📌 Phase 1: Problem Understanding & Industry Analysis

Goal: Frame the business case.

Tasks:

Document airline challenges (scheduling, booking, refunds, etc.).

Define scope (Ticketing, Scheduling, Passenger Mgmt, Refunds).

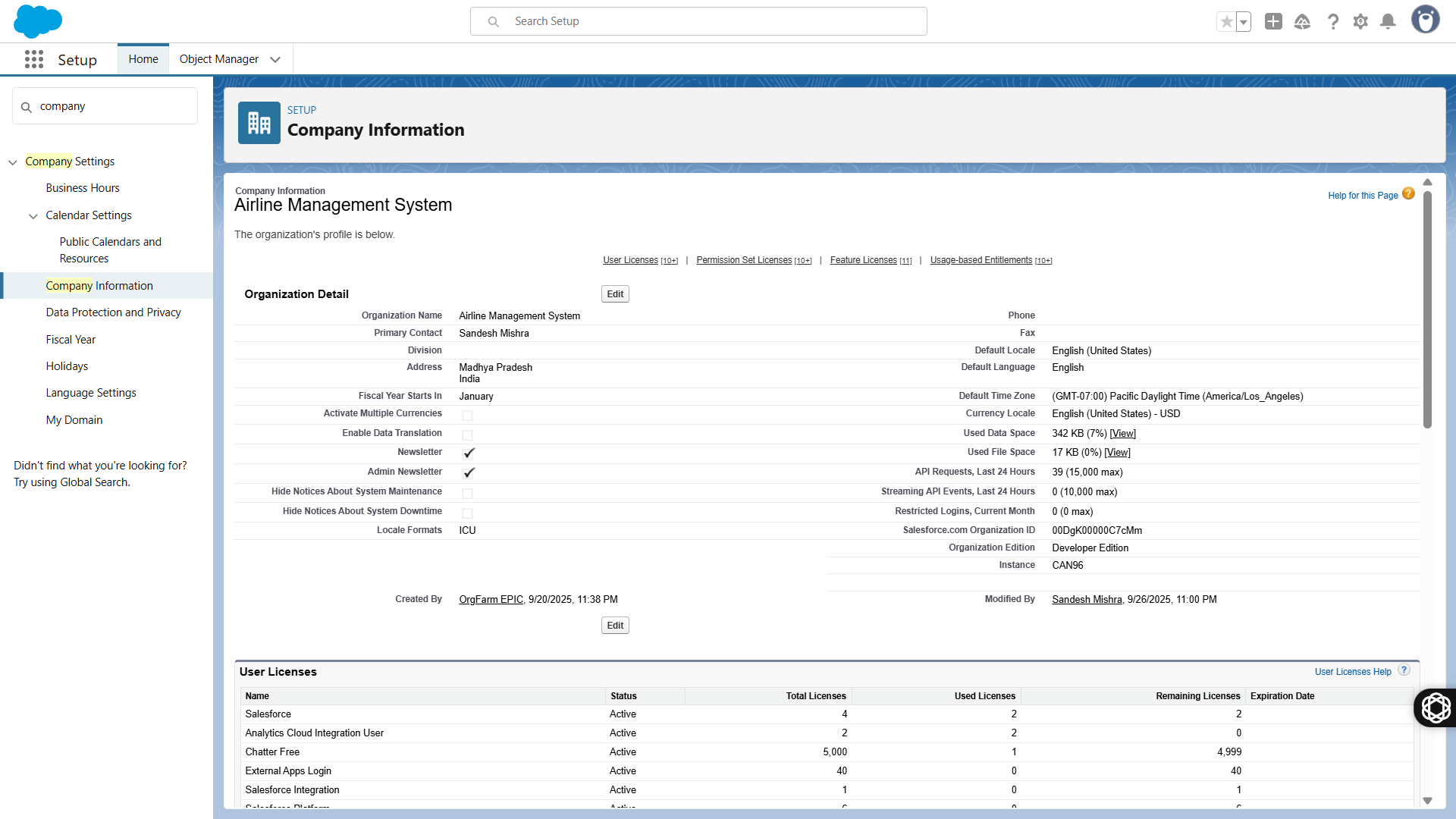
Identify stakeholders:

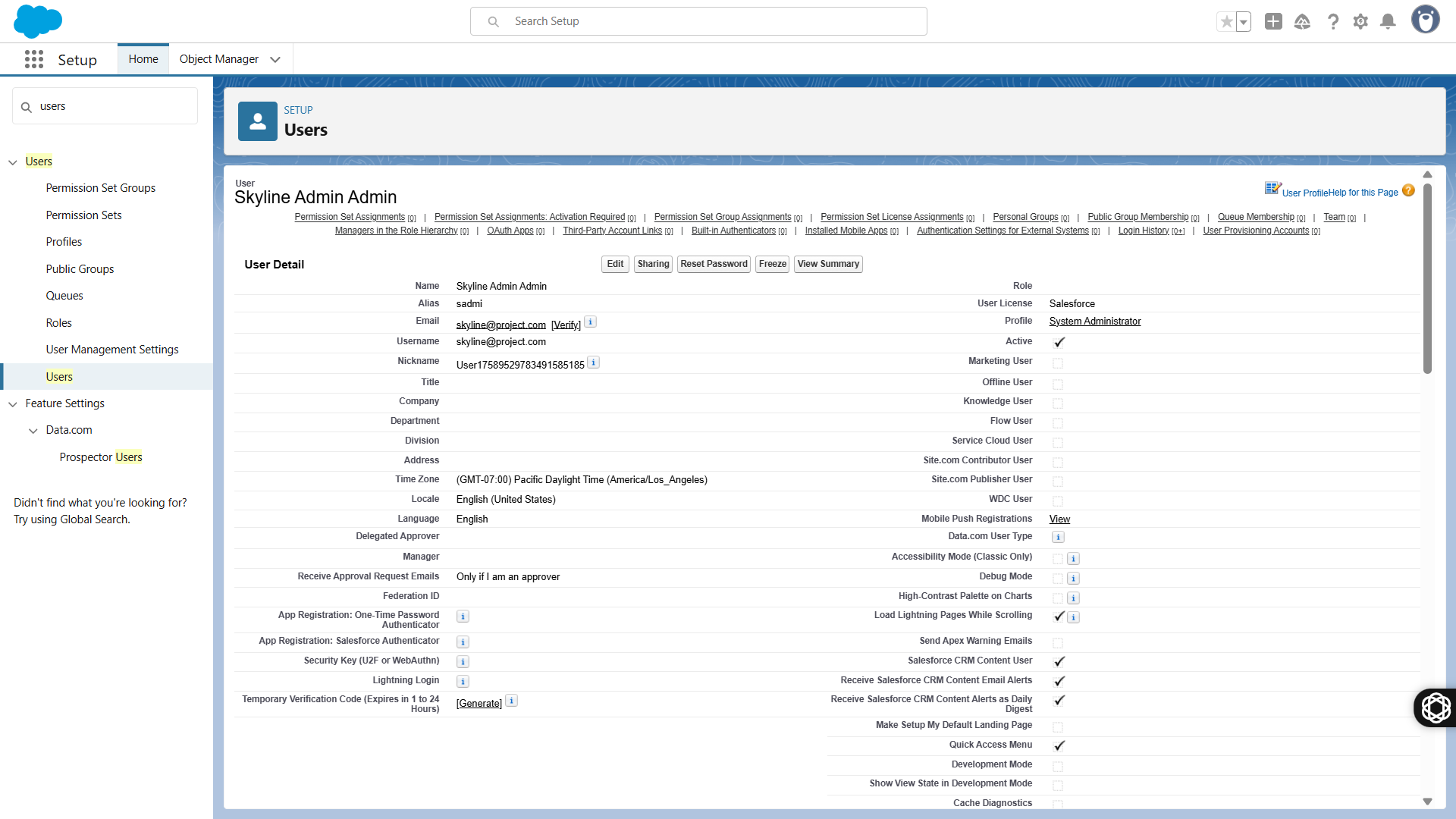
Admin (system owner)

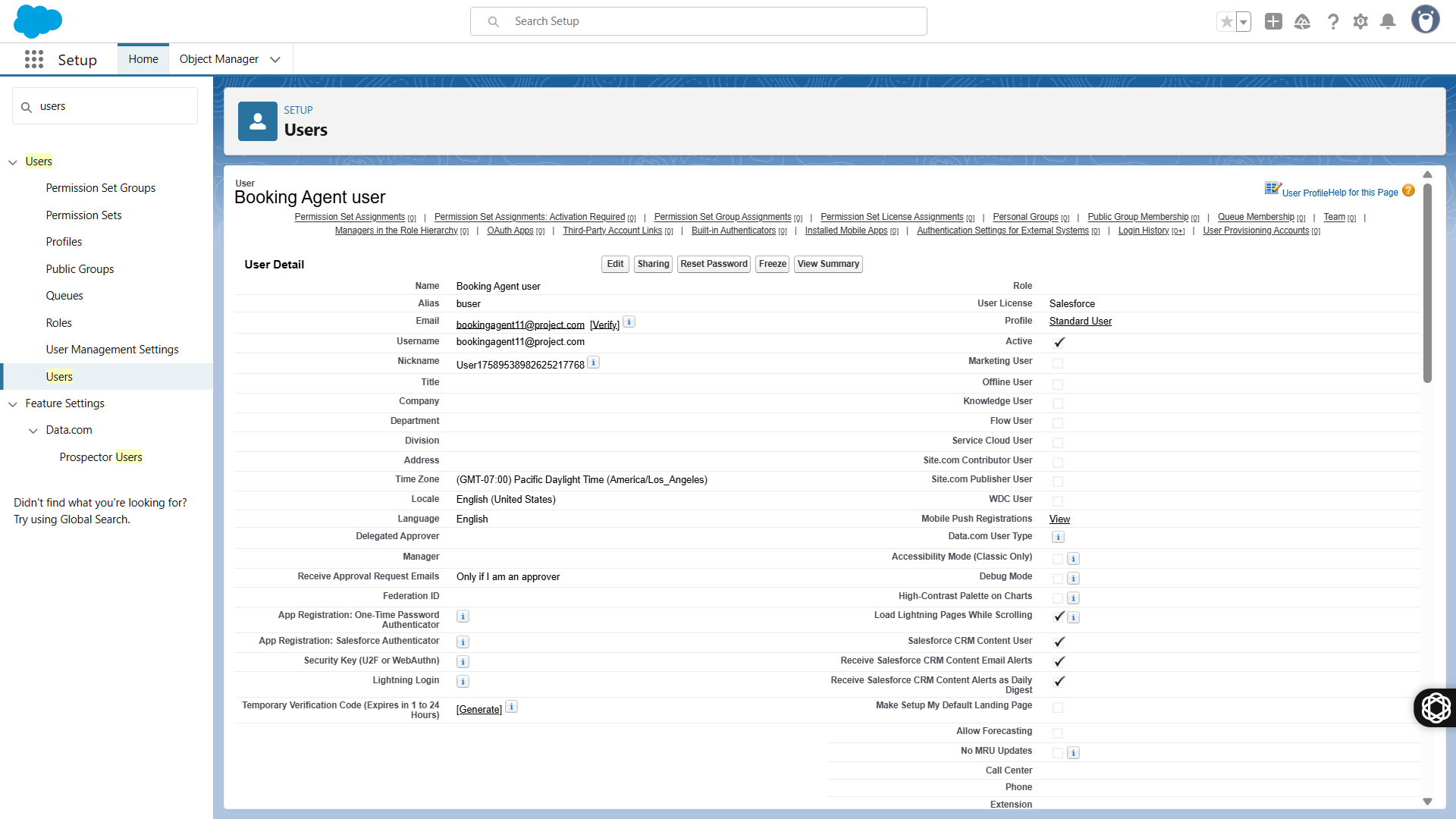
Booking Agent (internal Salesforce user)

Passenger (Contact, no login).

✅ Output: Problem doc + scope doc (like the one you already wrote).







📌 Phase 2: Org Setup & Configuration

Goal: Prepare the Developer Org.

Tasks:

Signup: developer.salesforce.com

.

Company Info → rename to Skyline Airlines.

Create Users:

Admin (you).

Booking Agent (profile: Airline Staff).

Roles & Profiles:

Admin → Full Access.

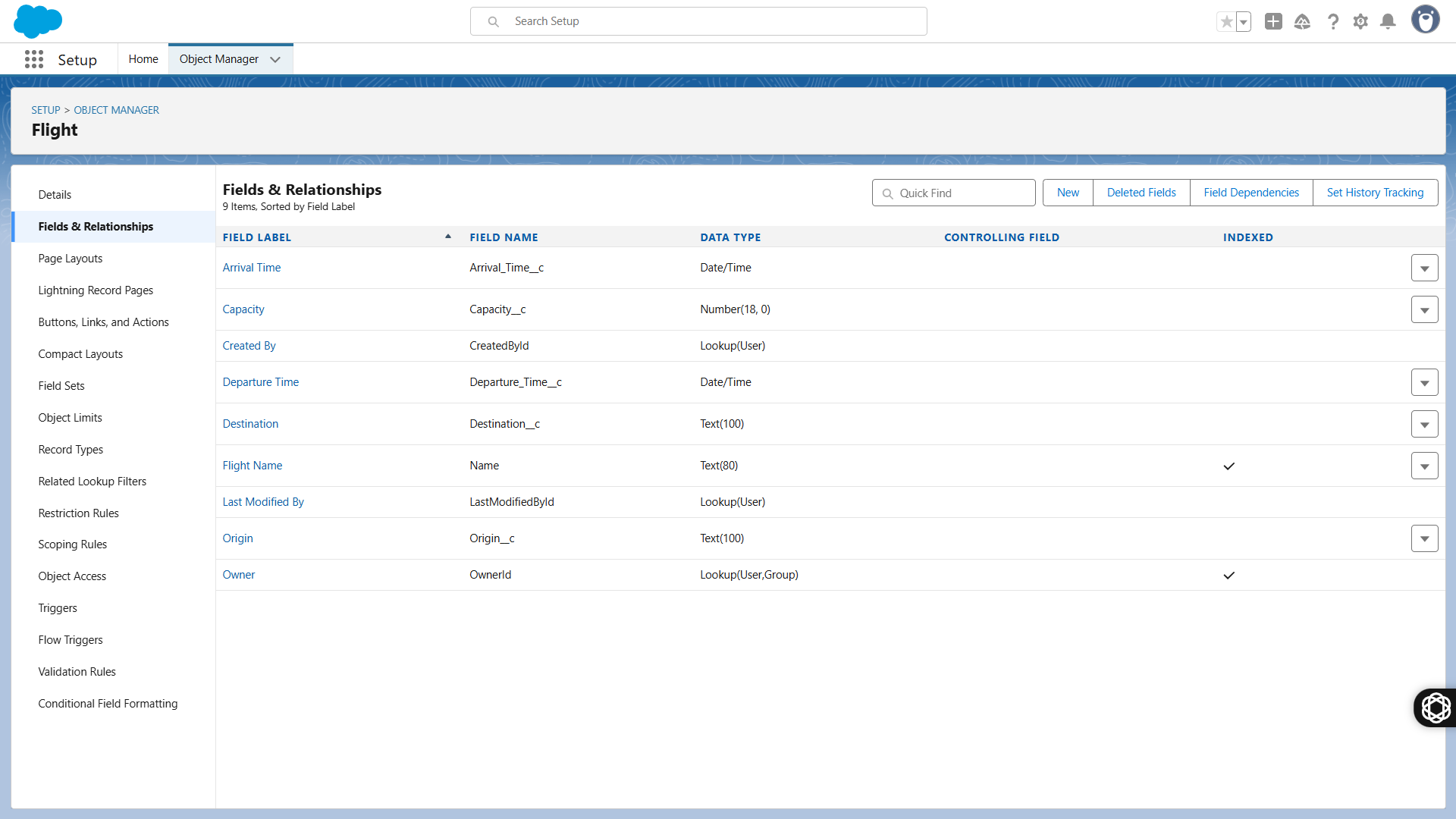
Airline Staff → Limited to Bookings, Flights.

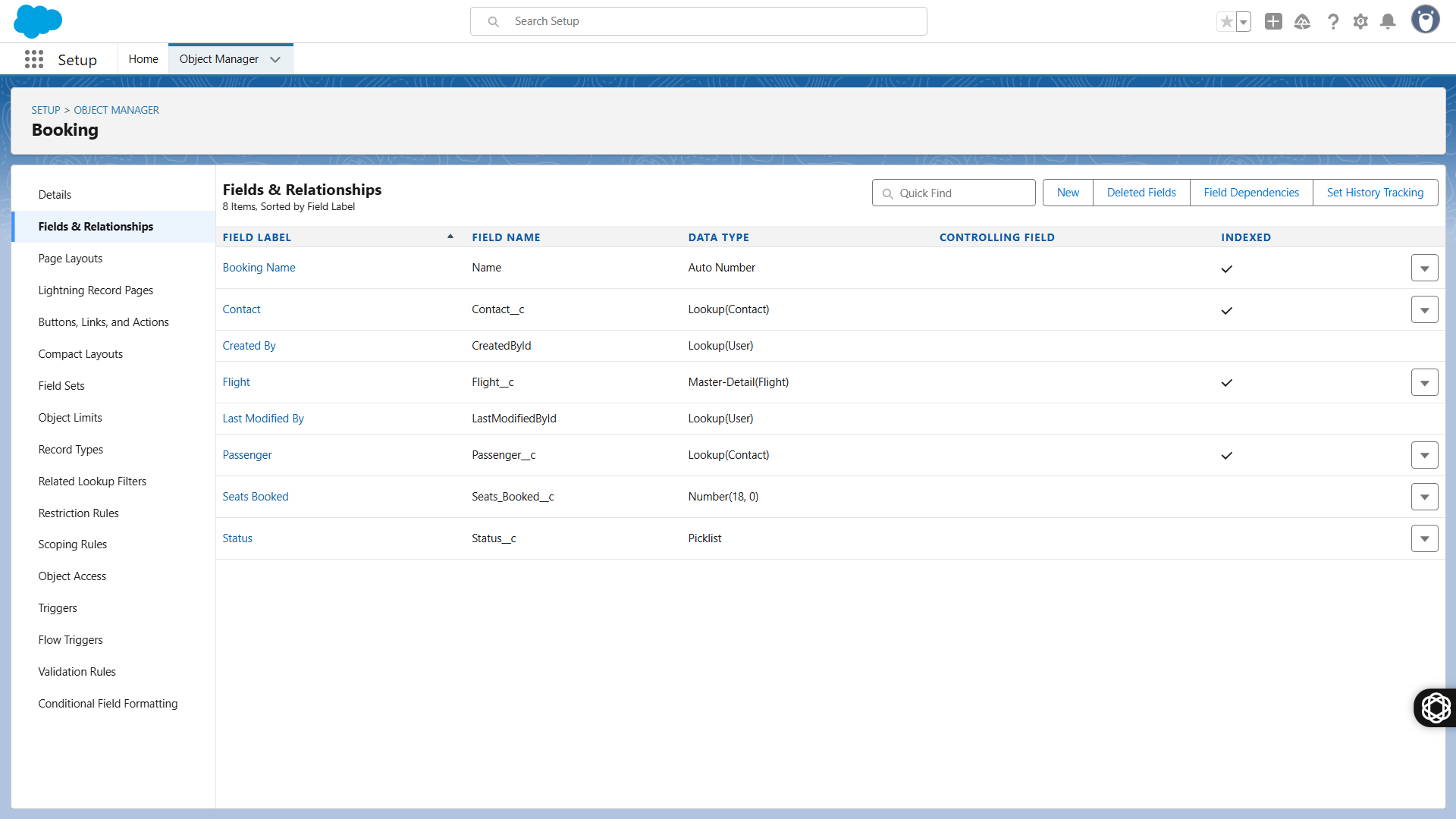
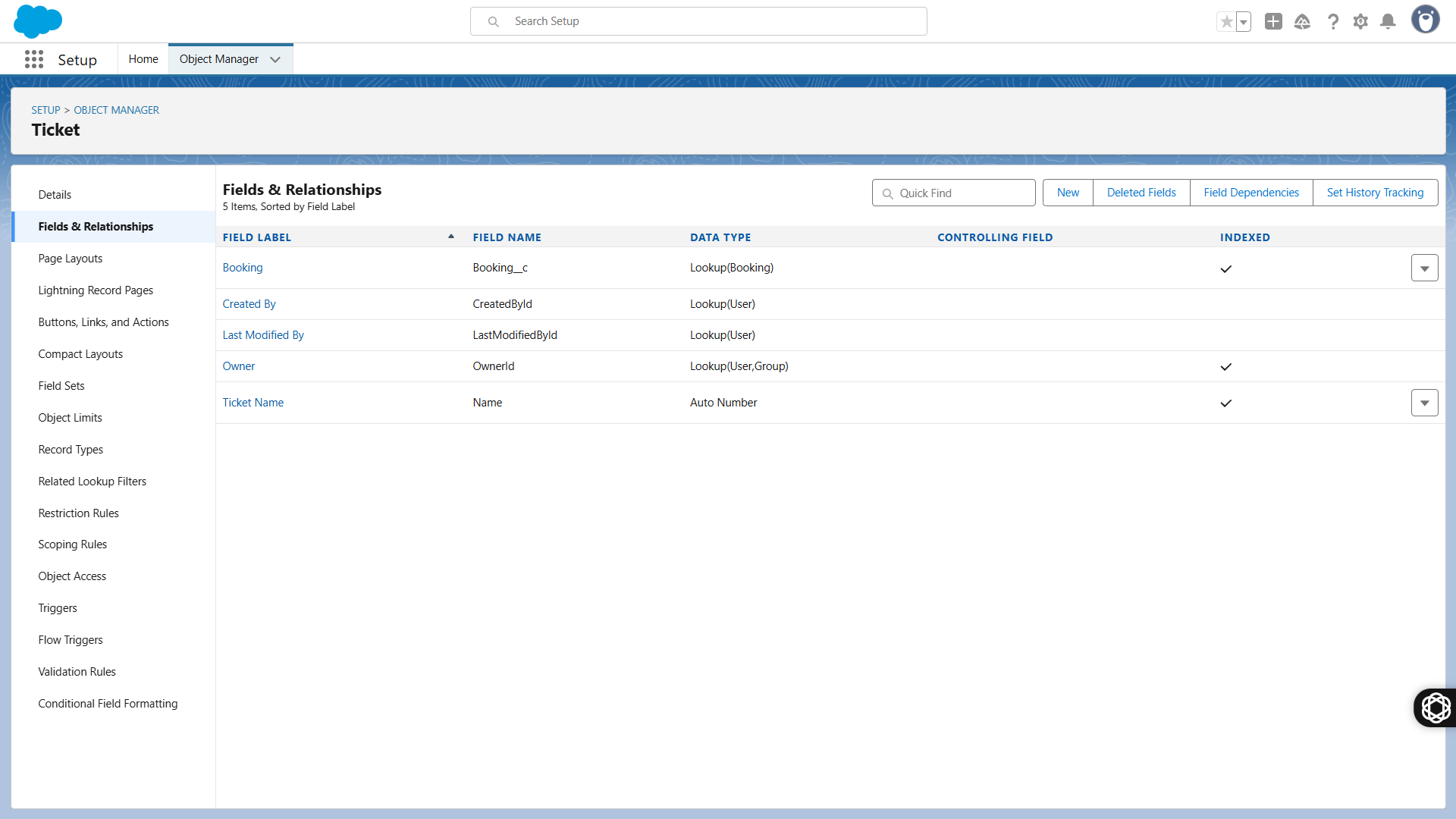
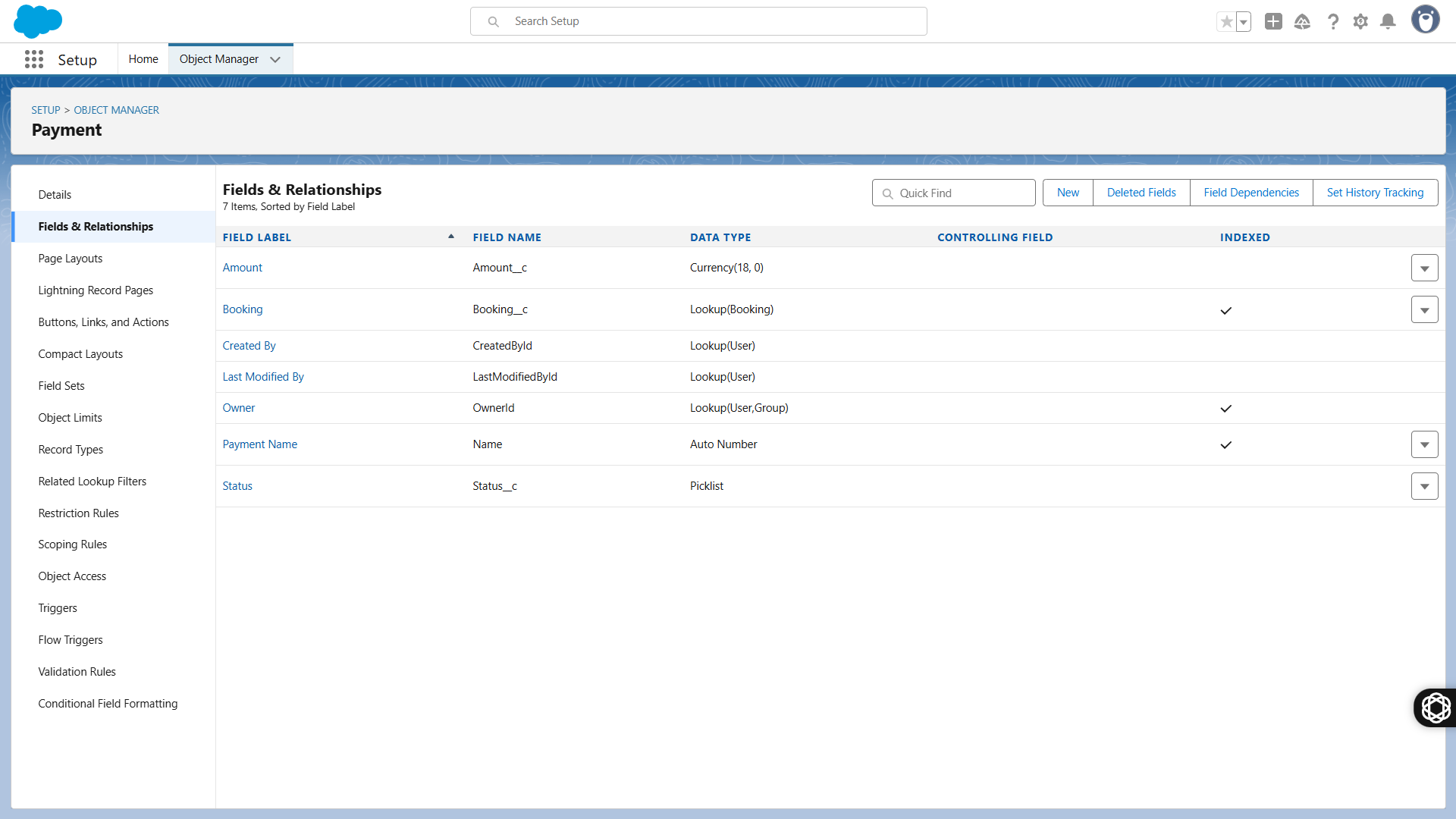
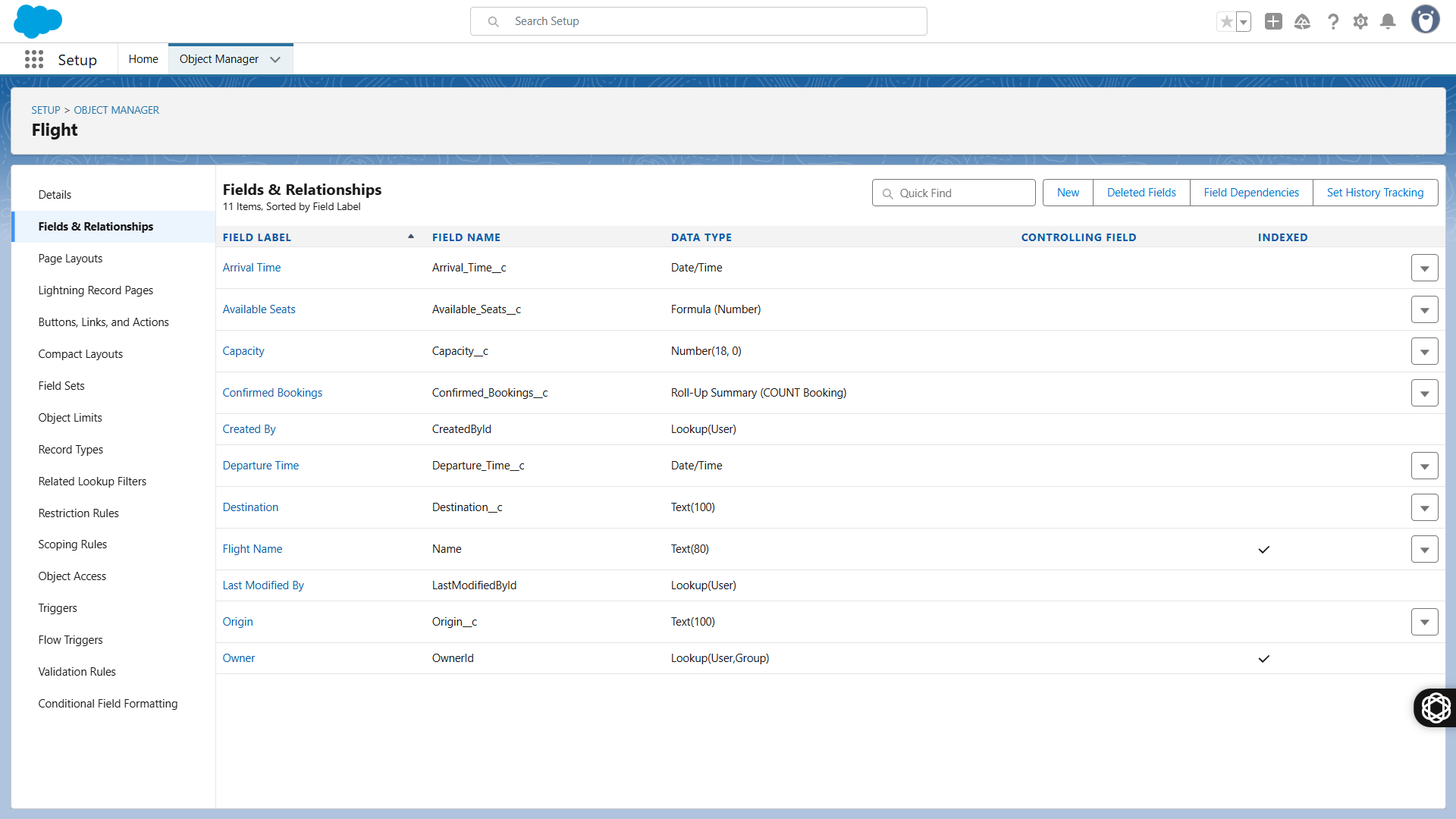
OWD (Org-Wide Defaults):

Flights = Public Read/Write.

Bookings = Private.

✅ Output: Org ready with users, profiles, OWD.





📌 Phase 3: Data Modeling & Relationships

Goal: Build objects + relationships.

Custom Objects:

✈️ Flight (fields: Flight No, Route, Capacity, Status).

🎟️ Booking (fields: Booking No, Status, Passenger Lookup, Flight Lookup).

💳 Ticket (fields: Ticket No, Seat No, Booking Lookup).

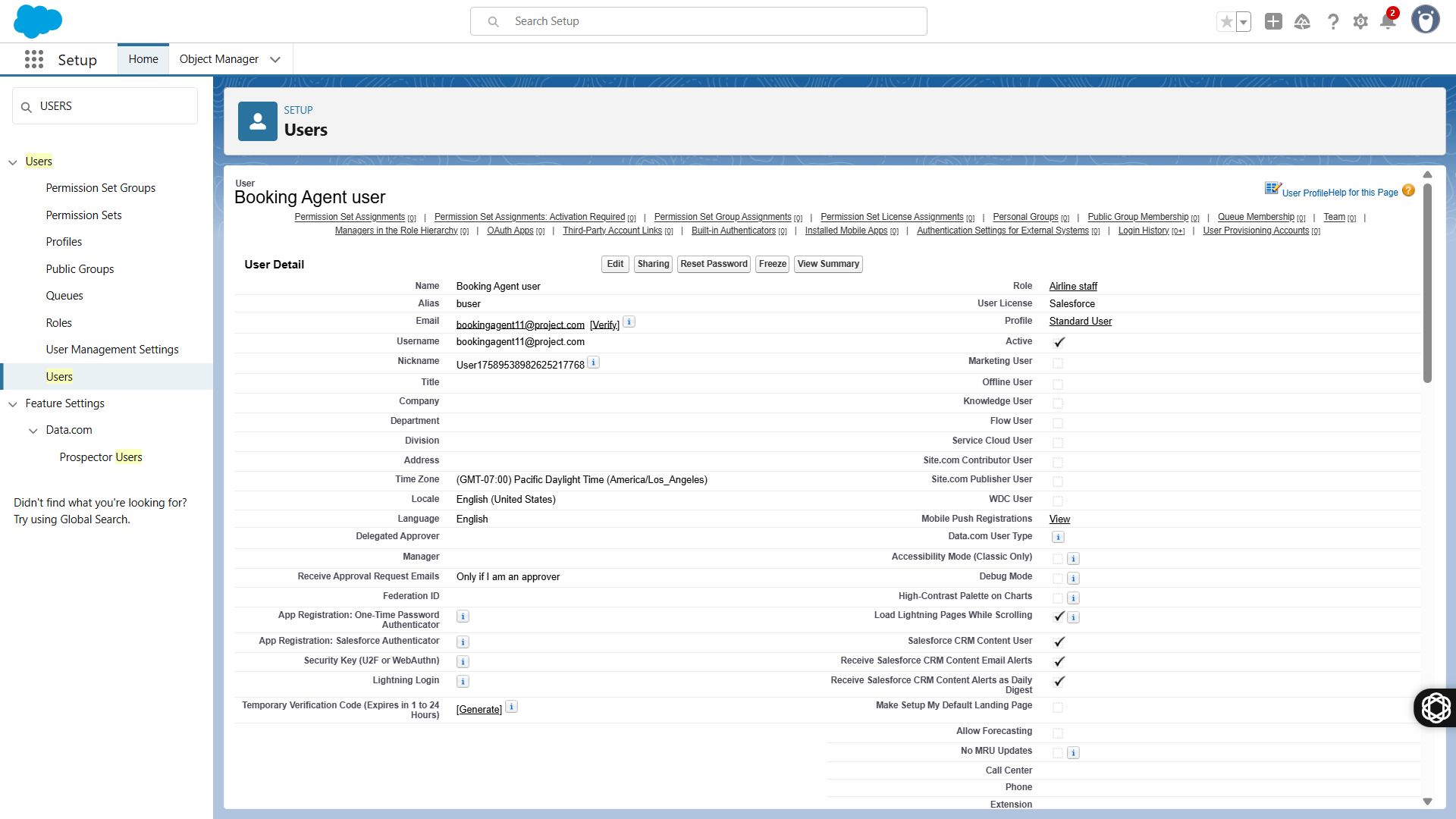
💰 Payment (fields: Payment ID, Amount, Mode, Status, Booking Lookup).

🛤 Route (fields: Origin, Destination, Distance, Duration).

Standard Object Extensions:

Contact → Passenger (fields: Passport No, DOB, Nationality).

✅ Output: ERD + objects built in Salesforce.



📌 Phase 4: Process Automation

Goal: Declarative automation (Flows, Validation).

Tasks:

Validation Rules: Prevent overbooking → Seats\_Booked <= Capacity.

Flows:

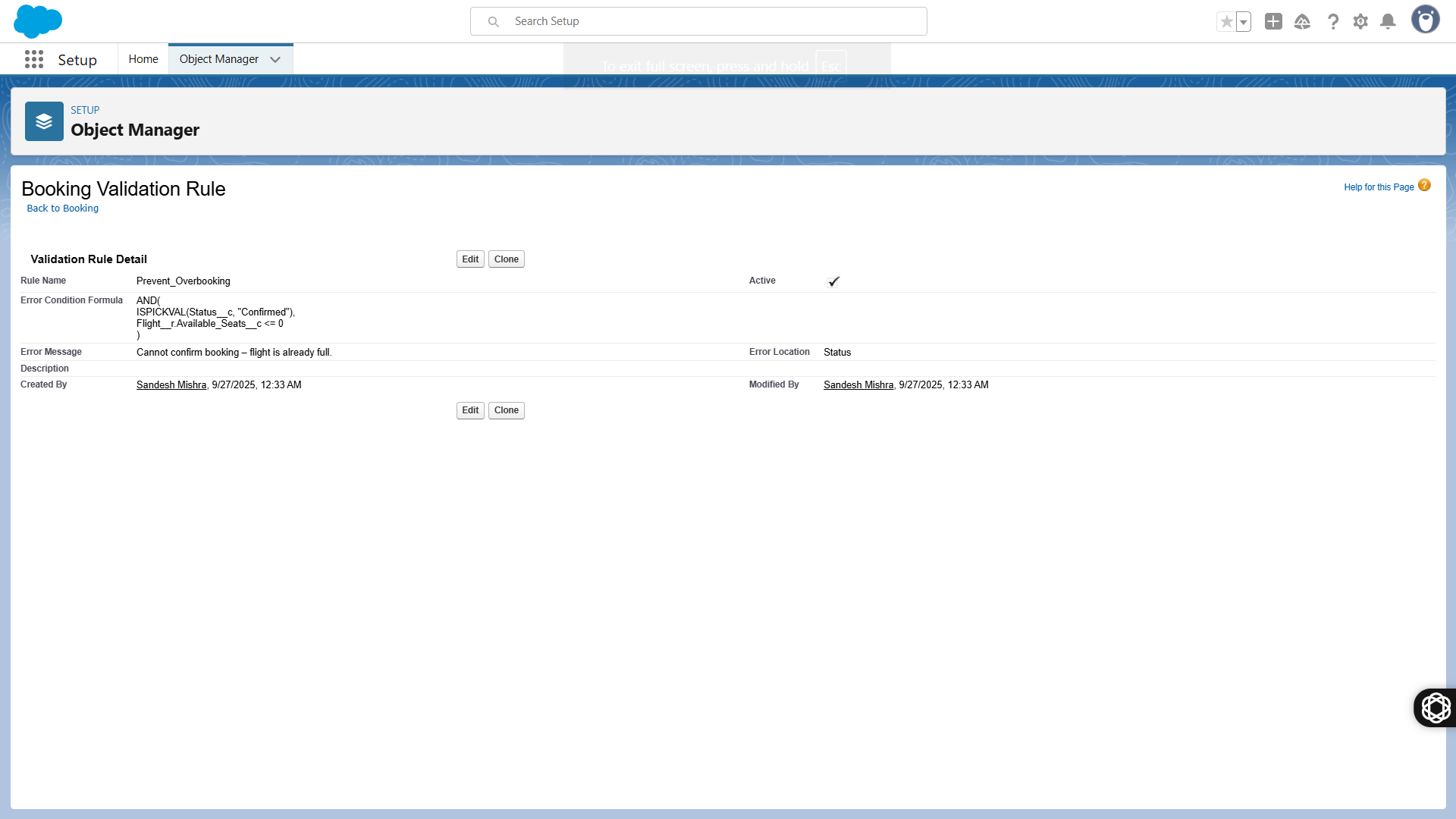
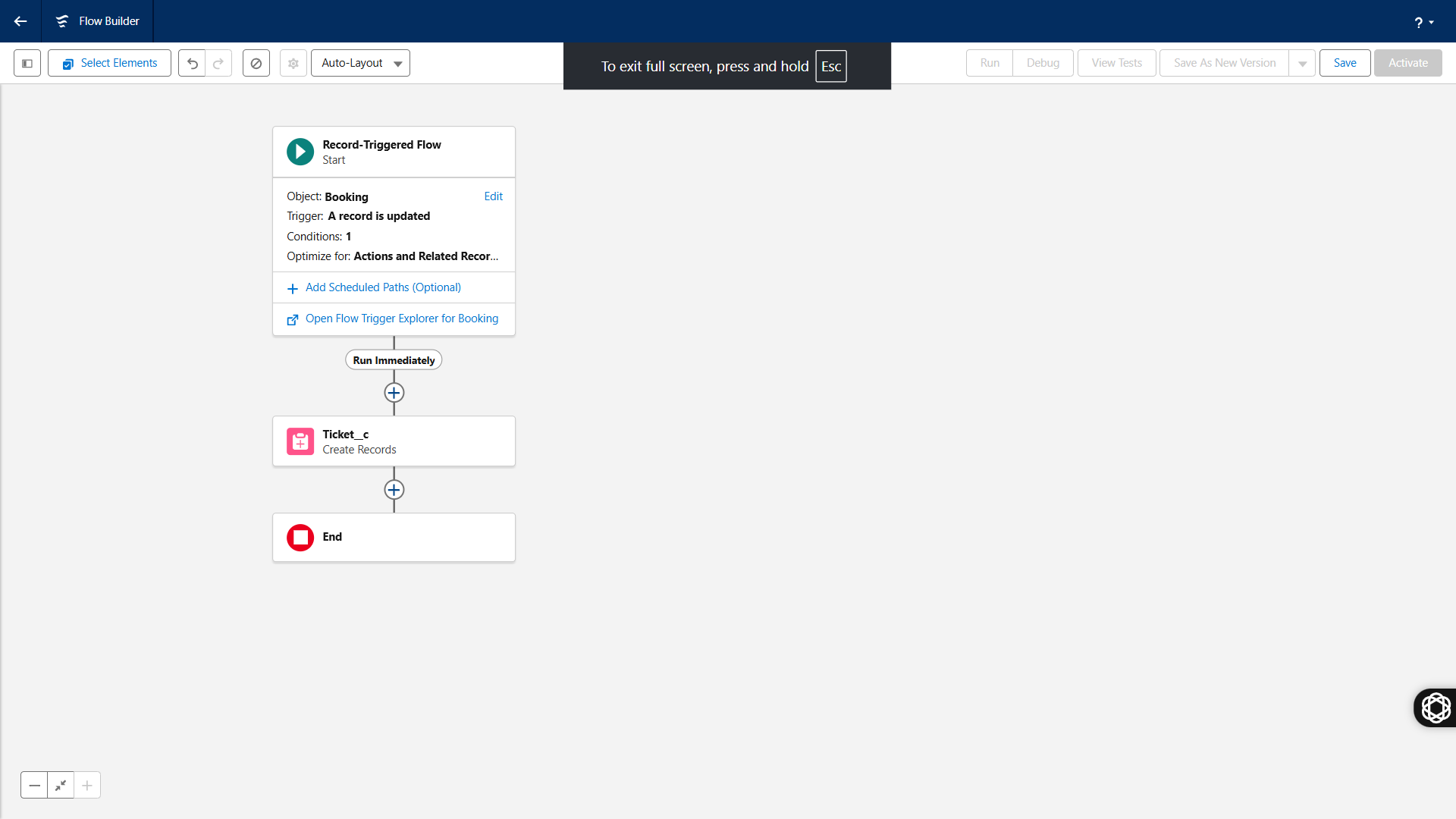
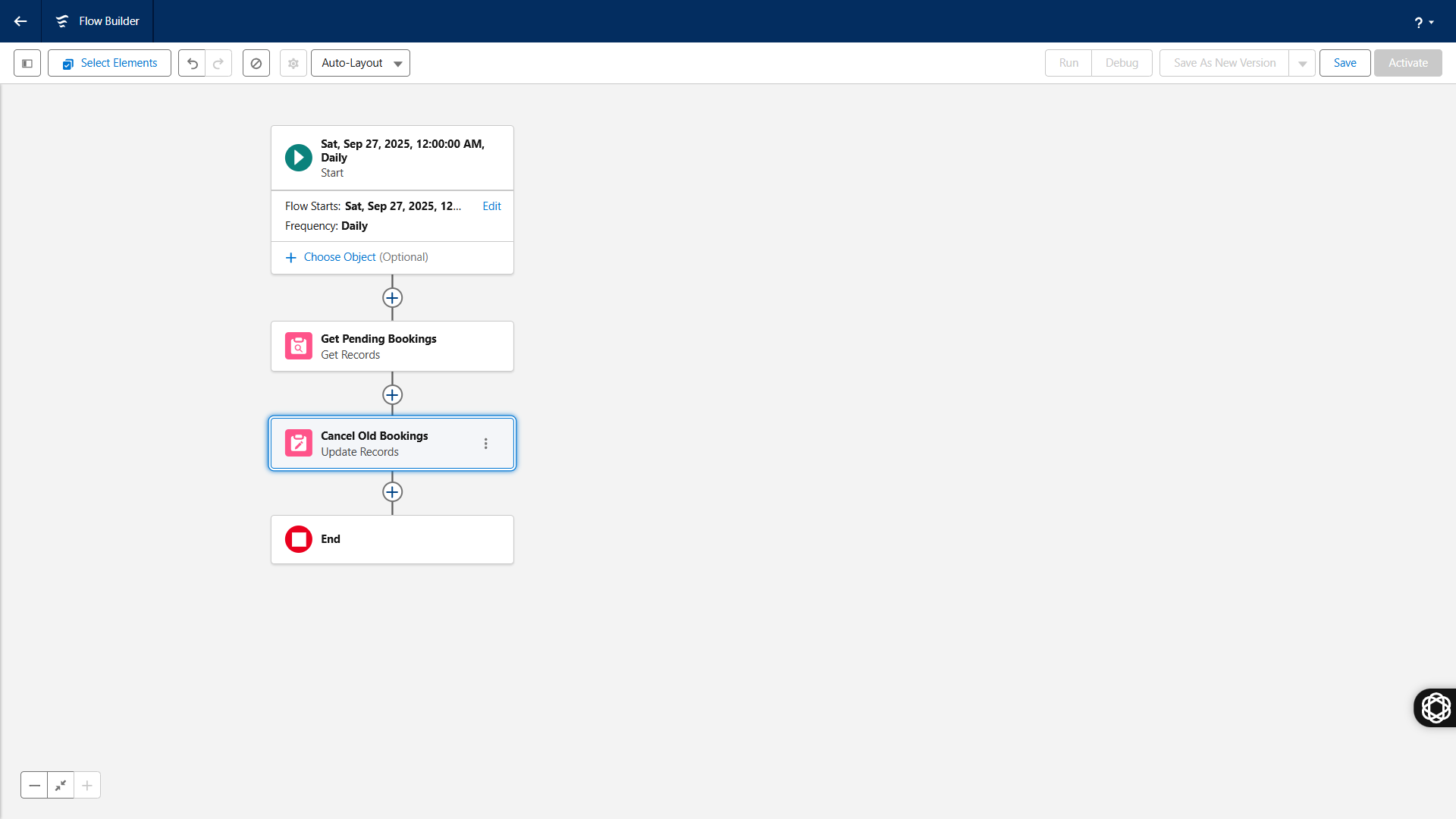
On Booking → auto-create Ticket.

On Booking Confirmation → send email.

Auto-cancel Booking if Payment not received in 24 hrs (Scheduled Flow).

Approval Process: For Refund requests.

✅ Output: Automated booking → ticket → payment flow.



📌 Phase 5: Apex Programming

Goal: Add logic that Flows can’t cover.

Triggers:

After Insert Booking → Update Flight available seats.

Before Insert Booking → Prevent duplicate booking for same passenger + flight.

Batch Apex:

Archive flights older than 1 year.

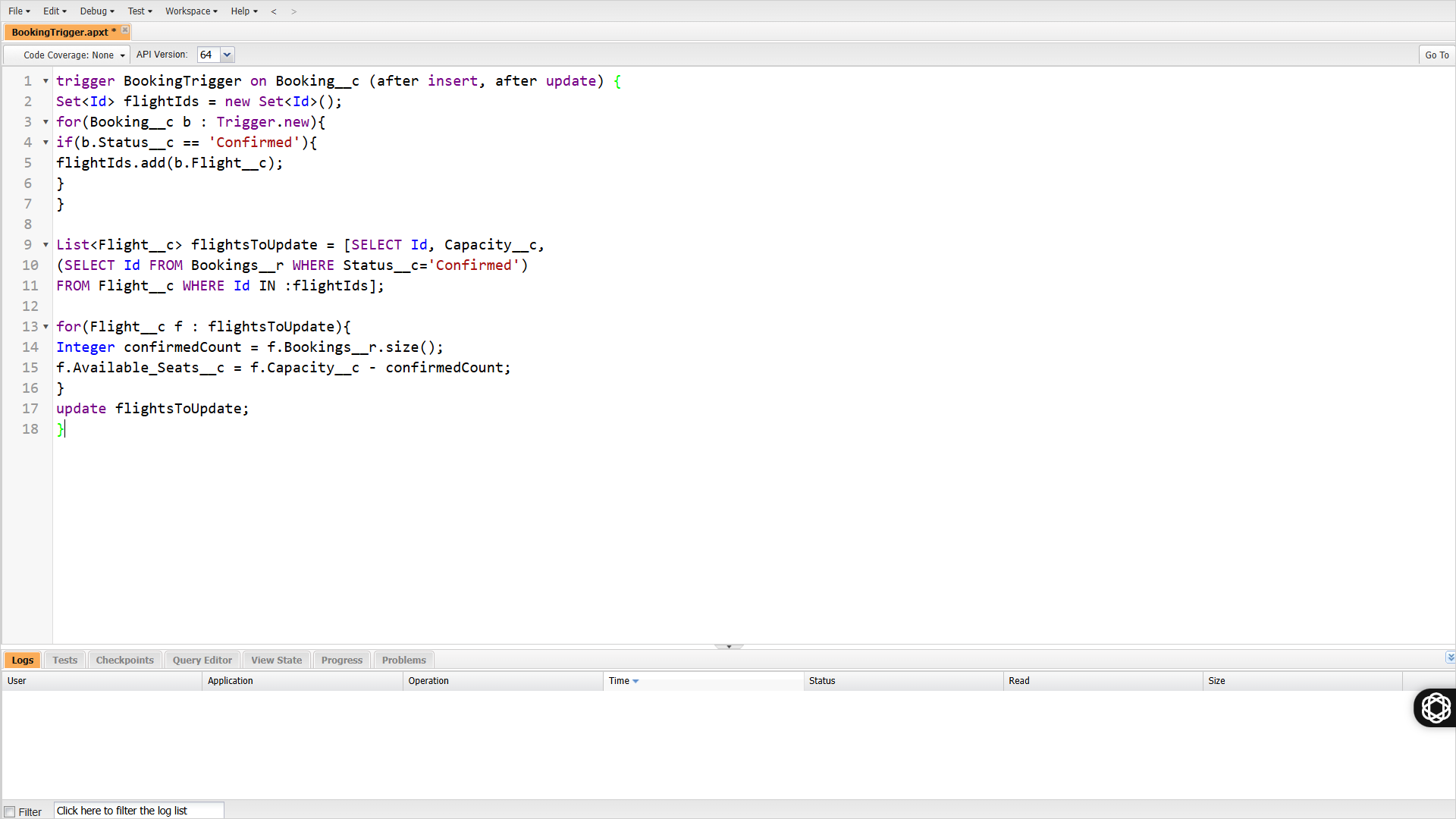
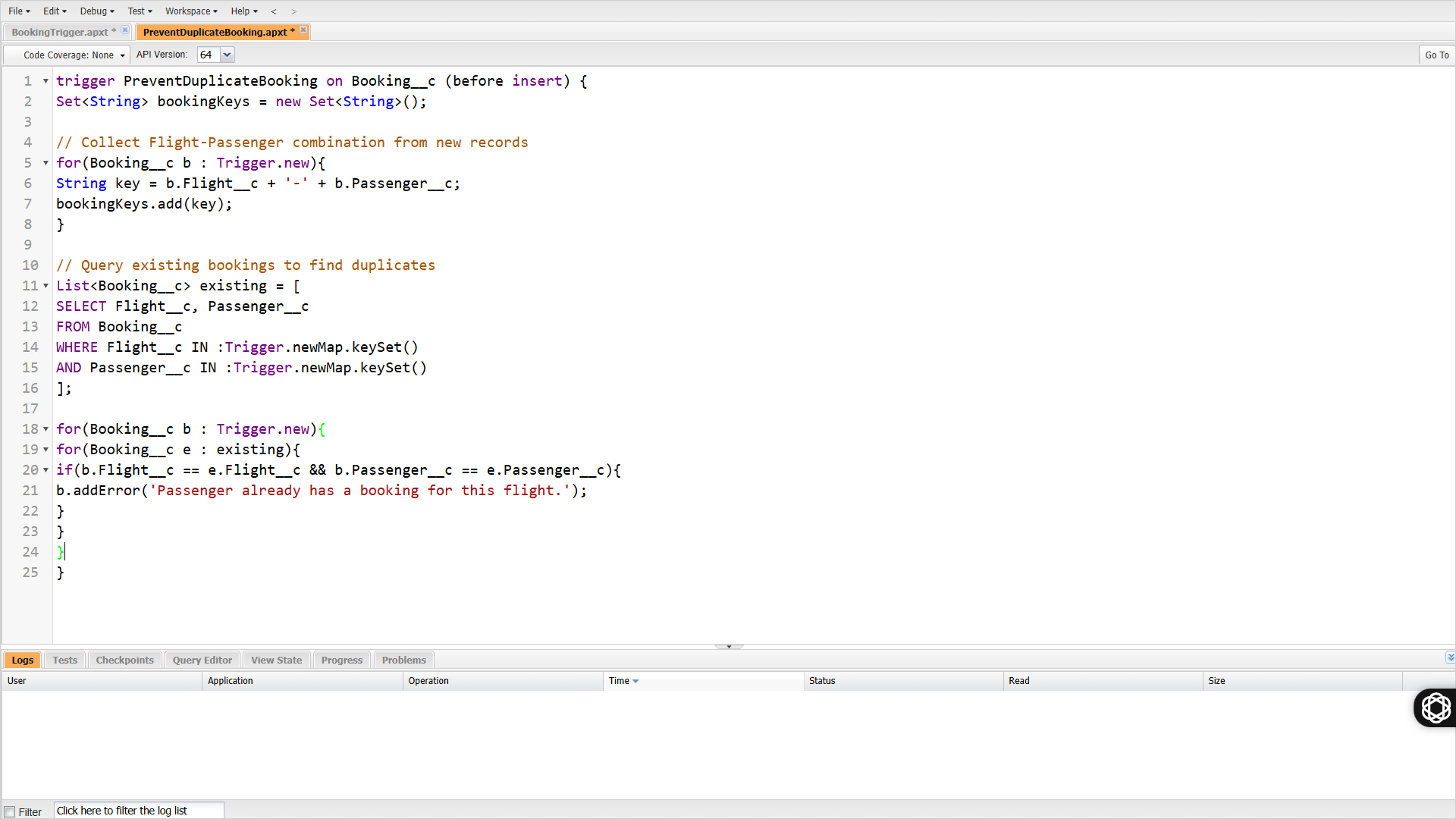
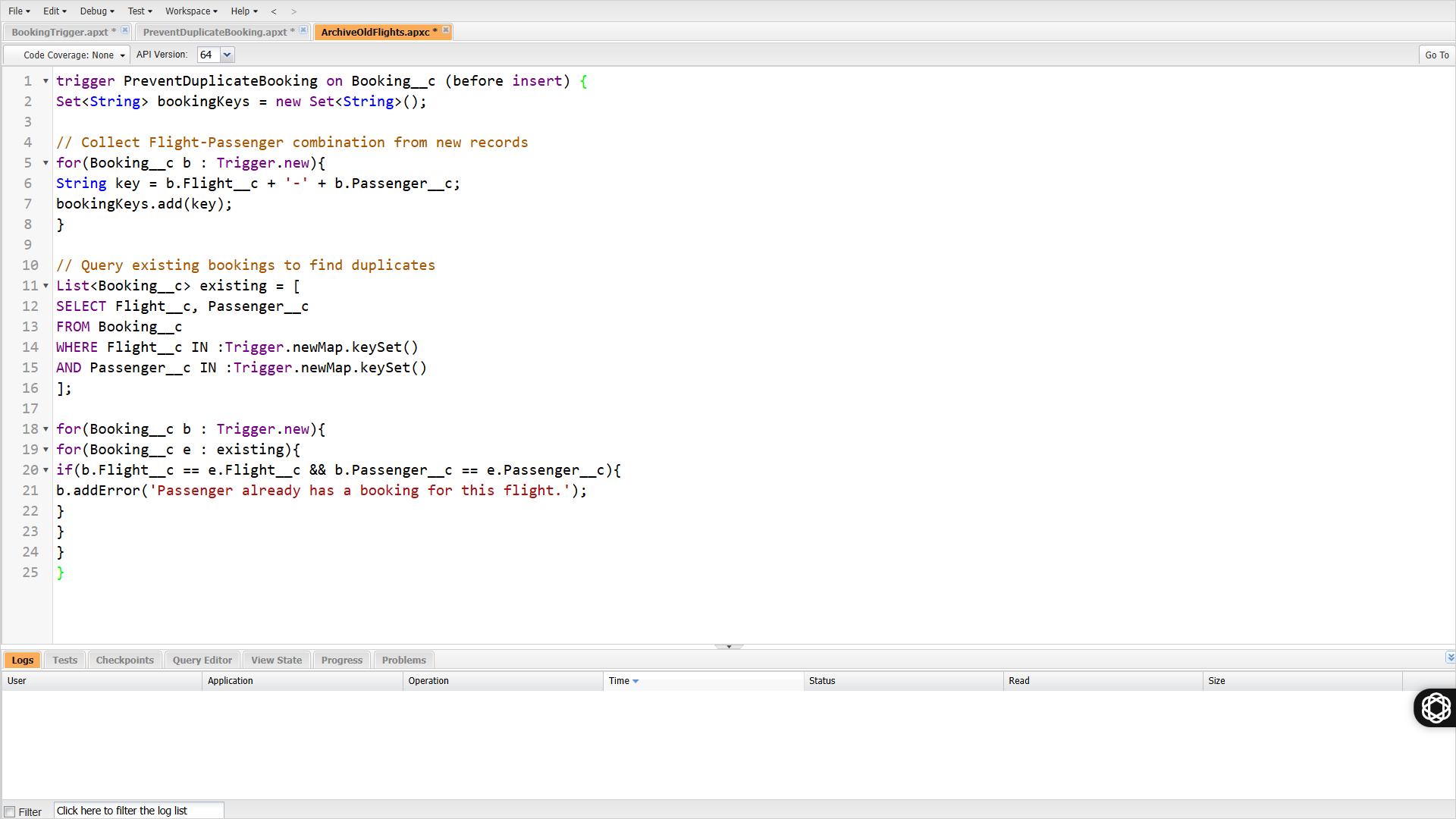
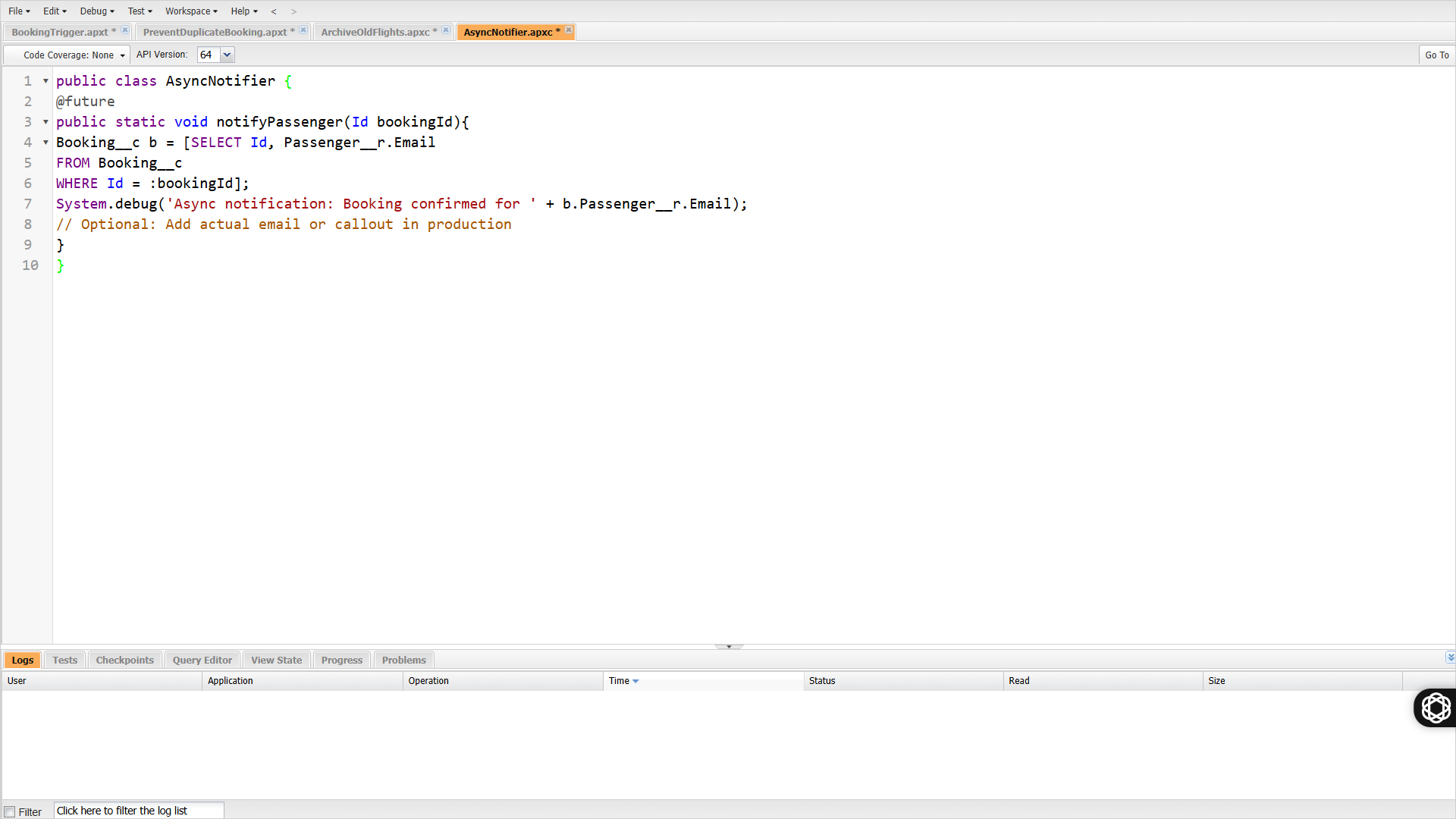
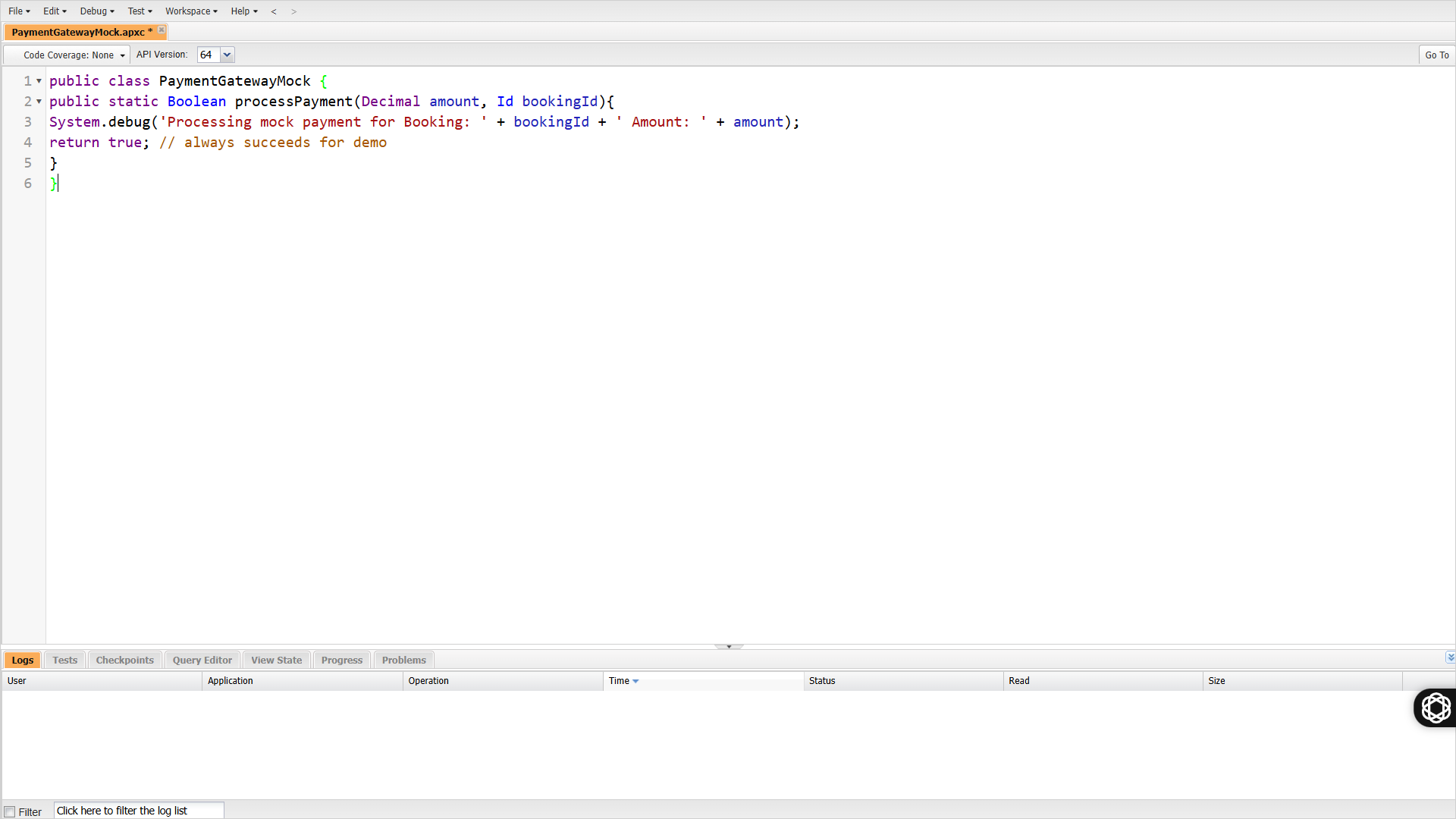
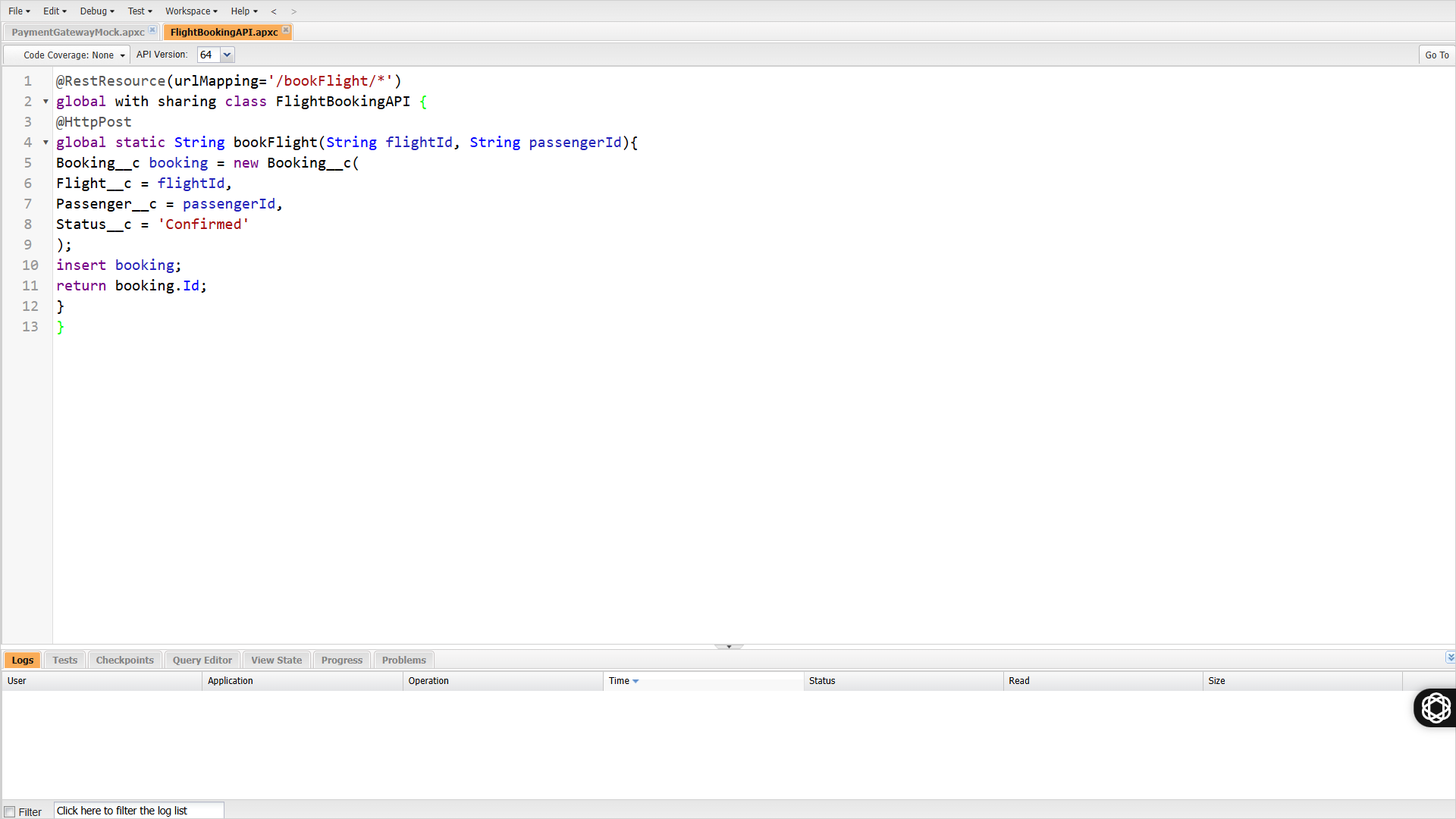
Future / Queueable Apex:

Send async SMS/email notifications (demo).

Test Classes:

Cover >75% to deploy.

✅ Output: Apex classes + triggers with test coverage.



📌 Phase 6: User Interface

Goal: Build a usable airline console.

Tasks:

Create Lightning App: Skyline Airlines AMS.

Record Pages:

Passenger → show Bookings, Tickets, Payments.

Flight → show capacity, route, passengers.

LWCs:

Flight Search.

Seat Selection (demo grid).

Flight Status Board (auto-refresh via Platform Events).

✅ Output: Agent console with LWCs.

📌 Phase 7: Integration

Goal: Showcase Salesforce integration ability.

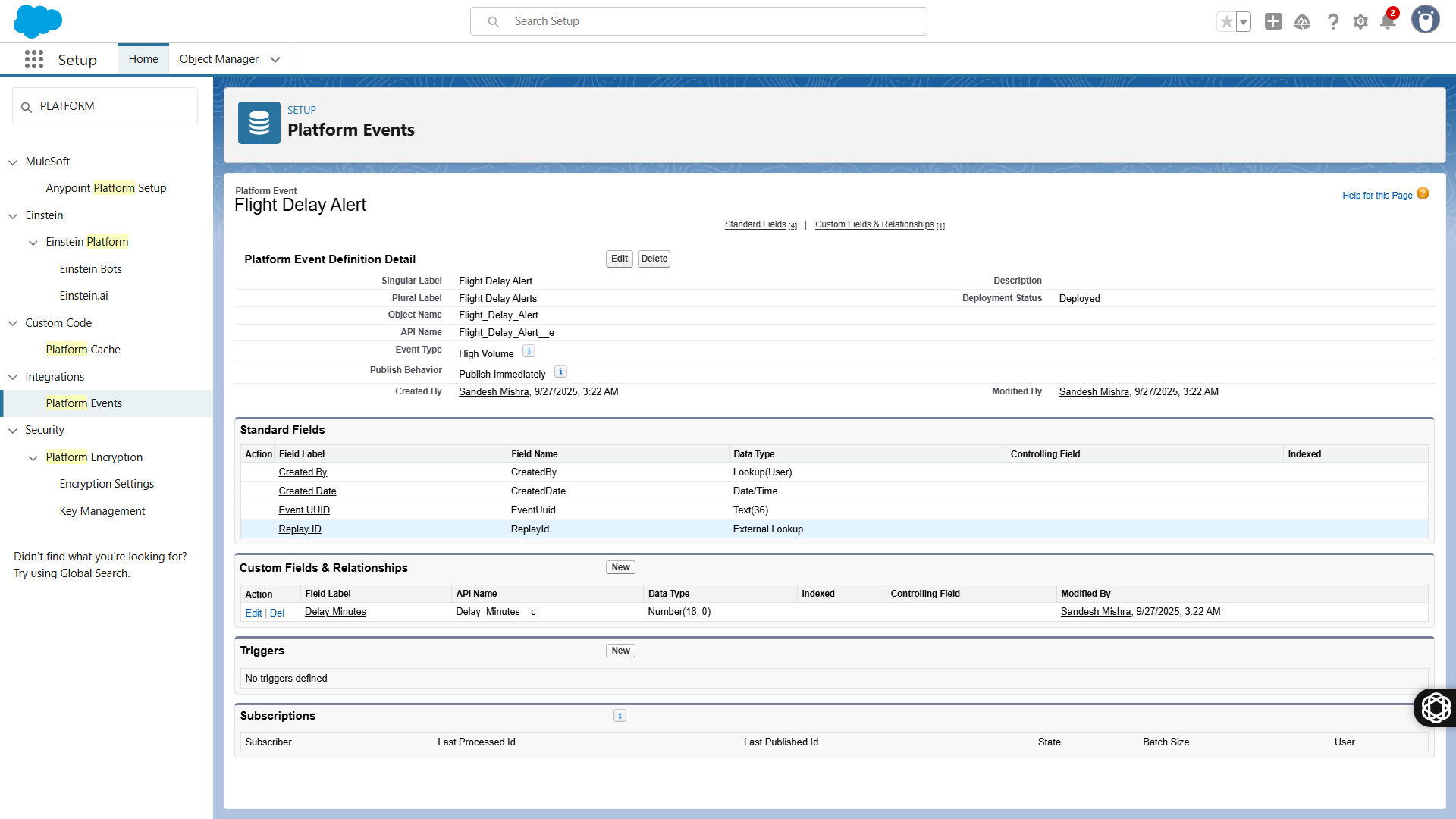
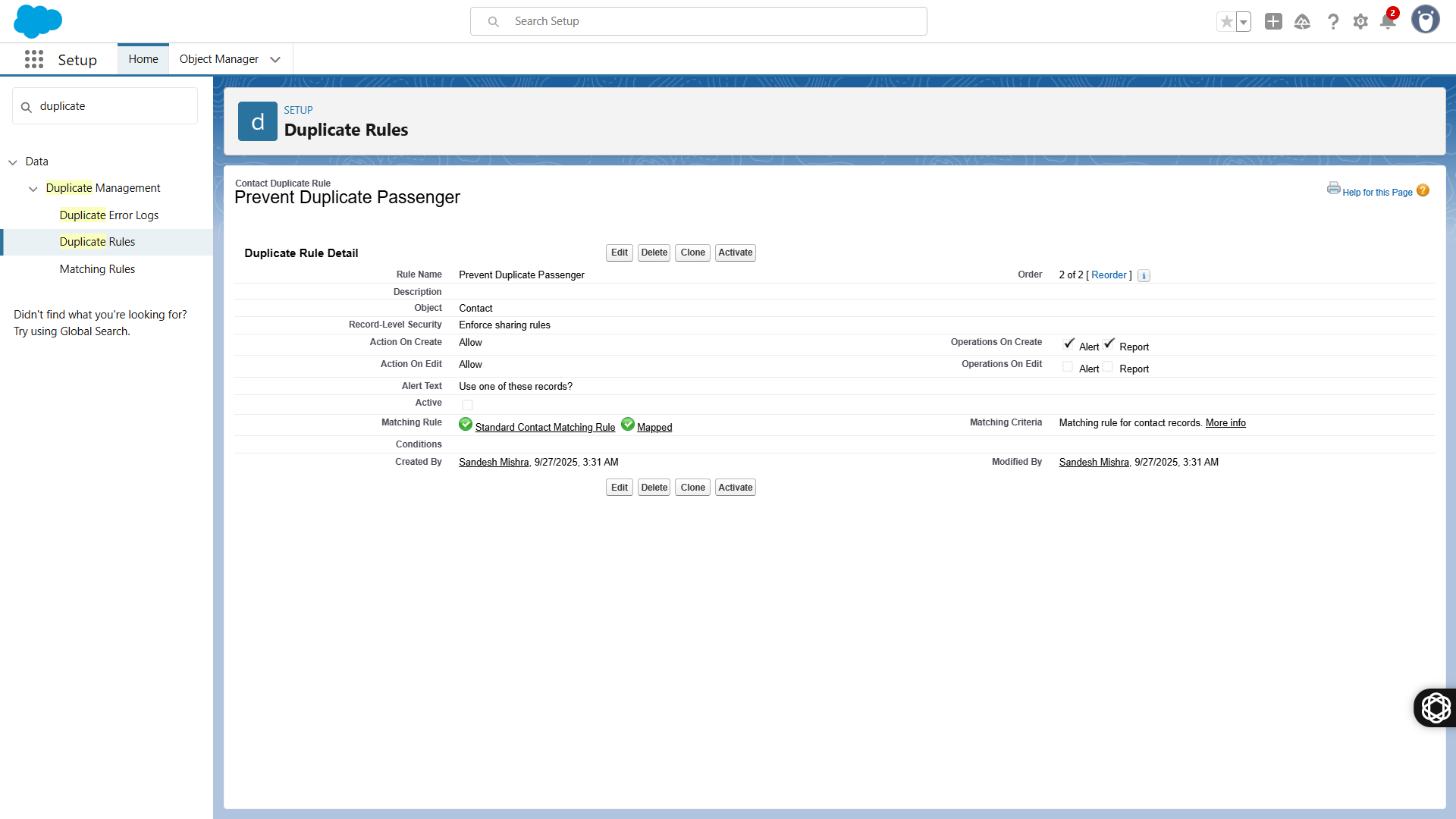
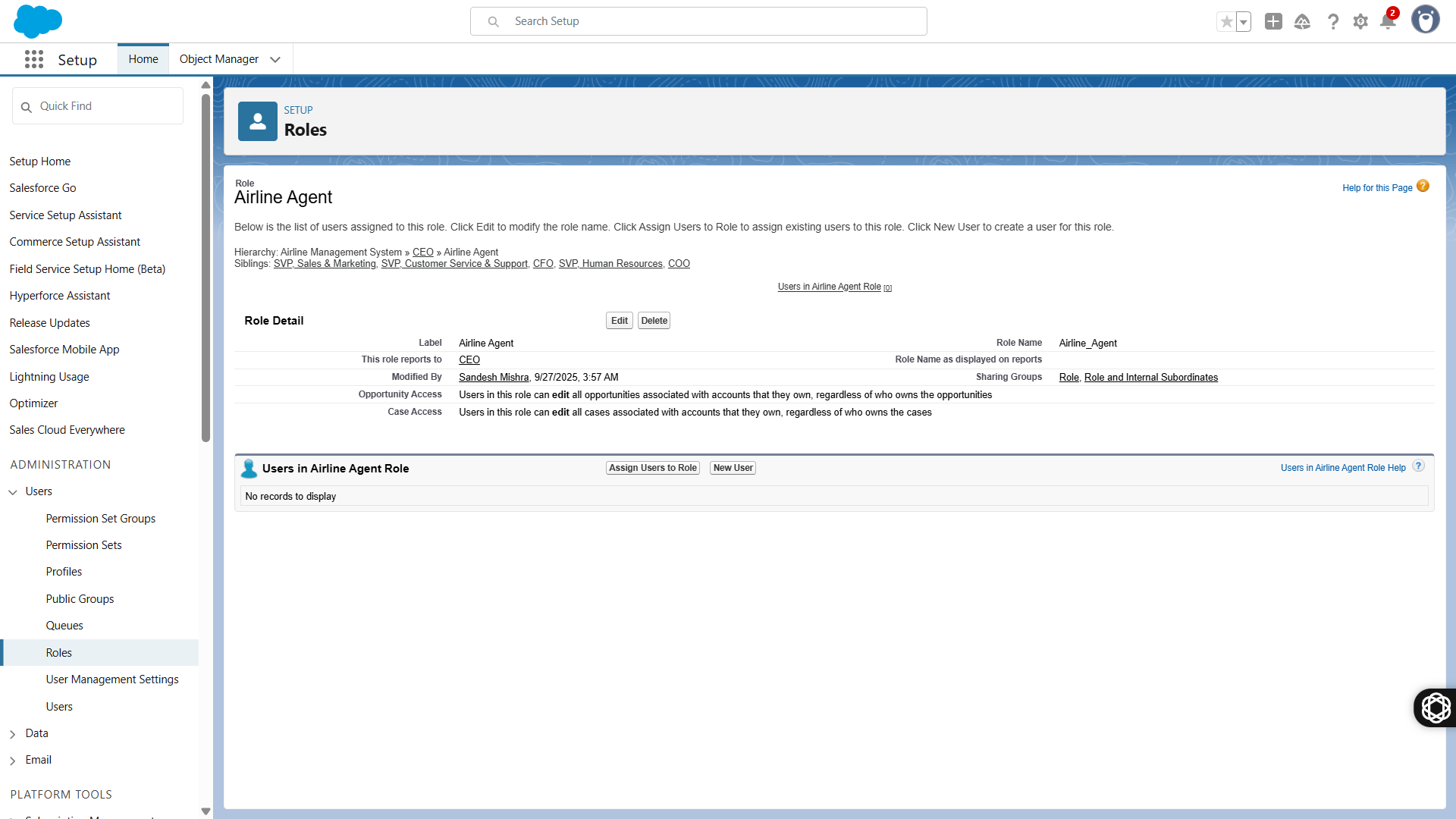
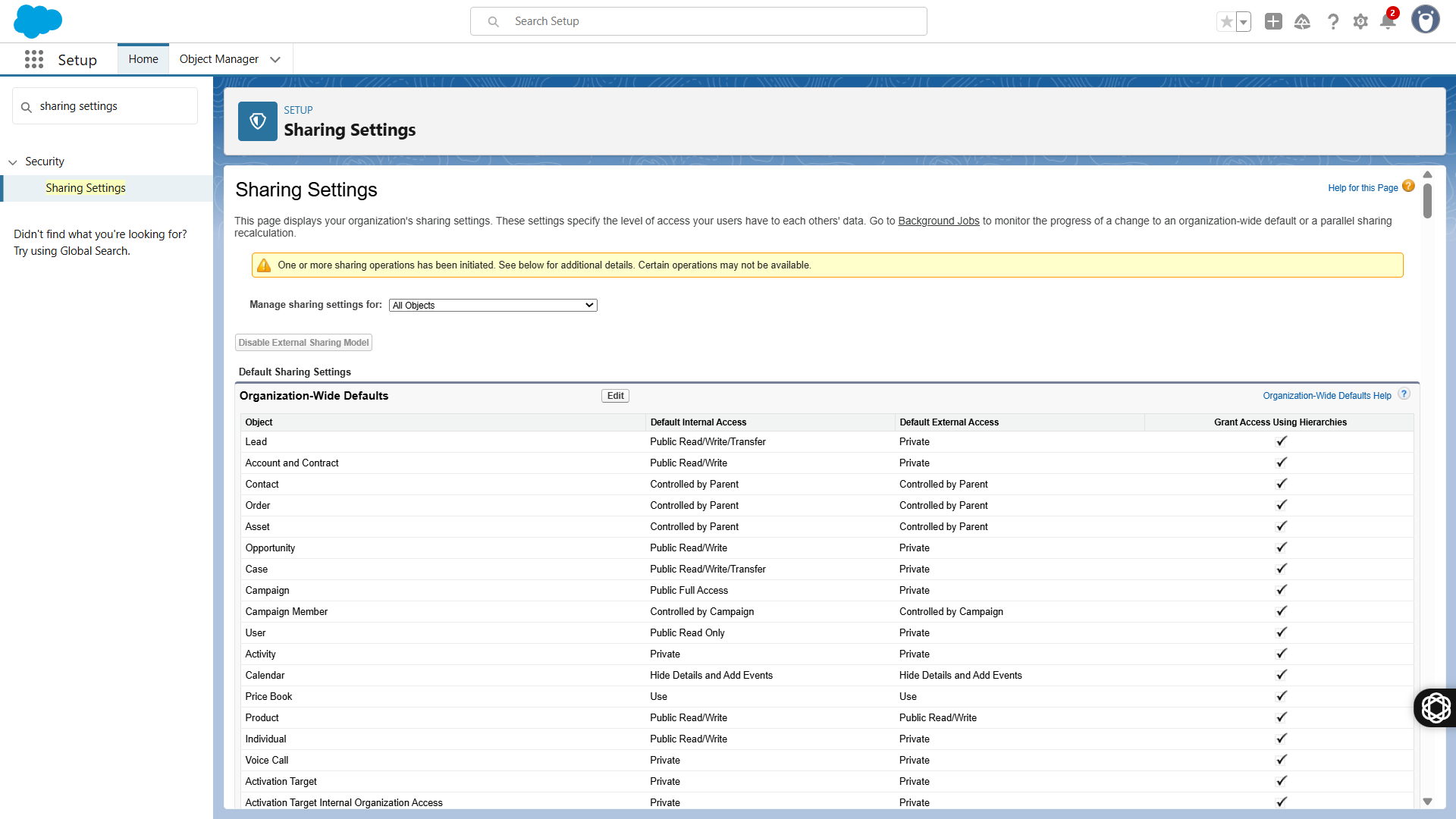
Tasks:

REST Apex Class → /bookFlight (accept JSON: passenger + flight → create booking).

Platform Events → Flight Delay Alerts (subscribed by Agents).

Mock External Payment → callout stub returning success/failure.

✅ Output: Simple integration demo with REST + Platform Events.



📌 Phase 8: Data Management

Goal: Populate with sample data.

Tasks:

Import 10 Flights, 20 Passengers (Contacts), 20 Bookings.

Setup Duplicate Rules for Passenger (Passport No must be unique).

Backup: create Unmanaged Package for re-deployment.

✅ Output: Working demo dataset.

📌 Phase 9: Reporting & Security

Reports:

Bookings by Flight.

Revenue by Flight.

Passenger Trends (e.g., top destinations).

Dashboards:

Flight Occupancy %

Monthly Revenue

Security:

Role Hierarchy → Admin > Airline Staff.

Enable Audit Trail for monitoring.

✅ Output: Dashboards + security model.

📌 Phase 10: Demo & Presentation

Goal: Deliver project like a real product.

Demo Flow:

Create Passenger.

Create Booking → Ticket auto-created.

Record Payment.

Refund request → Approval.

Show dashboards auto-updating.

Deliverables:

ERD Diagram.

Flow screenshots.

Apex Code repo (GitHub).

Dashboards (screenshots).

Demo Presentation (PPT).

✅ Output: Full working AMS demo in Salesforce Dev Org.