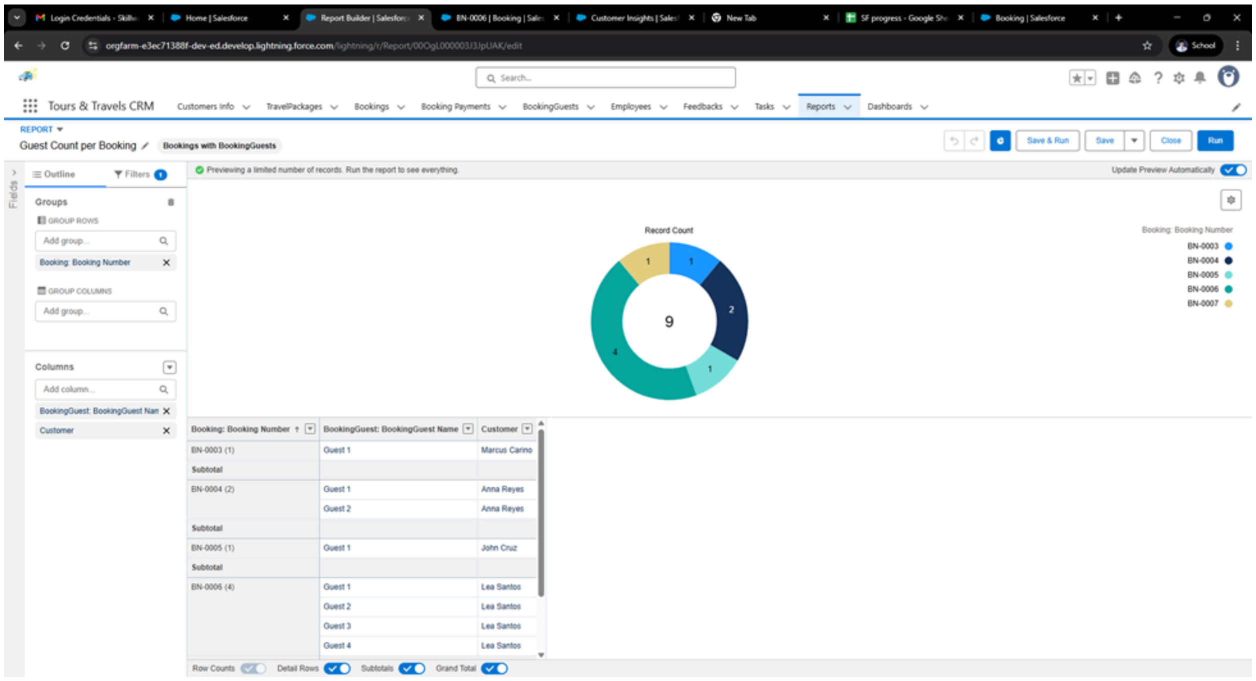
Figure 14. Users

Due to system constraints, only three active test users could be created. Each account was verified for login functionality, object access, and role-based record visibility.

Reports

Custom reports were developed to support operational monitoring and business decision-making. The reports leveraged standard and custom fields across objects, and were filtered and grouped according to business use cases. Some examples include:

- **Monthly Revenue Report**: Grouped by revenue range to classify low, medium, and high-earning bookings.
- **Pending Payments Report**: Shows bookings with unpaid status for follow-up.
- **Top Travel Packages Report**: Displays packages with the highest number of bookings.
- **Employee Role Distribution**: Visualizes the count of employees by assigned roles.



Booking: Booking Number	BookingGuest: BookingGuest Name	Customer
BN-0003 (1)	Guest 1	Marcus Carino
Subtotal		
BN-0004 (2)	Guest 1	Anna Reyes
	Guest 2	Anna Reyes
Subtotal		
BN-0005 (1)	Guest 1	John Cruz
Subtotal		
BN-0006 (4)	Guest 1	Lea Santos
	Guest 2	Lea Santos
	Guest 3	Lea Santos
	Guest 4	Lea Santos

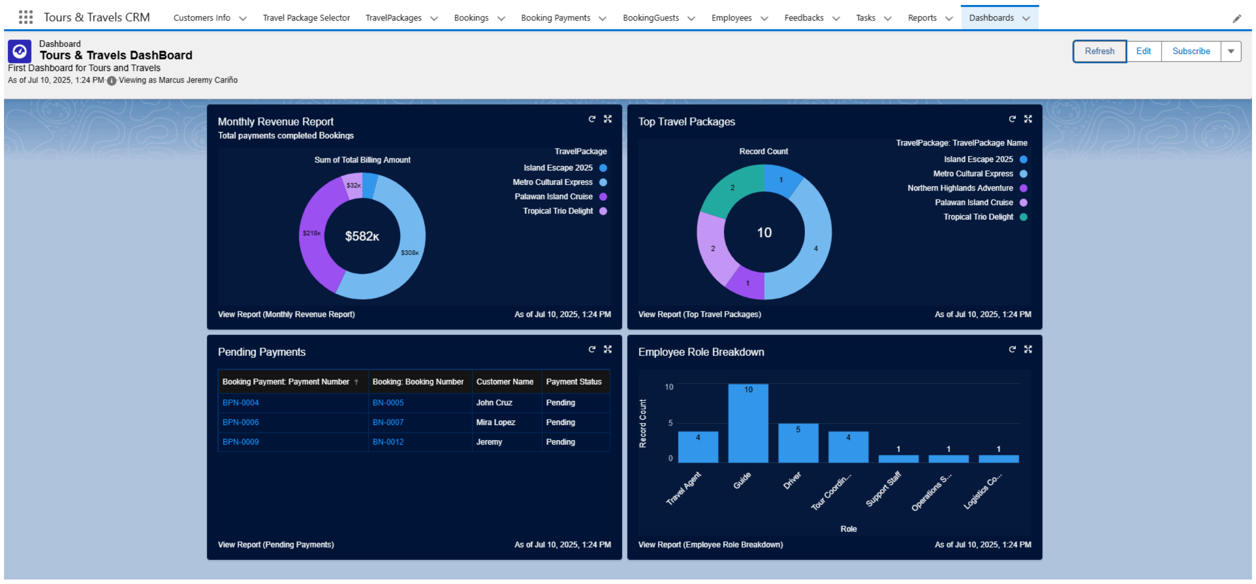
Figure 15. Sample Reports (Guest Count per Booking)

Each report was tested to confirm correct field inclusion, data grouping, and aggregation.

Dashboards

Dashboards were created to provide graphical representations of key performance indicators. The **"Tours & Travels Dashboard"** was configured using data from the above reports. Visual components include:

- Pie charts (e.g., for revenue segmentation)
- Bar charts (e.g., for package popularity)
- Metrics summaries (e.g., count of completed bookings)



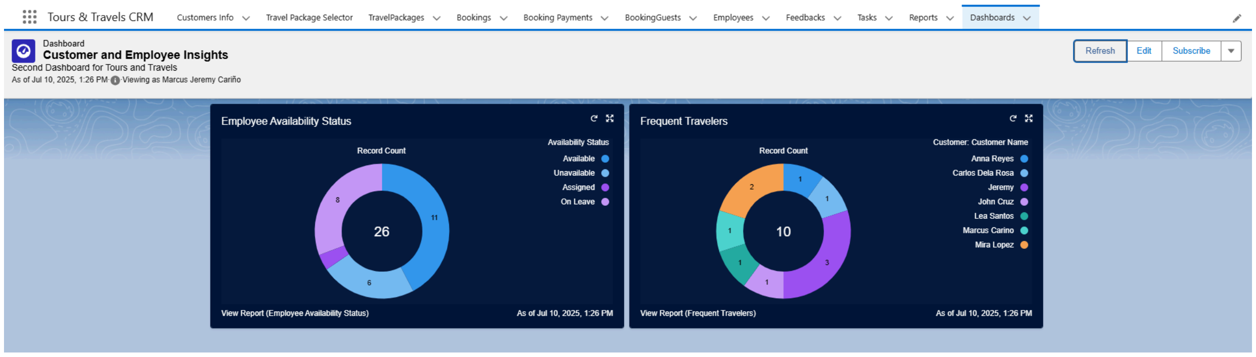
Dashboard
Tours & Travels DashBoard
First Dashboard for Tours and Travels
As of Jul 10, 2025, 1:24 PM Viewing as Marcus Jeremy Cariño

Pending Payments

Booking Payment: Payment Number ↑	Booking: Booking Number	Customer Name	Payment Status
BPN-0004	BN-0005	John Cruz	Pending
BPN-0006	BN-0007	Mira Lopez	Pending
BPN-0009	BN-0012	Jeremy	Pending

View Report (Monthly Revenue Report) As of Jul 10, 2025, 1:24 PM
View Report (Top Travel Packages) As of Jul 10, 2025, 1:24 PM
View Report (Pending Payments) As of Jul 10, 2025, 1:24 PM
View Report (Employee Role Breakdown) As of Jul 10, 2025, 1:24 PM

Figure 16. Sample Dashboard (Tours & Travels DashBoard)



Dashboard
Customer and Employee Insights
Second Dashboard for Tours and Travels
As of Jul 10, 2025, 1:26 PM Viewing as Marcus Jeremy Cariño

View Report (Employee Availability Status) As of Jul 10, 2025, 1:26 PM
View Report (Frequent Travelers) As of Jul 10, 2025, 1:26 PM

Figure 17. Sample Dashboard (Customer and Employee Insights)

These dashboards were embedded into the Lightning App and set to auto-refresh daily.

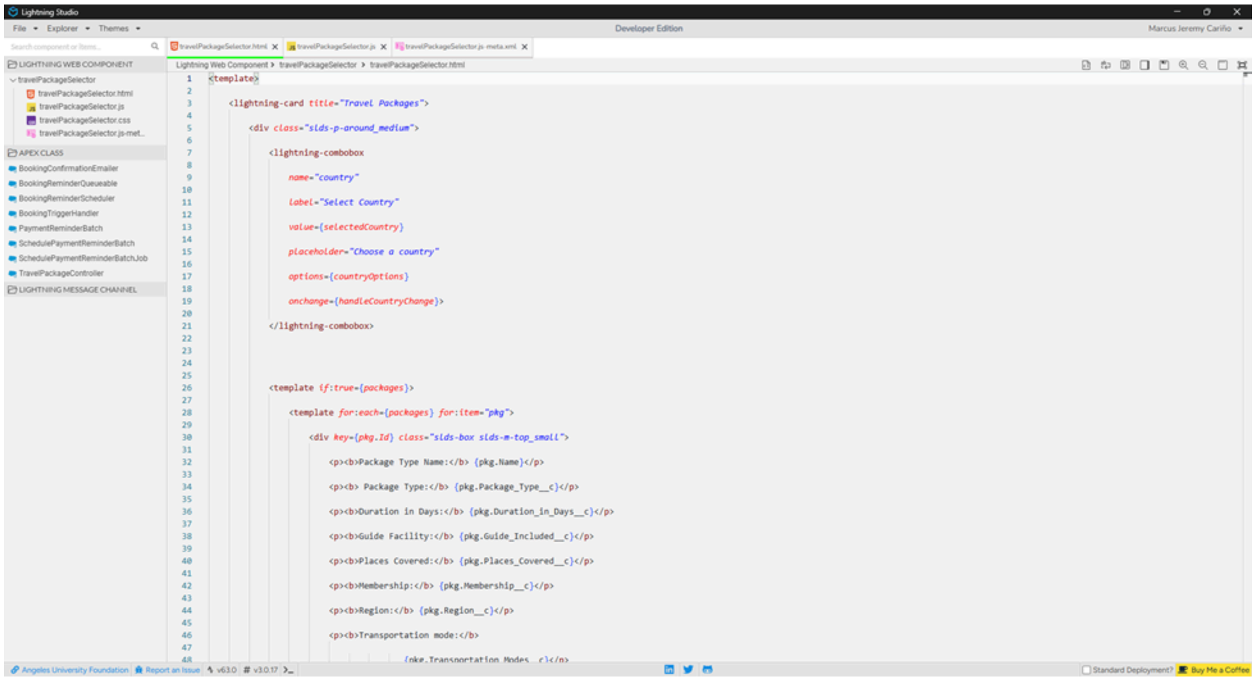
Lightning Web Component (LWC)

An interactive **Lightning Web Component** named `travelPackageSelector` was developed to display available travel packages filtered by country. This feature enhances usability by enabling dynamic, real-time package filtering without page reloads.

The component was powered by a custom Apex controller (`TravelPackageController`) and was deployed on the App Home Page, the Travel Package record page, and the Lightning App Page. This reusable component improves package browsing efficiency for agents and customers.

Example Functionalities:

- Drop-down filter to select country
- Automatic update of visible packages based on filter
- Display of key package details including destination, pricing, and availability


Lightning Studio
File
Explorer
Themes
Developer Edition
Marcus Jeremy Cariño
travelPackageSelector.html
travelPackageSelector.js
travelPackageSelector.js-meta.xml
LIGHTNING WEB COMPONENT
travelPackageSelector
travelPackageSelector.html
travelPackageSelector.js
travelPackageSelector.css
travelPackageSelector.js-met...
APEX CLASS
BookingConfirmationEmailer
BookingReminderQueueable
BookingReminderScheduler
BookingTriggerHandler
PaymentReminderBatch
SchedulePaymentReminderBatch
SchedulePaymentReminderBatchJob
TravelPackageController
LIGHTNING MESSAGE CHANNEL
Lightning Web Component › travelPackageSelector › travelPackageSelector.html
Angeles University Foundation
Report an Issue
v63.0
v3.0.17
Standard Deployment?
Buy Me a Coffee

```html
<template>

    <lightning-card title="Travel Packages">

        <div class="slds-p-around_medium">

            <lightning-combobox

                name="country"

                label="Select Country"

                value={selectedCountry}

                placeholder="Choose a country"

                options={countryOptions}

                onchange={handleCountryChange}>

            </lightning-combobox>



            <template if:true={packages}>

                <template for:each={packages} for:item="pkg">

                    <div key={pkg.Id} class="slds-box slds-m-top_small">

                        <p><b>Package Type Name:</b> {pkg.Name}</p>

                        <p><b> Package Type:</b> {pkg.Package_Type__c}</p>

                        <p><b>Duration in Days:</b> {pkg.Duration_in_Days__c}</p>

                        <p><b>Guide Facility:</b> {pkg.Guide_Included__c}</p>

                        <p><b>Places Covered:</b> {pkg.Places_Covered__c}</p>

                        <p><b>Membership:</b> {pkg.Membership__c}</p>

                        <p><b>Region:</b> {pkg.Region__c}</p>

                        <p><b>Transportation mode:</b>

                                    {pkg.Transportation_Modes__c}</p>
```

Figure 18. Lightning Web Component UI

Lightning App Page

A dedicated Lightning App Page titled **"Travel Package Selector"** was created using the Lightning App Builder. This page serves as a standalone interface to host the LWC and was included in the CRM's navigation menu. The app page was activated and set to be accessible by all user roles that interact with travel packages.

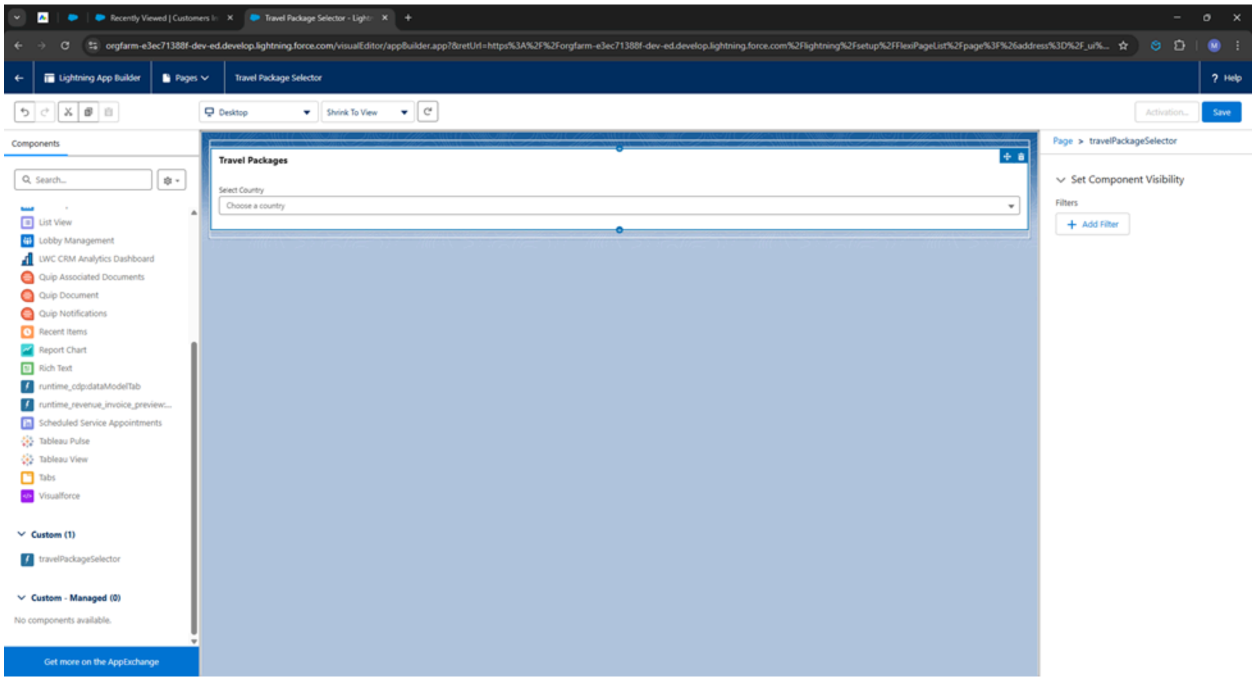
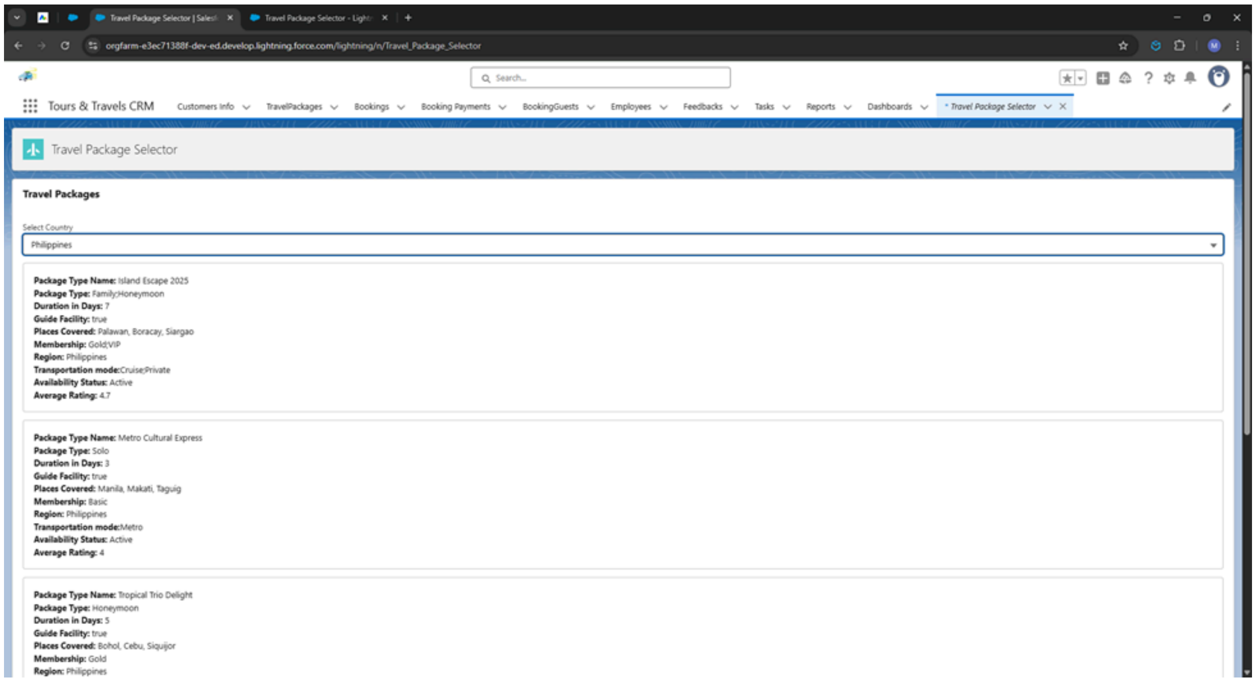


Figure 19. Lightning App Builder UI



Travel Packages

Select Country: Philippines

Package Type Name: Island Escape 2025
Package Type: Family;Honeymoon
Duration in Days: 7
Guide Facility: true
Places Covered: Palawan, Boracay, Siargao
Membership: Gold;VIP
Region: Philippines
Transportation mode: Cruise;Private
Availability Status: Active
Average Rating: 4.7

Package Type Name: Metro Cultural Express
Package Type: Solo
Duration in Days: 3
Guide Facility: true
Places Covered: Manila, Makati, Taguig
Membership: Basic
Region: Philippines
Transportation mode: Metro
Availability Status: Active
Average Rating: 4

Package Type Name: Tropical Trio Delight
Package Type: Honeymoon
Duration in Days: 5
Guide Facility: true
Places Covered: Bohol, Cebu, Siquijor
Membership: Gold
Region: Philippines

Figure 20. Lightning App Travel Packages

Phase 4: Data Migration, Testing & Security

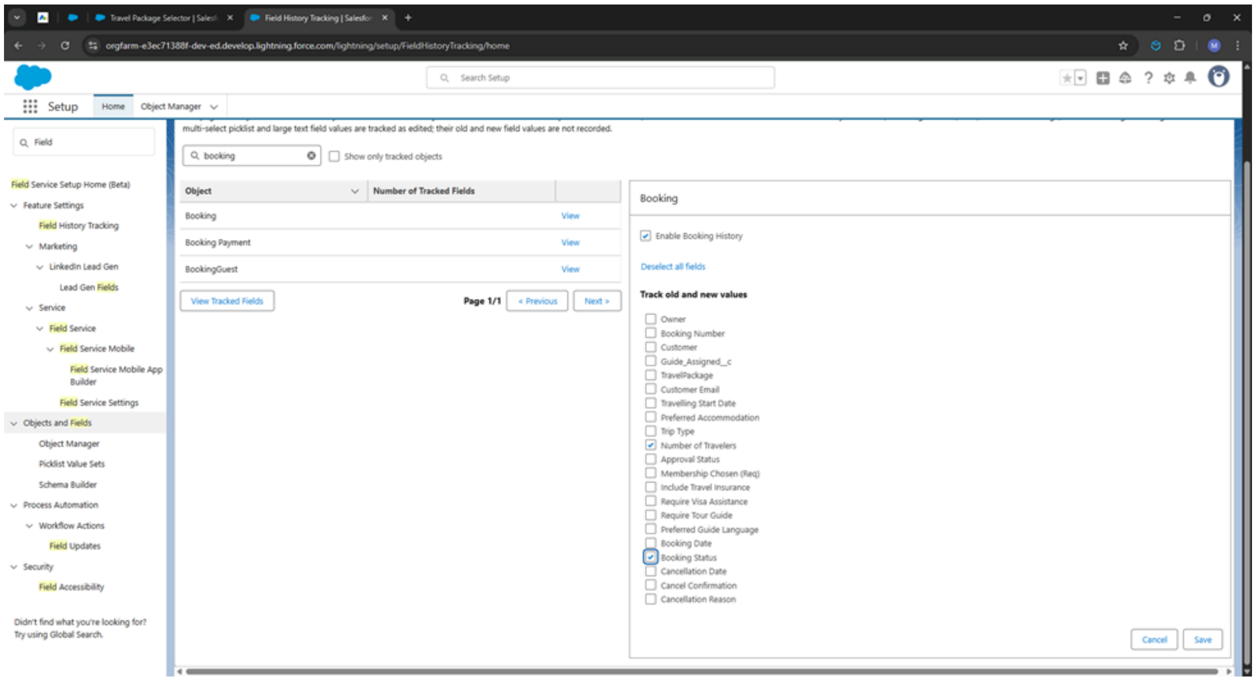
This phase focused on ensuring the reliability, consistency, and security of the Tours & Travels CRM system. It involved the configuration of field tracking mechanisms, enforcement of data integrity rules, secure user access management, creation of test classes, and preparation of sample datasets for migration. The outcome of this phase was a secure and audit-ready CRM system, capable of supporting real-world travel operations with reliable data handling.

Field History Tracking

Field History Tracking was enabled to monitor changes made to critical fields. This ensured transparency, supported future audits, and helped in understanding record modifications over time.

Tracked Fields Included:

- **Booking__c**: `Booking_Status__c`, `Number_of_Travelers__c`, `TravelPackage__c`
- **TravelPackage__c**: `Price_Per_Person__c`, `Availability_Status__c`


multi-select picklist and large text field values are tracked as edited; their old and new field values are not recorded.

Object	Number of Tracked Fields	
Booking		View
Booking Payment		View
BookingGuest		View

Booking

- [✓] Enable Booking History

Track old and new values

- [ ] Owner
- [ ] Booking Number
- [ ] Customer
- [ ] Guide_Assigned__c
- [ ] TravelPackage
- [ ] Customer Email
- [ ] Travelling Start Date
- [ ] Preferred Accommodation
- [ ] Trip Type
- [✓] Number of Travelers
- [ ] Approval Status
- [ ] Membership Chosen (Req)
- [ ] Include Travel Insurance
- [ ] Require Visa Assistance
- [ ] Require Tour Guide
- [ ] Preferred Guide Language
- [ ] Booking Date
- [✓] Booking Status
- [ ] Cancellation Date
- [ ] Cancel Confirmation
- [ ] Cancellation Reason

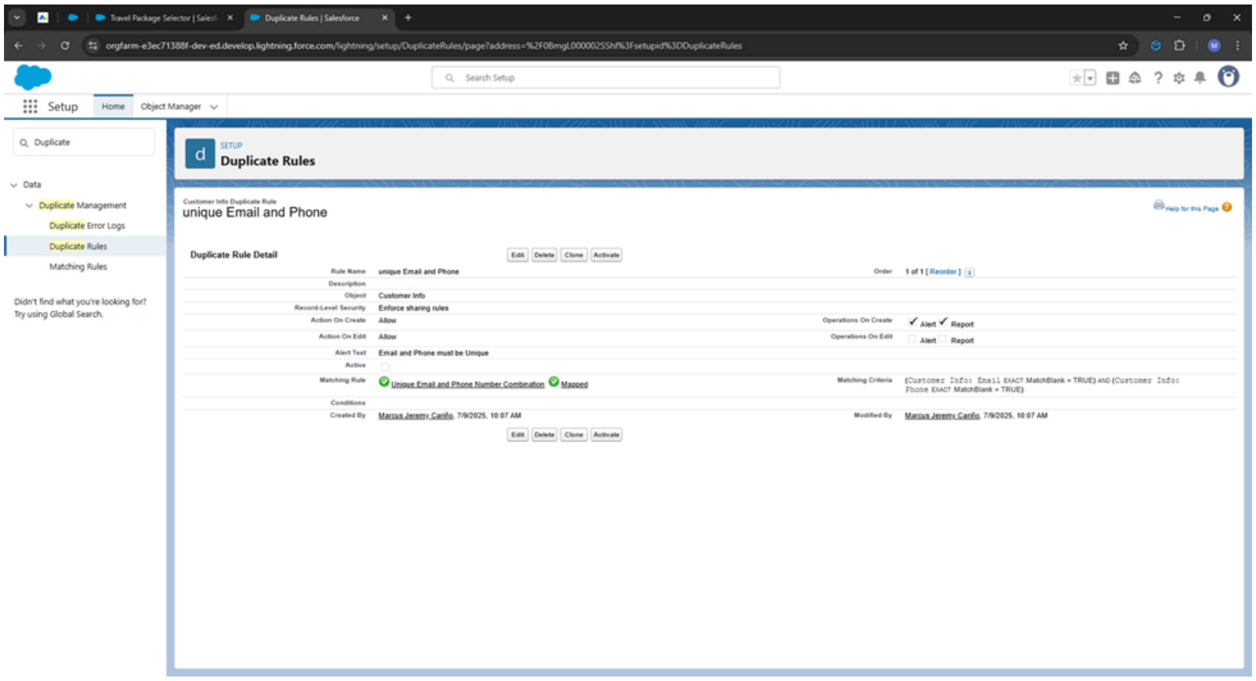
Figure 21. Field History Tracking

This configuration supports both operational tracking and post-travel data analysis.

Duplicate & Matching Rules

To prevent redundant or inconsistent data, Duplicate and Matching Rules were configured for the **Customer_Info__c** object.

- A **Matching Rule** was created to compare records using an exact match on `Email` and `Phone`.
- A **Duplicate Rule** was implemented to display a warning message to users when a duplicate record is detected, without blocking record creation (for manual review cases).


Customer Info Duplicate Rule

unique Email and Phone

Duplicate Rule Detail

Rule Name	unique Email and Phone	Order	1 of 1 [Reorder]
Description			
Object	Customer Info		
Record-Level Security	Enforce sharing rules		
Action On Create	Allow	Operations On Create	✓ Alert ✓ Report
Action On Edit	Allow	Operations On Edit	Alert Report
Alert Text	Email and Phone must be Unique		
Active			
Matching Rule	Unique Email and Phone Number Combination Mapped	Matching Criteria	(Customer Info: Email EXACT MatchBlank = TRUE) AND (Customer Info: Phone EXACT MatchBlank = TRUE)
Conditions			
Created By	Marcus Jeremy Cariño, 7/9/2025, 10:07 AM	Modified By	Marcus Jeremy Cariño, 7/9/2025, 10:07 AM

Figure 22. Duplicate & Matching Rules

This approach maintains data accuracy while allowing flexibility in user data entry.

Profiles

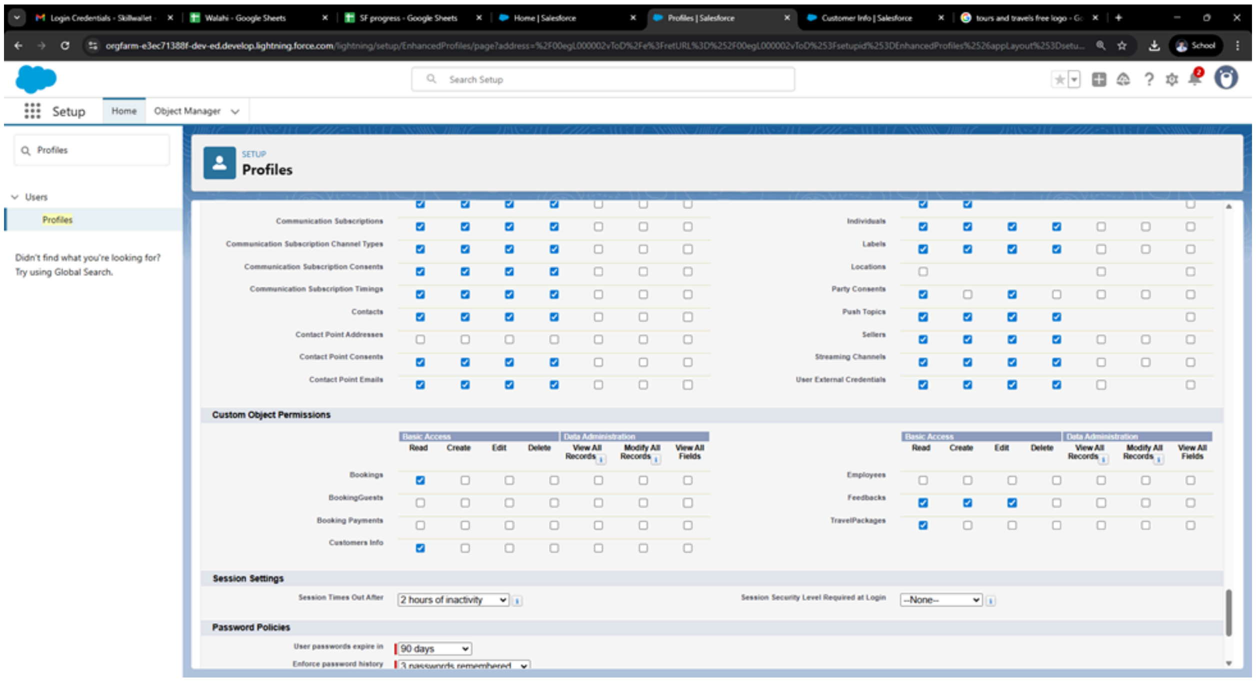
Custom profiles were created to define the object-level and field-level permissions based on the employee’s role within the organization. This ensures that users have access only to the data necessary for their job function.

Profiles Configured:

- **Travel Agent**: Full access to Booking and Customer Info objects.
- **Tour Guide**: Read-only access to assigned bookings and package details.
- **Finance Officer**: Access to Booking and Payment data.
- **Marketing Executive**: Access to Travel Packages and Customer Feedback.
- **Customer Service Rep**: Access to Feedback object only.

Standard Salesforce security policies were also applied:

- Session timeout set to 2 hours
- Password policy enforced (minimum 8 characters, no expiration)



SETUP

Profiles

Communication Subscriptions	✓	✓	✓	✓	☐	☐	☐
Communication Subscription Channel Types	✓	✓	✓	✓	☐	☐	☐
Communication Subscription Consents	✓	✓	✓	✓	☐	☐	☐
Communication Subscription Timings	✓	✓	✓	✓	☐	☐	☐
Contacts	✓	✓	✓	✓	☐	☐	☐
Contact Point Addresses	☐	☐	☐	☐	☐	☐	☐
Contact Point Consents	✓	✓	✓	✓	☐	☐	☐
Contact Point Emails	✓	✓	✓	✓	☐	☐	☐
Individuals	✓	✓	✓	✓	☐	☐	☐
Labels	✓	✓	✓	✓	☐	☐	☐
Locations	☐				☐		☐
Party Consents	✓	☐	✓	☐	☐	☐	☐
Push Topics	✓	✓	✓	✓			☐
Sellers	✓	✓	✓	✓	☐	☐	☐
Streaming Channels	✓	✓	✓	✓	☐	☐	☐
User External Credentials	✓	✓	✓	✓	☐		☐

Custom Object Permissions

	Basic Access				Data Administration		
	Read	Create	Edit	Delete	View All Records	Modify All Records	View All Fields
Bookings	✓	☐	☐	☐	☐	☐	☐
BookingGuests	☐	☐	☐	☐	☐	☐	☐
Booking Payments	☐	☐	☐	☐	☐	☐	☐
Customers Info	✓	☐	☐	☐	☐	☐	☐
Employees	☐	☐	☐	☐	☐	☐	☐
Feedbacks	✓	✓	✓	☐	☐	☐	☐
TravelPackages	✓	☐	☐	☐	☐	☐	☐

Session Settings

Session Times Out After: 2 hours of inactivity

Session Security Level Required at Login: --None--

Password Policies

User passwords expire in: 90 days

Enforce password history: 3 passwords remembered

Figure 23. Profiles

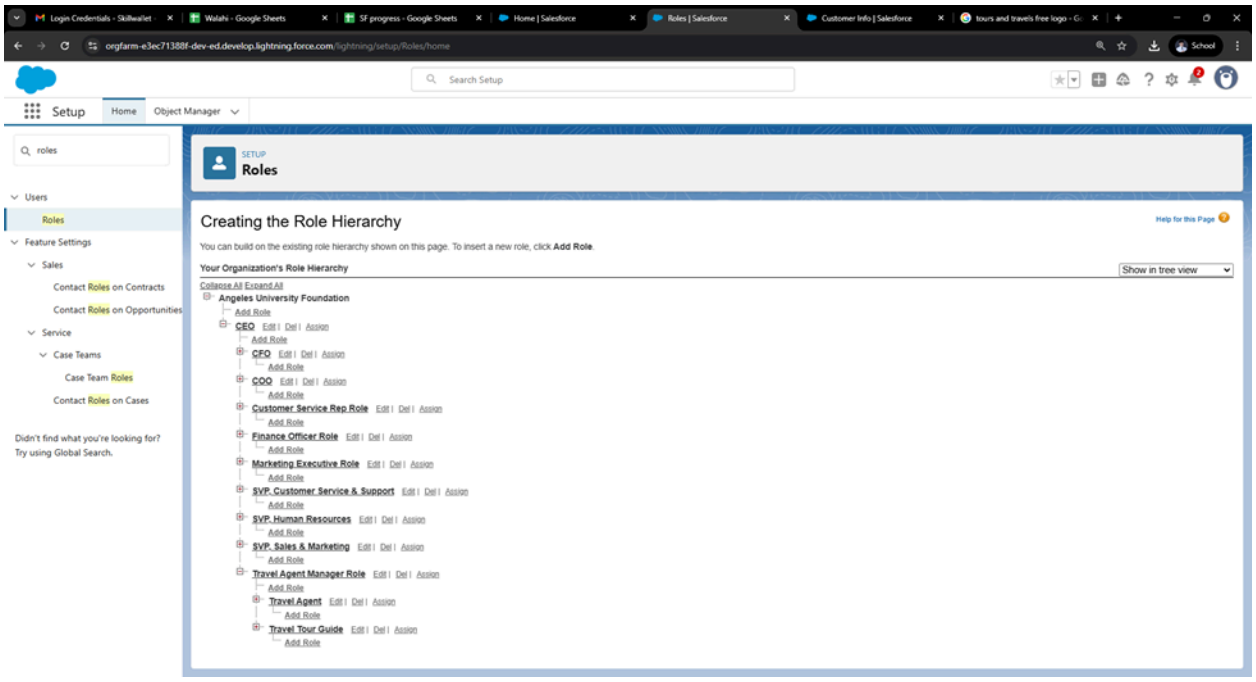
Role Hierarchy

A role hierarchy was implemented to control record visibility based on organizational structure. This allows higher roles to view records owned by subordinate roles while preserving confidentiality for unrelated users.

Hierarchy Structure:

- **Chief Executive Officer**
    - Travel Agent Manager

        - Travel Agent
        - Tour Guide
    - Finance Officer
    - Marketing Executive
    - Customer Service Rep



Didn't find what you're looking for?
Try using Global Search.

Creating the Role Hierarchy

You can build on the existing role hierarchy shown on this page. To insert a new role, click **Add Role**.

Your Organization's Role Hierarchy

```
Collapse All Expand All
Angeles University Foundation
  Add Role
  CEO  Edit | Del | Assign
    Add Role
    CFO  Edit | Del | Assign
      Add Role
    COO  Edit | Del | Assign
      Add Role
    Customer Service Rep Role  Edit | Del | Assign
      Add Role
    Finance Officer Role  Edit | Del | Assign
      Add Role
    Marketing Executive Role  Edit | Del | Assign
      Add Role
    SVP, Customer Service & Support  Edit | Del | Assign
      Add Role
    SVP, Human Resources  Edit | Del | Assign
      Add Role
    SVP, Sales & Marketing  Edit | Del | Assign
      Add Role
    Travel Agent Manager Role  Edit | Del | Assign
      Add Role
      Travel Agent  Edit | Del | Assign
        Add Role
      Travel Tour Guide  Edit | Del | Assign
        Add Role
```

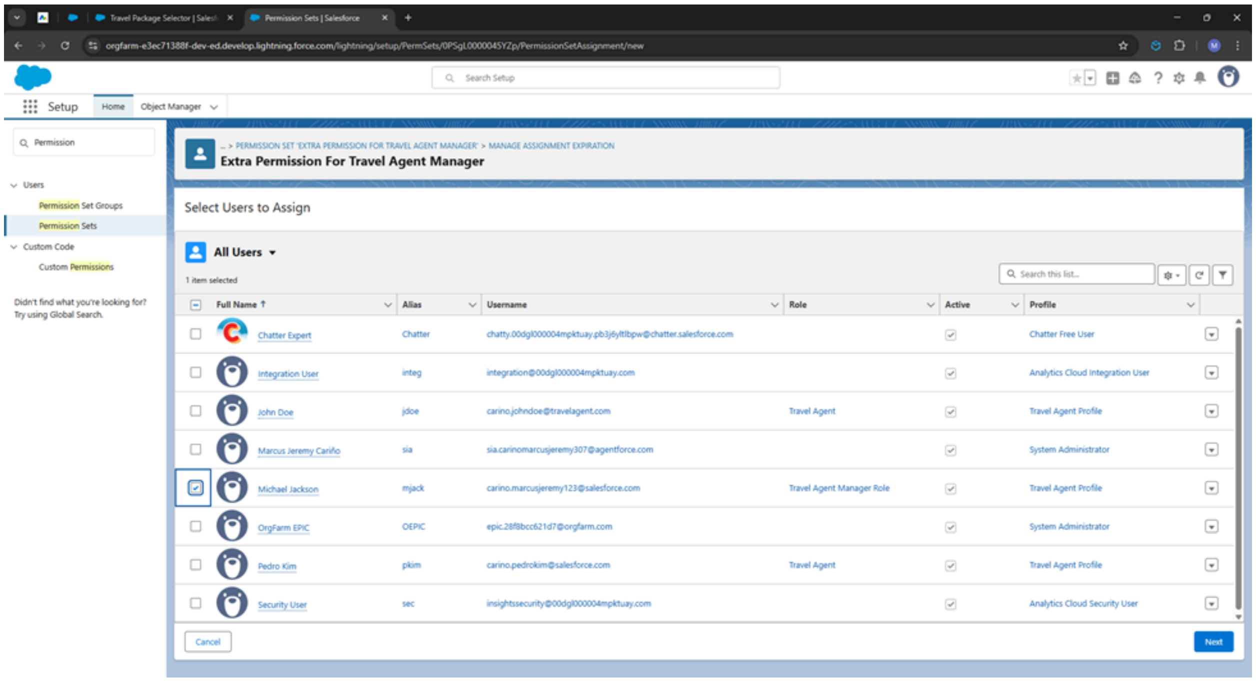
Figure 24. Role Hierarchy

This hierarchy enabled seamless delegation and oversight while respecting role boundaries.

Permission Sets

To grant additional access beyond the base profile, a Permission Set titled **"Extended Travel Package Access"** was created. This was assigned to selected Travel Agent Managers and enabled full CRUD access to the **TravelPackage__c** object, without affecting other users with the same profile.

Permission Sets were used to allow granular permission allocation based on operational needs.


... > PERMISSION SET 'EXTRA PERMISSION FOR TRAVEL AGENT MANAGER' > MANAGE ASSIGNMENT EXPIRATION

Extra Permission For Travel Agent Manager

Select Users to Assign

All Users

1 item selected

	Full Name	Alias	Username	Role	Active	Profile
☐	Chatter Expert	Chatter	chatty.00dgl000004mpktuay.pb3j6yltlbpw@chatter.salesforce.com		✓	Chatter Free User
☐	Integration User	integ	integration@00dgl000004mpktuay.com		✓	Analytics Cloud Integration User
☐	John Doe	jdoe	carino.johndoe@travelagent.com	Travel Agent	✓	Travel Agent Profile
☐	Marcus Jeremy Cariño	sia	sia.carinomarcusjeremy307@agentforce.com		✓	System Administrator
✓	Michael Jackson	mjack	carino.marcusjeremy123@salesforce.com	Travel Agent Manager Role	✓	Travel Agent Profile
☐	OrgFarm EPIC	OEPIC	epic.28f8bcc621d7@orgfarm.com		✓	System Administrator
☐	Pedro Kim	pkim	carino.pedrokim@salesforce.com	Travel Agent	✓	Travel Agent Profile
☐	Security User	sec	insightssecurity@00dgl000004mpktuay.com		✓	Analytics Cloud Security User

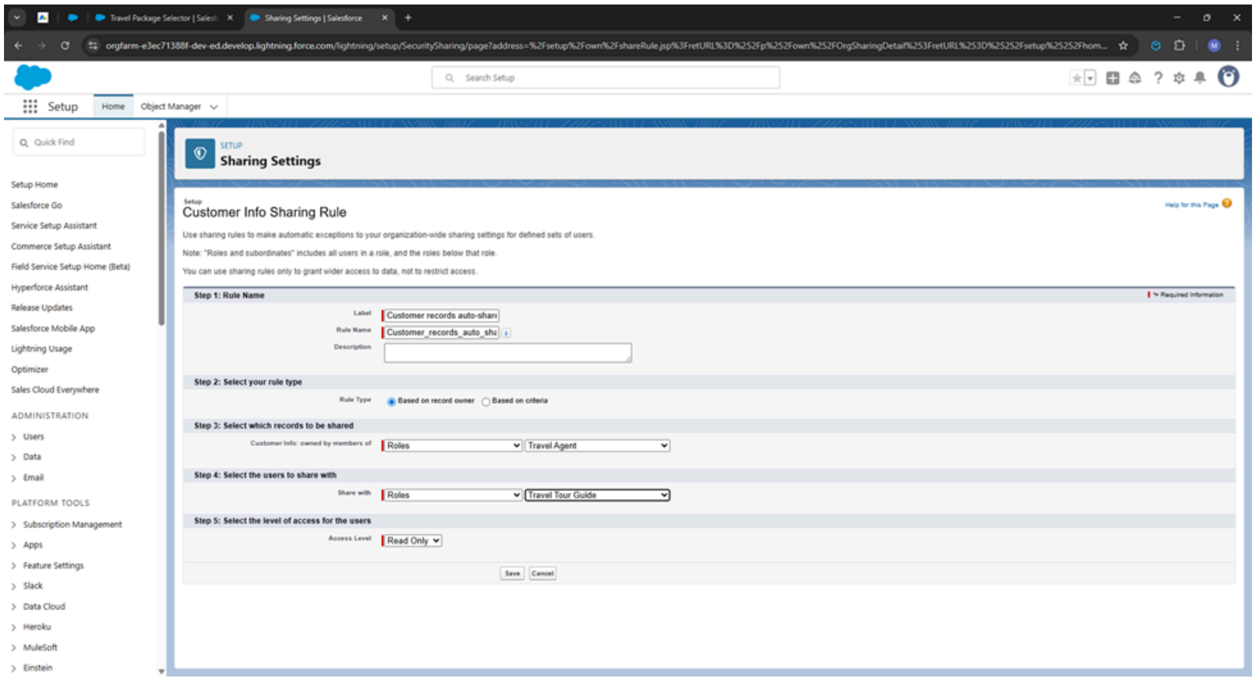
Figure 25. Permission Sets

Sharing Settings

The **Organization-Wide Default (OWD)** settings were reviewed and adjusted to align with the access model. The `Customer_Info__c` object was set to **Private** to restrict unauthorized data visibility.

A custom **Sharing Rule** was implemented to:

- Automatically grant read-only access of a Customer record to the assigned **Tour Guide**, when linked via a Booking.



Setup

Customer Info Sharing Rule

Use sharing rules to make automatic exceptions to your organization-wide sharing settings for defined sets of users.

Note: "Roles and subordinates" includes all users in a role, and the roles below that role.

You can use sharing rules only to grant wider access to data, not to restrict access.

Step 1: Rule Name — = Required Information

Label	Customer records auto-shar
Rule Name	Customer_records_auto_sha
Description	

Step 2: Select your rule type

Rule Type: [✓ Based on record owner] Based on criteria

Step 3: Select which records to be shared

Customer Info: owned by members of: Roles — Travel Agent

Step 4: Select the users to share with

Share with: Roles — Travel Tour Guide

Step 5: Select the level of access for the users

Access Level: Read Only

Figure 25. Sharing Settings

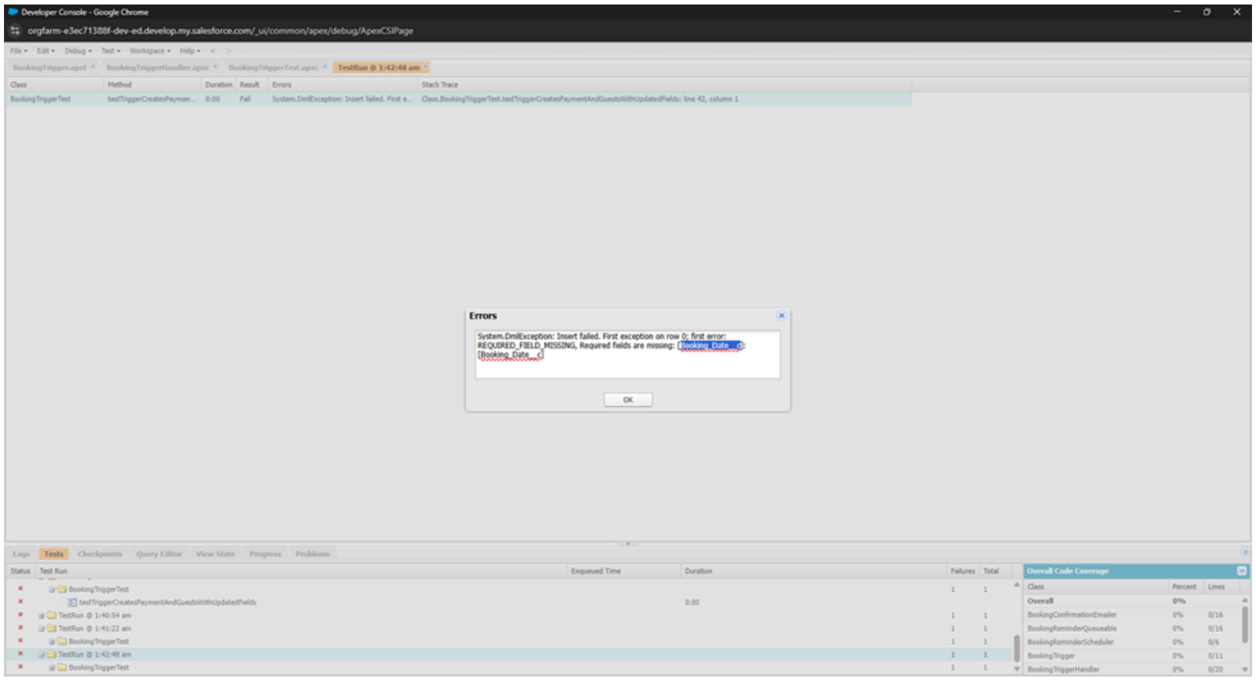
This ensured secure collaboration between agents and guides without compromising customer privacy.

Apex Test Classes

Custom Apex code written in earlier phases was validated using test classes to ensure functional correctness and code coverage.

Test Class Developed:

- **BookingTriggerTest**
  - Verified auto-creation of `Booking_Payment__c` and `BookingGuest__c` records
  - Checked trigger logic for Booking status updates and future email dispatch
  - Ensured validation rules were enforced as expected



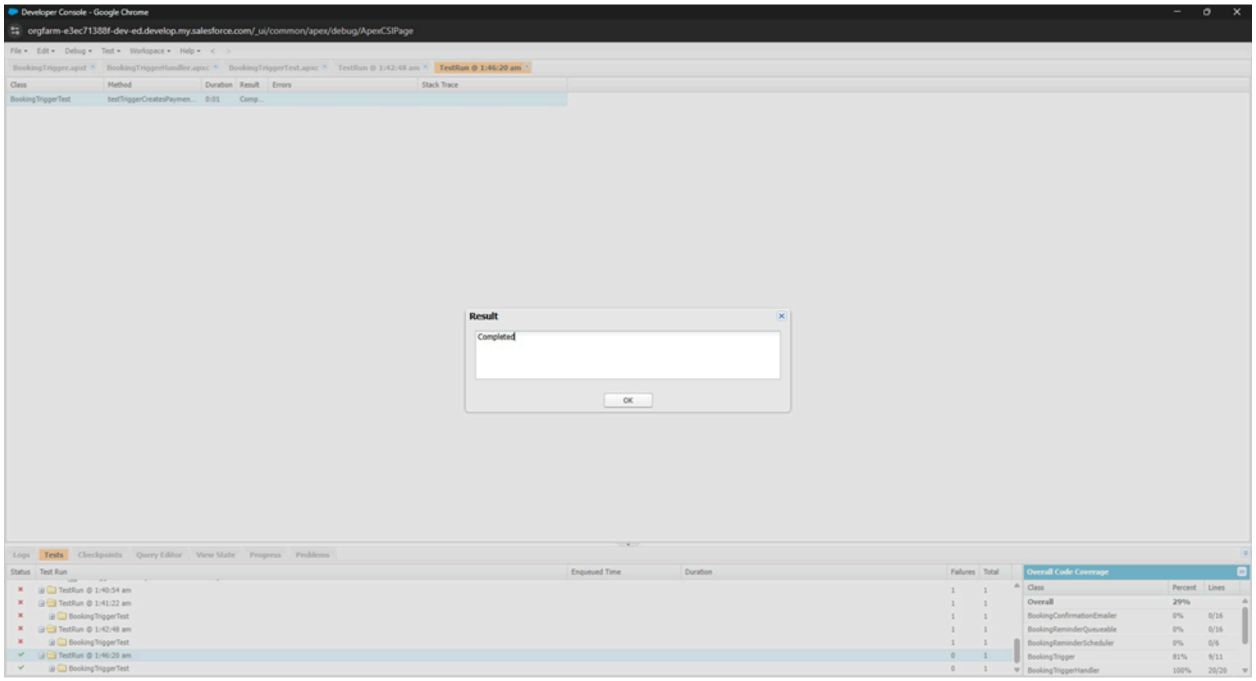
Class	Method	Duration	Result	Errors	Stack Trace
BookingTriggerTest	testTriggerCreatesPaymen...	0:00	Fail	System.DmlException: Insert failed. First e...	Class.BookingTriggerTest.testTriggerCreatesPaymentAndGuestsWithUpdatedFields: line 42, column 1

```
System.DmlException: Insert failed. First exception on row 0; first error:
REQUIRED_FIELD_MISSING, Required fields are missing: [Booking_Date__c]:
[Booking_Date__c]
```

Status	Test Run	Enqueued Time	Duration	Failures	Total
✗	BookingTriggerTest			1	1
✗	testTriggerCreatesPaymentAndGuestsWithUpdatedFields		0:00		
✗	TestRun @ 1:40:54 am			1	1
✗	TestRun @ 1:41:22 am			1	1
✗	BookingTriggerTest			1	1
✗	TestRun @ 1:42:48 am			1	1
✗	BookingTriggerTest			1	1

Class	Percent	Lines
Overall	0%	
BookingConfirmationEmailer	0%	0/16
BookingReminderQueueable	0%	0/16
BookingReminderScheduler	0%	0/6
BookingTrigger	0%	0/11
BookingTriggerHandler	0%	0/20

Figure 26. Sample Issues During Debugging the Test Class



Class	Method	Duration	Result	Errors	Stack Trace
BookingTriggerTest	testTriggerCreatesPaymen...	0:01	Comp...		

```
Completed
```

Status	Test Run	Enqueued Time	Duration	Failures	Total
✗	TestRun @ 1:40:54 am			1	1
✗	TestRun @ 1:41:22 am			1	1
✗	BookingTriggerTest			1	1
✗	TestRun @ 1:42:48 am			1	1
✗	BookingTriggerTest			1	1
✓	TestRun @ 1:46:20 am			0	1
✓	BookingTriggerTest			0	1

Class	Percent	Lines
Overall	29%	
BookingConfirmationEmailer	0%	0/16
BookingReminderQueueable	0%	0/16
BookingReminderScheduler	0%	0/6
BookingTrigger	81%	9/11
BookingTriggerHandler	100%	20/20

Figure 27. Successful Run of Test Class

The class was executed with the needed code coverage, meeting Salesforce deployment requirements.

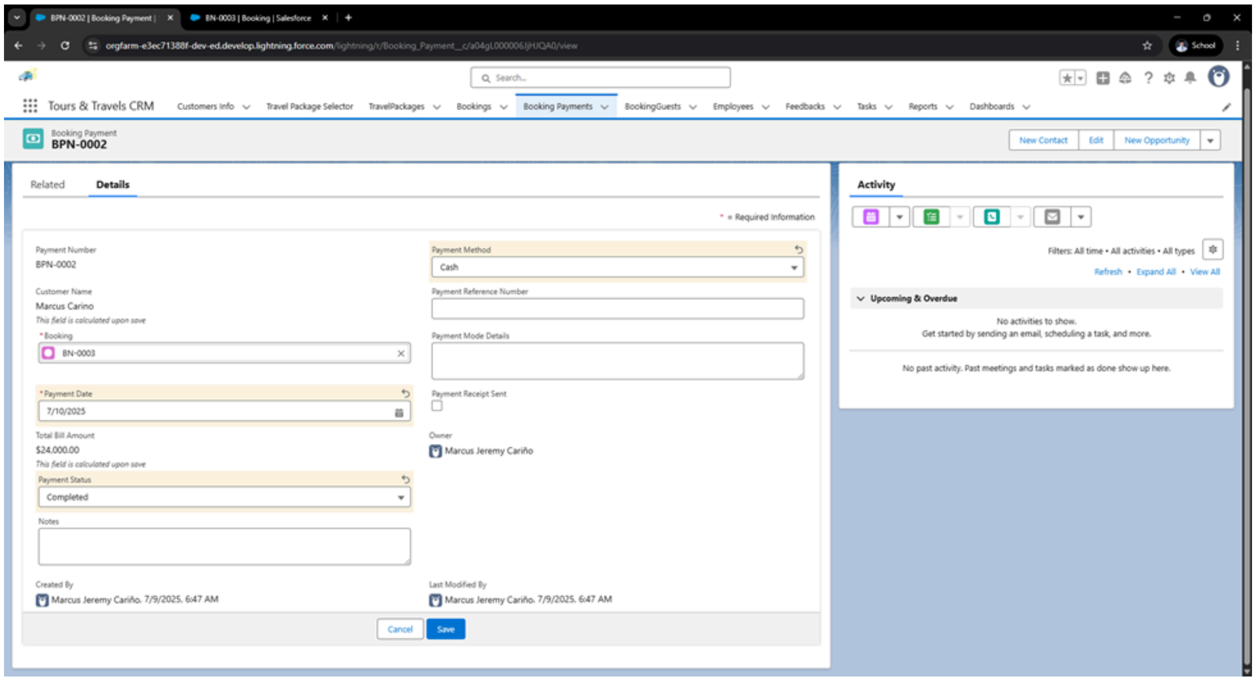
Test Cases and Defect Fixing

Manual testing was performed to validate the behavior of each automation and business logic component. For each feature, the following details were documented:

- **Scenario Description**
- **Input Values**
- **Expected Output**
- **Actual Output**
- **Pass/Fail Status**

Examples of test cases included:

- Creating a booking and verifying automatic creation of related payment and guest records
- Updating payment status to “Completed” and observing automatic Booking status change to “Confirmed”
- Triggering email notifications and feedback follow-up tasks
- Approval process test for booking cancellations
- Validation rules blocking incorrect data entries



Booking Payment
BPN-0002

* = Required Information

Payment Number	BPN-0002
Payment Method	Cash
Customer Name	Marcus Carino *This field is calculated upon save*
Payment Reference Number	
* Booking	BN-0003
Payment Mode Details	
* Payment Date	7/10/2025
Payment Receipt Sent	☐
Total Bill Amount	$24,000.00 *This field is calculated upon save*
Owner	Marcus Jeremy Cariño
Payment Status	Completed
Notes	
Created By	Marcus Jeremy Cariño, 7/9/2025, 6:47 AM
Last Modified By	Marcus Jeremy Cariño, 7/9/2025, 6:47 AM

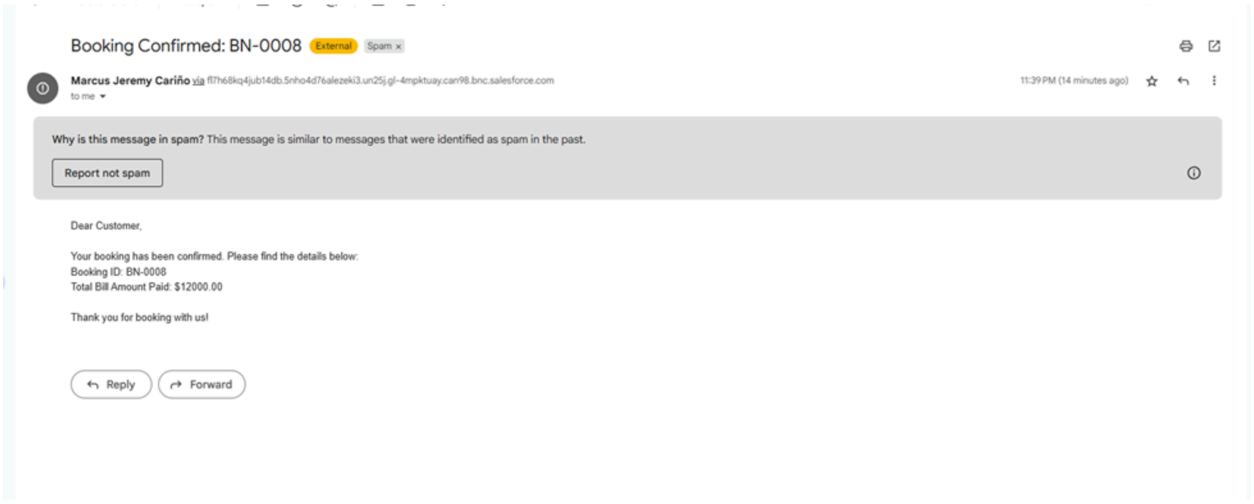
Filters: All time • All activities • All types

Upcoming & Overdue

No activities to show.
Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

Figure 28. Completed Booking Payment



Booking Confirmed: BN-0008

Marcus Jeremy Cariño via fl7h68kq4jub14db.5nho4d76alezeki3.un25j.gl-4mpktuay.can98.bnc.salesforce.com
to me

11:39 PM (14 minutes ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Dear Customer,

Your booking has been confirmed. Please find the details below:
Booking ID: BN-0008
Total Bill Amount Paid: $12000.00

Thank you for booking with us!

Figure 29. Email Confirmation of the Booking

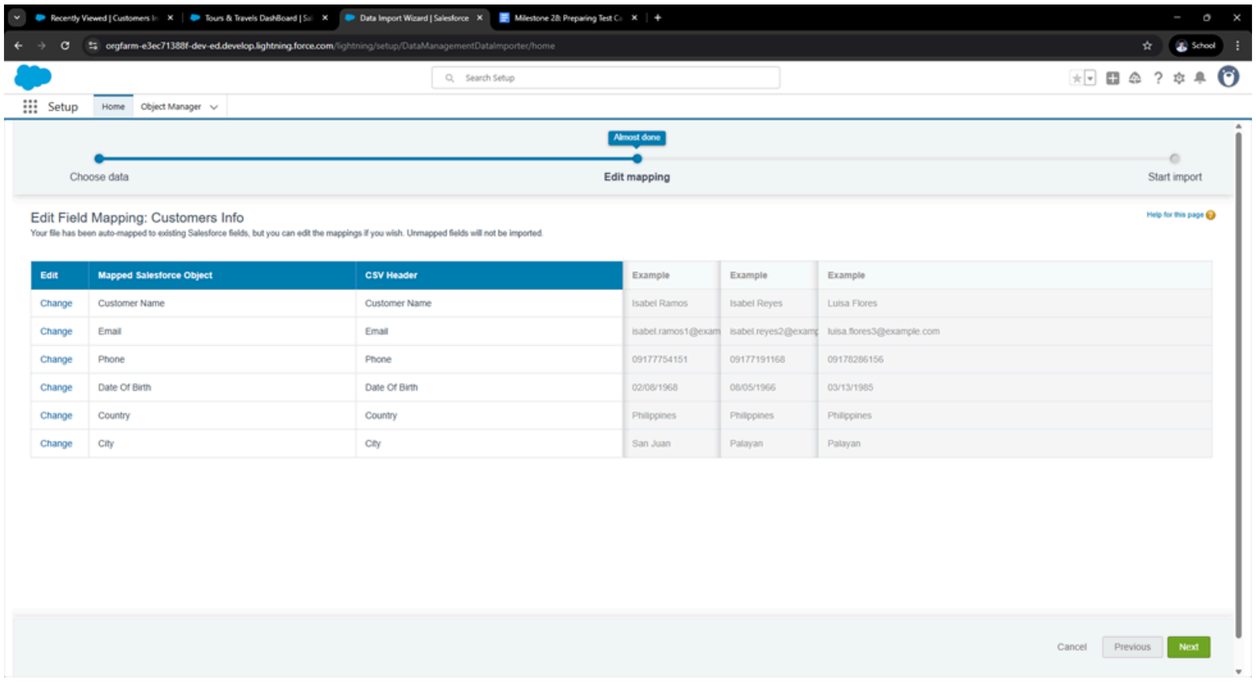
A separate file was prepared containing all test case results, including screenshots of both input and output. Milestone 28: Preparing Test Cases & Fixing Defects

Data Migration via Import Wizard

To simulate a real-world data import scenario, Salesforce's **Data Import Wizard** was used to load initial data sets.

Imported Objects and Sample Records:

- **Customer_Info__c** – 20 sample customers with valid emails and contact numbers
- **TravelPackage__c** – 20 travel packages with region and pricing details
- **Employee__c** – 20 employees with roles, departments, and cities



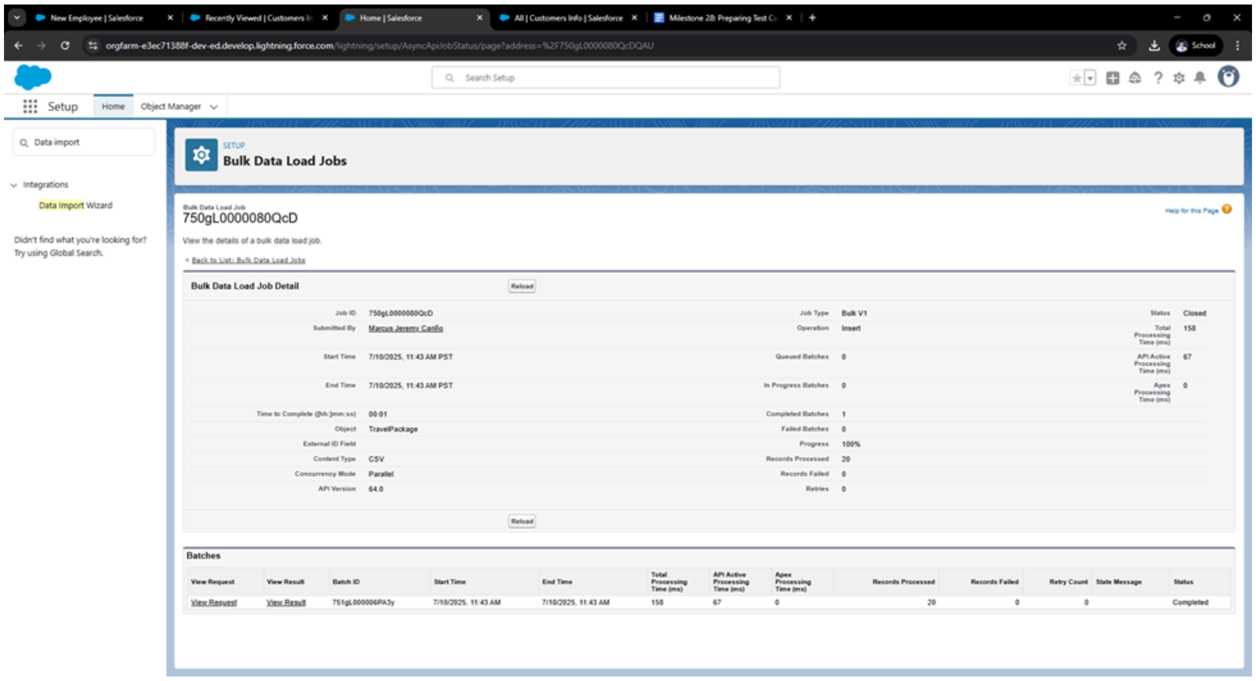
Recently Viewed | Customers In
Tours & Travels DashBoard | Sa
Data Import Wizard | Salesforce
Milestone 28: Preparing Test Ca
orgfarm-e3ec71388f-dev-ed.develop.lightning.force.com/lightning/setup/DataManagementDataImporter/home
School
Search Setup
Setup
Home
Object Manager
Almost done
Choose data
Edit mapping
Start import
Help for this page
Cancel
Previous
Next

Edit Field Mapping: Customers Info

Your file has been auto-mapped to existing Salesforce fields, but you can edit the mappings if you wish. Unmapped fields will not be imported.

Edit	Mapped Salesforce Object	CSV Header	Example	Example	Example
Change	Customer Name	Customer Name	Isabel Ramos	Isabel Reyes	Luisa Flores
Change	Email	Email	isabel.ramos1@exam	isabel.reyes2@examp	luisa.flores3@example.com
Change	Phone	Phone	09177754151	09177191168	09178286156
Change	Date Of Birth	Date Of Birth	02/08/1968	08/05/1966	03/13/1985
Change	Country	Country	Philippines	Philippines	Philippines
Change	City	City	San Juan	Palayan	Palayan

Figure 30. Data Migration via Import Wizard



Bulk Data Load Jobs

Bulk Data Load Job
750gL0000080QcD

View the details of a bulk data load job.

« Back to List: Bulk Data Load Jobs

Help for this Page

Bulk Data Load Job Detail

Job ID	750gL0000080QcD	Job Type	Bulk V1	Status	Closed
Submitted By	Marcus Jeremy Cariño	Operation	Insert	Total Processing Time (ms)	158
Start Time	7/10/2025, 11:43 AM PST	Queued Batches	0	API Active Processing Time (ms)	67
End Time	7/10/2025, 11:43 AM PST	In Progress Batches	0	Apex Processing Time (ms)	0
Time to Complete ([hh:]mm:ss)	00:01	Completed Batches	1		
Object	TravelPackage	Failed Batches	0		
External ID Field		Progress	100%		
Content Type	CSV	Records Processed	20		
Concurrency Mode	Parallel	Records Failed	0		
API Version	64.0	Retries	0		

Batches

View Request	View Result	Batch ID	Start Time	End Time	Total Processing Time (ms)	API Active Processing Time (ms)	Apex Processing Time (ms)	Records Processed	Records Failed	Retry Count	State Message	Status
View Request	View Result	751gL000006PA3y	7/10/2025, 11:43 AM	7/10/2025, 11:43 AM	158	67	0	20	0	0		Completed

Figure 30. Sample Bulk Data Load Jobs (Travel Packages)

All field mappings were manually reviewed. Post-import verification was performed by inspecting record counts, validating picklist values, and confirming data integrity.

Phase 5: Deployment, Documentation & Maintenance

Although actual deployment to a live business environment was not required for this academic project, this phase outlines the theoretical deployment approach, ongoing maintenance strategies, and documentation protocols that would be followed in a real-world production scenario. It also includes reflections on system reliability and suggestions for future enhancements.

Deployment Strategy

If the CRM were to be deployed in a business setting, Change Sets would be the recommended deployment method within the Salesforce ecosystem. Change Sets allow developers to migrate components (e.g., objects, fields, Apex classes, Flows, dashboards) from a Developer Sandbox to a Production Environment while preserving dependencies.

Deployment Process Overview:

1. **Outbound Change Sets** would be created in the Sandbox containing all custom objects, fields, triggers, flows, and other components.
2. **Inbound Change Sets** would be received in the Production org and validated before deployment.
3. **Test Classes** (developed in Phase 4) would be executed to confirm functional readiness and meet Salesforce's code coverage requirements.
4. Post-deployment data loading (e.g., travel packages and employee data) would be performed using **Data Import Wizard** or **Data Loader** depending on the record volume.

All components would be version-controlled and reviewed by stakeholders before migration to ensure alignment with business requirements.

Maintenance Strategy

To ensure long-term reliability and adaptability of the CRM, the following maintenance practices are proposed:

- **Monthly Performance Audits**: Reviewing automation execution times, record volumes, and system errors.
- **User Feedback Collection**: Using internal forms or surveys to gather feedback from users (agents, finance officers, etc.) regarding UI usability or automation responsiveness.
- **Quarterly Data Review**: Identifying and resolving data inconsistencies such as missing relationships or duplicate entries.
- **Security Audit**: Verifying that sharing rules, role assignments, and field-level permissions continue to comply with data privacy policies.
- **Bug Resolution Logs**: Maintaining a centralized log of bugs and their resolutions to prevent recurrence and streamline support.

The system should also be monitored using Salesforce's built-in audit trails and diagnostic tools to track administrative and user activity.

Documentation Strategy

Comprehensive documentation is essential for future development, onboarding new administrators, and system troubleshooting. The following deliverables were prepared:

- **Technical Documentation**: Contains summaries of each object, relationship, validation rule, Apex class, and automation logic. This document is suitable for future developers or Salesforce administrators.
- **User Guide**: Provides a functional overview of each module, including how to create bookings, assign employees, manage travel packages, and review dashboards.
- **Testing Documentation**: Contains the test plan, test case results, and sample input/output screenshots (developed in Phase 4).
- **Security Model Document**: Illustrates the profiles, roles, permission sets, and sharing rules applied across the system.

All documentation is included in the capstone submission and maintained in a structured format for future reference.

Troubleshooting Approach

Common system issues were anticipated and categorized to ensure efficient resolution:

Issue Category	Description	Suggested Resolution
Data Validation Errors	Triggered when input fails business rules	Review validation criteria and provide user-friendly messages
Missing Lookup Relationships	Occurs when required related records are absent	Use validation rules or default lookup population

Incomplete Automation	When Flows or Triggers do not execute properly	Use debug logs and Flow error emails for diagnosis
Role/Permission Conflicts	Access errors for users	Review profile and permission set assignments
Duplicate Entry Conflicts	Triggered by duplicate rule configuration	Alert user and guide them through manual resolution

Each error encountered during development was logged, tested, and corrected. Debug Logs and System Audit Trails were especially useful during troubleshooting.

Future Enhancements

To scale and improve the system for long-term business use, the following enhancements are proposed:

1. **Chatbot Integration**:
   - Enable conversational assistance for booking inquiries and FAQs using Salesforce Einstein Bot.
2. **AI-Based Recommendations**:
   - Recommend travel packages based on user history and preferences through predictive analytics.
3. **Customer Portal**:
   - Develop a secure external portal where customers can view their bookings, provide feedback, and make payments.

4. **Mobile-Optimized LWC Extensions**:
    - Build additional mobile-friendly Lightning Web Components to streamline guide assignments and payment collections.
5. **Email-to-Case Integration**:
    - Automatically convert customer emails into cases for efficient customer service follow-up.

Conclusion

The completion of the *Tours & Travels Salesforce CRM* project demonstrates how a custom-built CRM platform can streamline end-to-end operations in the travel industry. By leveraging Salesforce's declarative tools and Apex programming capabilities, the system addresses challenges such as manual bookings, delayed communication, and inefficient task assignments.

From requirement gathering to backend configuration, user interface design, security enforcement, and test validation, the CRM has been structured to meet the real-world needs of travel agencies. It supports a wide range of operations including booking lifecycle management, employee coordination, payment processing, and customer feedback collection.

Overall, the project not only showcases technical competency in Salesforce development but also reflects an understanding of business process optimization, data integrity, and user-centered design. This CRM solution can serve as a strong foundation for future expansions and real-world deployment.