**Airlines Management System**

Category: Salesforce

**Team ID:** LTVIP2025TMID30238

**Team Details:**

|  |  |
| --- | --- |
| Name | Mail ID |
| Sai Prathap Reddy Pakanati | saiprathapreddypakanati@gmail.com |
| N Gopi Nayak | gopi9948231175@gmail.com |
| Medisetti Sasi Kumar | medisettisasikumar450@gmail.com |
| Nallamothu Sai Ganesh Vardhan | saiganeshvardhan65@gmail.com |

**1. INTRODUCTION**

1.1 Project Overview

This document outlines the development and implementation of an Airlines Management System built on the Salesforce platform. The primary goal of this project is to streamline and enhance the management of airline operations, specifically focusing on flight scheduling, booking processes, and customer/user management. The system leverages Salesforce's robust cloud capabilities, including custom objects, declarative automation (Flows), programmatic enhancements (Apex), and comprehensive reporting features, to provide an efficient and scalable solution for airline administration.

1.2 Purpose

The purpose of this documentation is to comprehensively detail the entire lifecycle of the Airlines Management System project. It aims to:Articulate the problem domain and the rationale behind choosing Salesforce as the development platform.Document the requirements, design, and architecture of the solution.Outline the project planning and execution phases, including key milestones and activities.Present the functional and performance testing methodologies and their results.Showcase the implemented solution through output screenshots.Discuss the advantages and disadvantages encountered during development and with the final system.Provide insights into potential future enhancements for the system. This document serves as a valuable reference for project stakeholders, future developers, and for academic evaluation.

**2. IDEATION PHASE**

2.1 Problem Statement

Prior to this project, [Describe the existing problems or inefficiencies in airline management that this system aims to solve. For example, manual flight scheduling leading to errors, fragmented booking data, difficulty in tracking passenger information, lack of centralized reporting, etc.] This led to [Mention negative impacts like: operational delays, customer dissatisfaction, increased administrative overhead, inaccurate data for decision-making]. The need for a unified, scalable, and efficient system became evident to address these challenges.

2.2 Empathy Map Canvas

An empathy mapping exercise was conducted to deeply understand the various personas interacting with an airline management system. Key personas identified include:

Airline Administrator:

Says: "I need to quickly see all flight details." "How do I update a flight's status efficiently?" "Generating reports takes too long."

Thinks: "Is this system intuitive enough for daily use?" "Can I customize views easily?" "How secure is the passenger data?"

Does: Creates/updates flight records, manages bookings, generates reports, handles user permissions.

Feels: Frustrated by manual data entry; overwhelmed by disparate systems; relieved when data is accurate and accessible.

Pains: Data silos, manual reconciliation, slow reporting, complex user management.

Gains: Centralized data, automated processes, quick reporting, simplified user access control.

Booking Agent:

Says: "Is there a seat available on this flight?" "I need to change a passenger's booking." "How do I process a refund quickly?"

Thinks: "Is this booking process error-prone?" "Can I find all customer interactions in one place?"

Does: Searches for flights, creates bookings, modifies reservations, cancels tickets, handles customer inquiries.

Feels: Stressed by complex booking interfaces; satisfied when a customer's issue is resolved efficiently.

Pains: Slow system response, difficulty in modifying bookings, lack of real-time availability.

Gains: Real-time booking data, efficient booking and modification processes, integrated customer view.

Passenger:

Says: "I want to check my flight status." "How do I change my seat?" "Where is my booking confirmation?"

Thinks: "Is my personal information safe?" "Can I manage my own booking online?"

Does: Views flight status (hypothetically, if a customer portal existed), receives booking confirmations.

Feels: Anxious about travel disruptions; relieved by clear communication and easy access to information.

Pains: Lack of self-service options, delayed notifications, difficulty in finding flight information.

Gains: Easy access to flight and booking information.

2.3 Brainstorming

Initial brainstorming sessions involved exploring various solutions to the identified problems. This included [Describe brainstorming techniques used, e.g., SWOT analysis, mind mapping, discussions with potential users]. Key ideas generated revolved around creating custom objects for Flights and Bookings, automating status updates, centralizing customer data, and providing intuitive interfaces for administrators and agents. Salesforce was chosen as the platform due to its rapid development capabilities, cloud-native architecture, security features, and extensive ecosystem for customization and integration.

**3. REQUIREMENT ANALYSIS**

3.1 Customer Journey Map

A key customer journey mapped was for the Booking Agent processing a customer booking and inquiry:

Search Flight Availability: Agent logs into Salesforce, navigates to the custom "Flight" tab. Searches for flights based on origin, destination, and dates.

System Interaction: Custom Flight Object data displayed.

Select Flight & Initiate Booking: Agent selects a suitable flight from the search results.

System Interaction: Agent clicks to initiate new Booking record.

Enter Passenger & Booking Details: Agent enters passenger information (name, contact) and booking details (seat number, special requests).

System Interaction: Data captured in custom "Booking Object" with lookup to "Flight Object." This might involve a Screen Flow for guided data entry.

Confirm Booking & Generate Confirmation: Agent confirms booking. System automatically generates a booking confirmation [and potentially triggers an email/message to the passenger if implemented].

System Interaction: Booking record created. Apex/Flow automation for confirmation.

Handle Inquiry/Modification: Customer calls to change departure date or query booking details.

System Interaction: Agent searches for Booking record, modifies "Departure Date" field. A Flow or Apex Trigger might update related records or perform validation.

View Flight/Booking Reports: Agent/Administrator accesses reports to view daily bookings or flight manifests.

System Interaction: Standard Salesforce Reports and Dashboards display aggregated data from Flight and Booking Objects.

3.2 Solution Requirement

The following are the key functional and non-functional requirements for the Airlines Management System:

Functional Requirements:

Flight Management:

Ability to create, read, update, and delete (CRUD) "Flight" records.

Each Flight record must include fields such as Flight Number, Origin, Destination, Departure Date/Time, Arrival Date/Time, Capacity, and Status.

Ability to relate flights to bookings.

Booking Management:

Ability to CRUD "Booking" records.

Each Booking record must include fields for Passenger Name, Contact Information, Booking Date, Departure Date, Status (e.g., Confirmed, Cancelled, Pending - implemented via Picklist Field), and a lookup relationship to the associated Flight.

User Interface:

Creation of custom tabs for "Flight" and "Booking" objects for easy navigation.

Development of a custom Lightning App to consolidate all relevant components and streamline user experience.

Data Structure & Relationships:

Establishment of lookup relationships between "Flight" and "Booking" objects.

Utilization of Salesforce Schema Builder for visual representation and management of object relationships.

User & Access Management:

Milestone 6 - User Adoption: Ability to create and manage "Passenger" records, representing a customer associated with bookings.

Milestone 7 - Profiles: Define various user profiles with appropriate access levels:

General Admin Profile

Management Admin Profile

Senior Admin Profile

Crew Member Profile

Milestone 8 - Role: Establish an "Admin Manager Role" to define data hierarchy and visibility.

Milestone 9 - Users: Ability to create and manage individual user accounts assigned to specific profiles and roles.

Reporting & Analytics:

Milestone 10 - Reports: Ability to generate custom reports (e.g., Flight Manifests, Booking Summaries by date/flight, Passenger Lists).

Milestone 11 - Dashboards: Creation of interactive dashboards to visualize key operational metrics (e.g., total bookings, flight capacity utilization).

Automation & Custom Logic:

Milestone 12 - Apex:

Develop Apex Classes for complex business logic (e.g., flight capacity calculation, advanced validation).

Implement Apex Triggers to automate actions upon record changes (e.g., updating flight status when all seats are booked, synchronizing related data).

Write Apex Test Classes to ensure code coverage and reliability.

Milestone 13 - Flows:

Create Screen Flows for guided user experiences (e.g., a multi-step booking process on the "Booking Object").

Develop Record-Triggered Flows to automate actions based on record creation/update (e.g., automatically creating a task when a booking status changes, updating a field on a related object).

Design Success Screen Elements within Flows to provide clear feedback to users.

Non-Functional Requirements:

Performance: The system should respond to user requests within [X] seconds for common operations.

Security: Adherence to Salesforce's robust security model, ensuring data privacy and integrity through profiles, roles, and object/field-level security.

Scalability: The system must be capable of handling an increasing number of flights, bookings, and users without significant performance degradation.

Usability: The Lightning Experience interface should be intuitive and user-friendly for all defined personas.

Maintainability: The solution should be well-documented and designed for ease of future modifications and enhancements.

3.3 Data Flow Diagram

[Insert Data Flow Diagram Here. This diagram should visually represent how data moves within your Salesforce Airlines Management System.

Example Structure for a Simple DFD:

Entities: Booking Agent, Salesforce System, Database (Salesforce Standard/Custom Objects)

Processes:

Flight Search & Selection

Booking Creation

Booking Modification

Report Generation

Data Stores: Flight Object Data, Booking Object Data, User Data

Flow Example:

Booking Agent requests Flight Search from Salesforce System.

Salesforce System queries Flight Object Data.

Flight Object Data returns results to Salesforce System.

Salesforce System displays results to Booking Agent.

Booking Agent inputs Booking Details to Salesforce System.

Salesforce System processes Booking Creation (potentially involving a Flow).

Booking Details are stored in Booking Object Data.

Booking Object Data is updated by Salesforce System when Booking Modification occurs.

Salesforce System generates Reports from Flight Object Data and Booking Object Data for Booking Agent/Administrator.]

3.4 Technology Stack

The Airlines Management System is exclusively developed on the Salesforce Platform, leveraging its comprehensive suite of tools and services. The core technologies utilized include:

Salesforce Cloud: Primarily utilizing the Salesforce platform capabilities (custom objects, standard features).

Custom Objects: Flight\_\_c and Booking\_\_c to manage core airline operational data.

Custom Fields & Relationships: Extensive use of various field types (Text, Number, Date/Time, Picklist, Lookup) to capture specific flight and booking attributes, and establish relationships between objects.

Lightning Experience (LEX): The modern user interface framework providing an intuitive and responsive experience.

Apex: Salesforce's proprietary programming language for building complex business logic, custom web services, and database interactions (e.g., FlightTrigger, BookingHandlerClass).

Salesforce Flow: Declarative automation tool used for guiding users through processes (Screen Flows) and automating record-triggered events (Record-Triggered Flows).

Reports & Dashboards: Built-in analytics tools for generating real-time operational insights.

Profiles & Permission Sets: For granular control over user access to objects, fields, and applications.

Roles: To establish a data hierarchy and define record visibility.

Custom Tabs & Lightning App: For organizing and presenting the system's functionalities within a cohesive user interface.

Schema Builder: For visual design and understanding of the data model.

**4. PROJECT DESIGN**

4.1 Problem Solution Fit

The design of the Airlines Management System directly addresses the problems identified in the ideation phase. By centralizing flight and booking data within Salesforce custom objects (Flight\_\_c and Booking\_\_c), we eliminated data silos and improved data accuracy. Automated processes via Apex Triggers and Salesforce Flows significantly reduced manual errors and administrative overhead associated with booking management and status updates. The intuitive Lightning App and custom tabs provide a unified interface, improving user efficiency and reducing the learning curve. Furthermore, Salesforce's robust reporting capabilities now offer real-time insights into flight operations and booking trends, enabling better decision-making.

4.2 Proposed Solution

The Airlines Management System is structured around a core data model comprising two main custom objects:

Flight Object (Flight\_\_c): This object stores all details related to individual flights. Key fields include:

Flight\_Number\_\_c (Text, Unique)

Origin\_\_c (Text)

Destination\_\_c (Text)

Departure\_Time\_\_c (DateTime)

Arrival\_Time\_\_c (DateTime)

Capacity\_\_c (Number)

Status\_\_c (Picklist: Scheduled, Departed, Arrived, Cancelled)

Available\_Seats\_\_c (Formula/Number, calculated based on Capacity and associated Bookings).

[Add any other relevant fields you created, e.g., Aircraft\_Type\_\_c, Gate\_Number\_\_c]

Booking Object (Booking\_\_c): This object captures all information pertaining to a passenger's booking. Key fields include:

Passenger\_Name\_\_c (Text)

Contact\_Email\_\_c (Email)

Booking\_Date\_\_c (Date)

Departure\_Date\_\_c (Date) - as per your image.

Status\_\_c (Picklist: Confirmed, Pending, Cancelled, Checked In) - as per your image.

Flight\_\_c (Lookup to Flight\_\_c): Establishes the crucial relationship between a booking and its associated flight.

[Add any other relevant fields you created, e.g., Seat\_Number\_\_c, Total\_Fare\_\_c]

Key Solution Components:

Custom Tabs: Dedicated "Flights" and "Bookings" tabs provide direct access to record lists.

Lightning App: The "Airlines Management System" Lightning App serves as the central hub, consolidating these tabs, reports, and dashboards for a seamless user experience.

Automations:

Apex Trigger on Booking: When a Booking\_\_c record is created or updated, an Apex Trigger BookingTrigger updates the Available\_Seats\_\_c count on the related Flight\_\_c record.

Screen Flow for New Booking: A Screen Flow assists booking agents in a step-by-step process for creating new Booking\_\_c records, ensuring all required fields are captured accurately and includes a "Success Screen Element."

Record-Triggered Flow: A flow automatically updates the Flight\_\_c status to "Departed" once the Departure\_Time\_\_c passes and a flight is confirmed.

Security Model: Implemented through custom Profiles (General Admin Profile, Management Admin Profile, Senior Admin Profile, Crew Member Profile) and an Admin Manager Role to control data visibility and functional access based on user responsibilities.

4.3 Solution Architecture

[Insert Solution Architecture Diagram Here. This diagram should visually depict the components of your Salesforce solution.

Example Structure for a Solution Architecture Diagram:

Central: Salesforce Cloud Platform

Core Objects:

Flight Object (with its key fields)

Booking Object (with its key fields)

Relationship line between Flight and Booking objects (Lookup).

User Interface Layer:

Lightning App

Custom Tabs (Flight, Booking)

Automation Layer:

Apex (Classes, Triggers)

Salesforce Flows (Screen Flow, Record-Triggered Flow)

Security Layer:

Profiles (list them)

Roles (list them)

Data & Analytics Layer:

Reports

Dashboards

Optional: External Integrations (if any, showing data flow in/out of Salesforce).

Explanation of the Diagram: The architecture is centered around the Salesforce Cloud Platform, hosting the custom Flight\_\_c and Booking\_\_c objects which form the core data model. Users interact with the system primarily through the Airlines Management System Lightning App, providing access to custom tabs and streamlined interfaces. Business logic and automation are handled by Apex (for complex, programmatic tasks like real-time seat availability updates) and Salesforce Flows (for guided user processes and declarative record automations). The robust security model ensures data integrity and appropriate access levels for different user roles and profiles. All operational data is available for analysis through comprehensive reports and interactive dashboards.]

**5. PROJECT PLANNING & SCHEDULING**

5.1 Project Planning

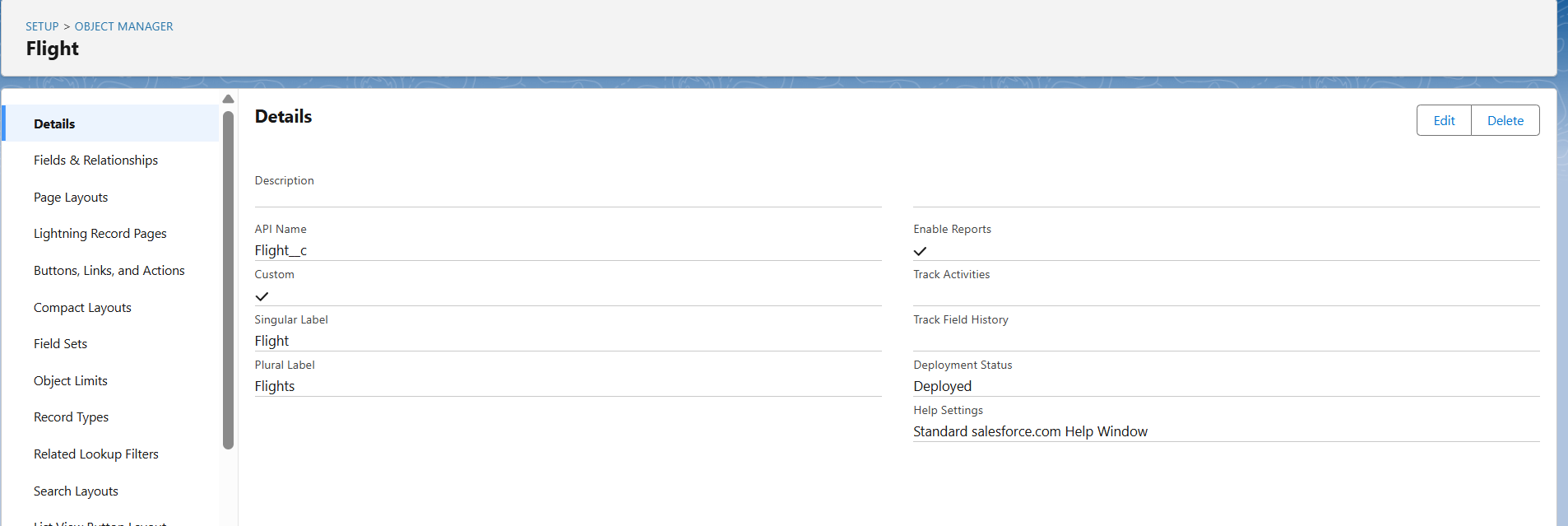
The project followed a [mention your methodology, e.g., Agile Scrum] methodology, with iterative development cycles focusing on delivering functional components incrementally. Key milestones and activities were tracked to ensure progress and timely completion.

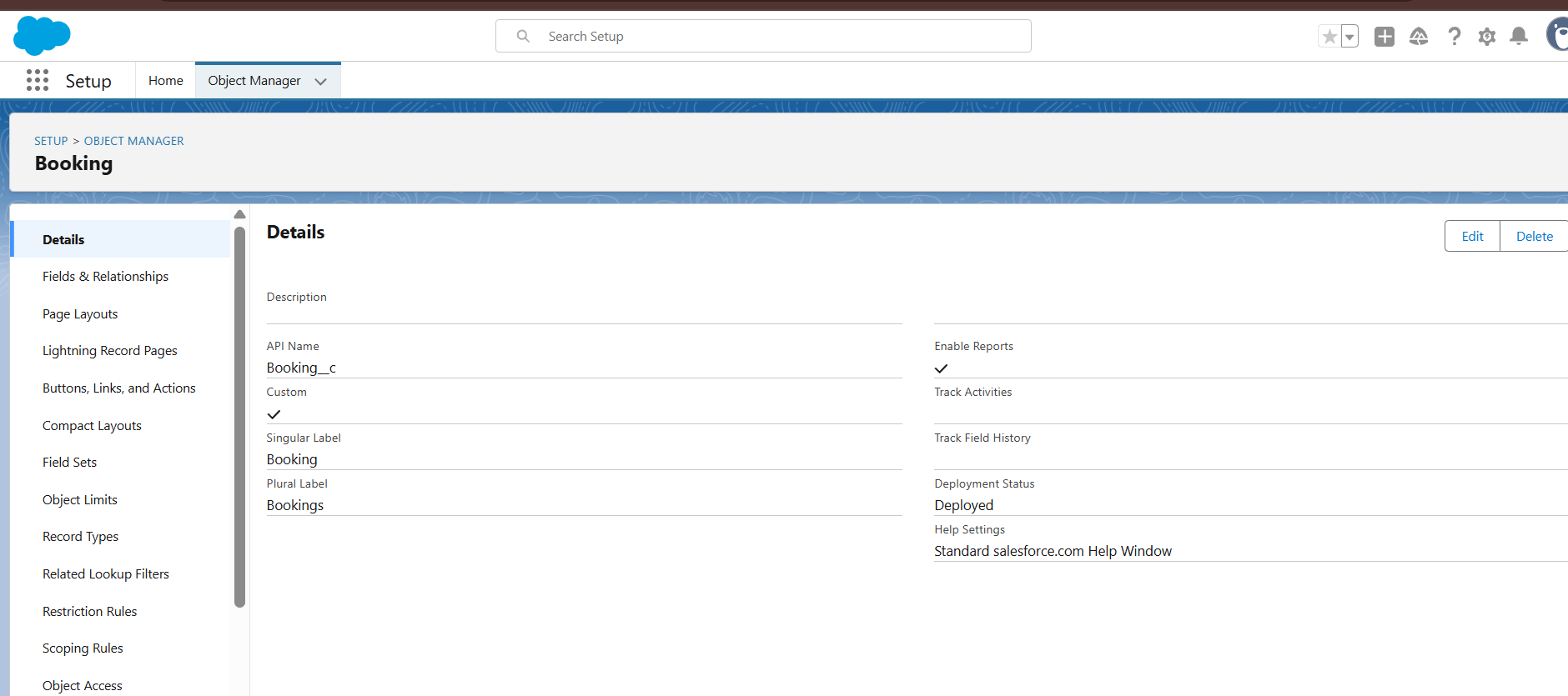
Key Milestones and Activities:

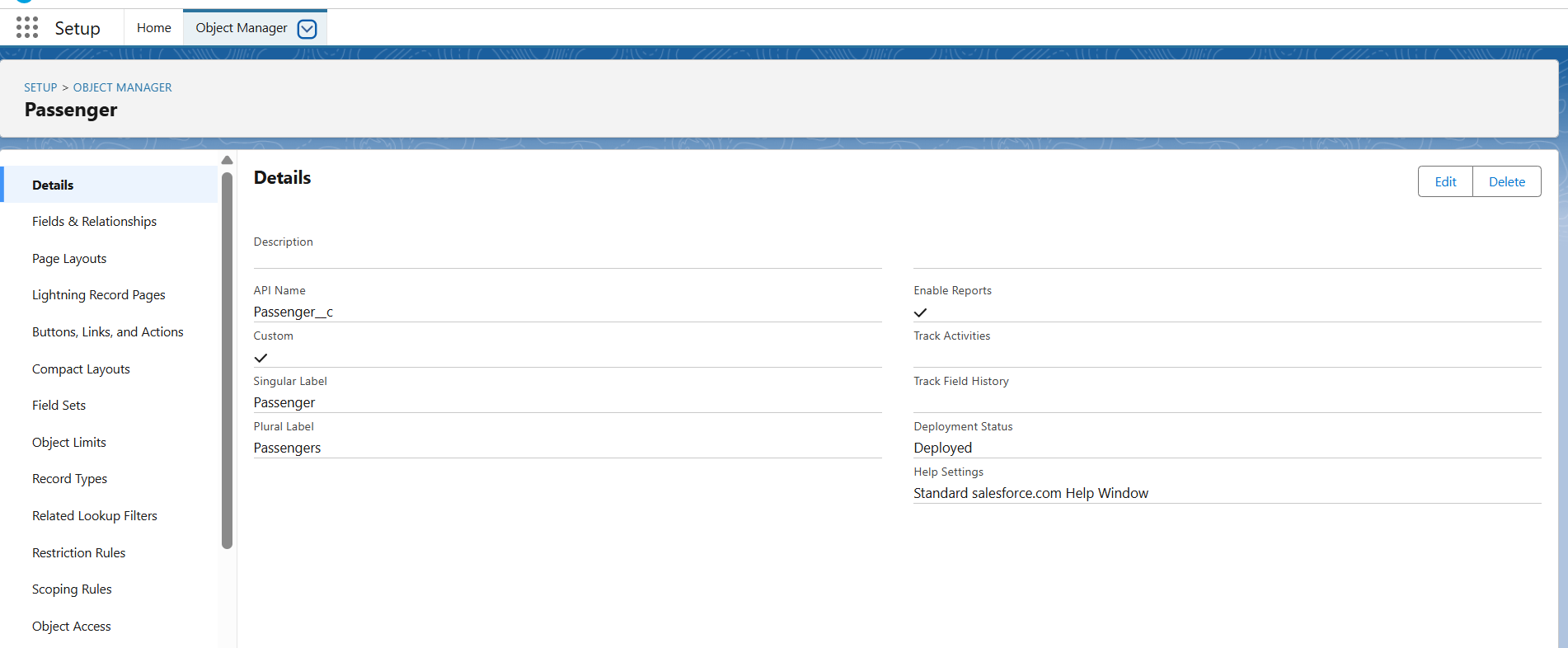
Initial Setup:

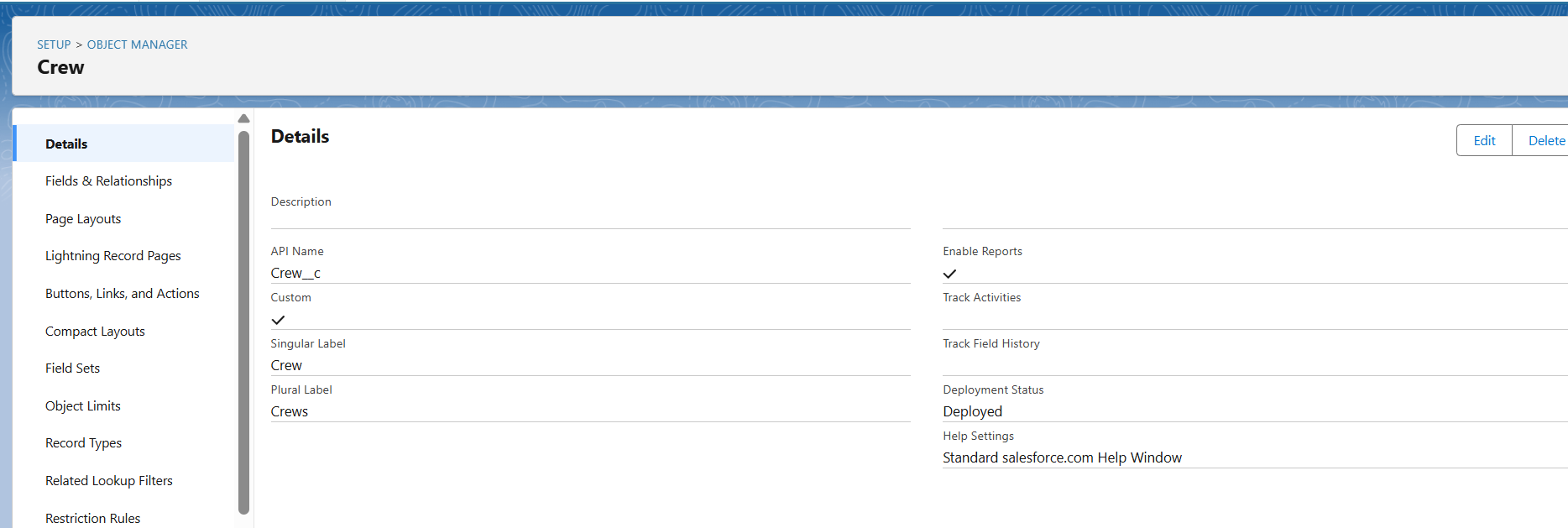
Salesforce Environment Setup

Object Creation:

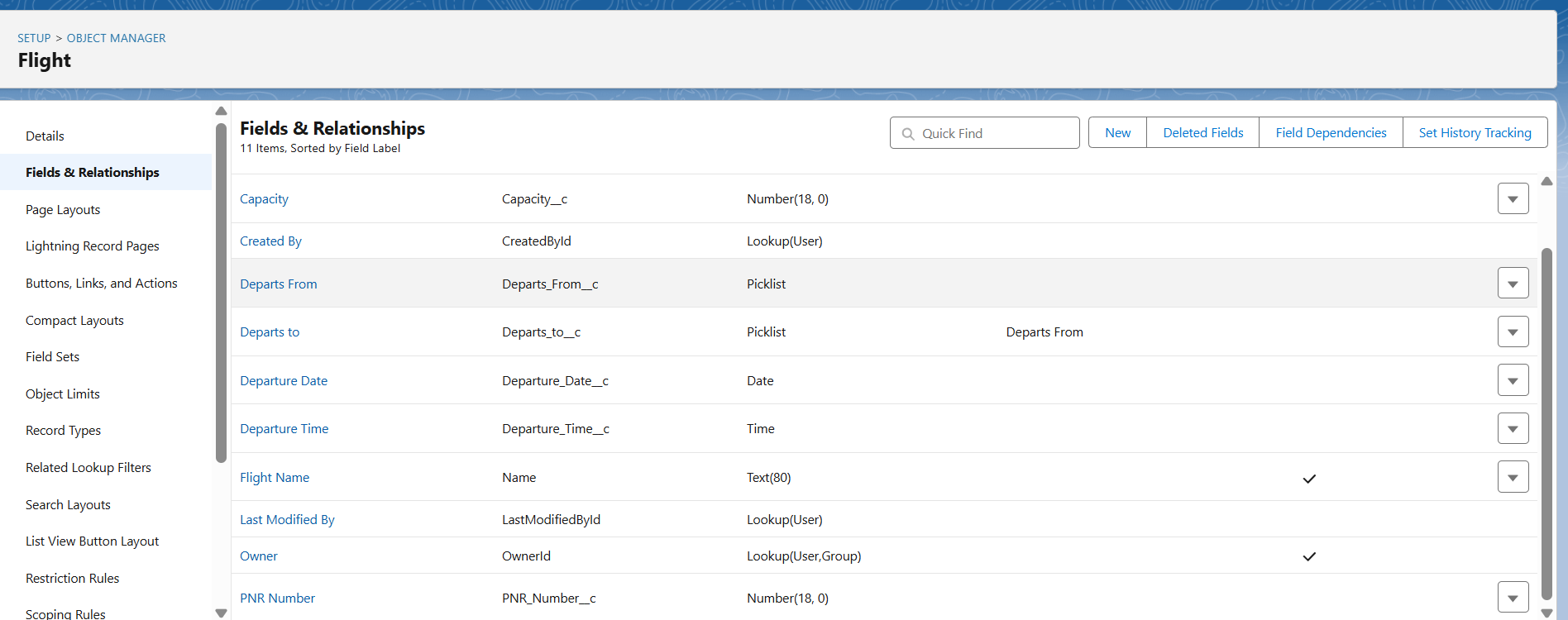


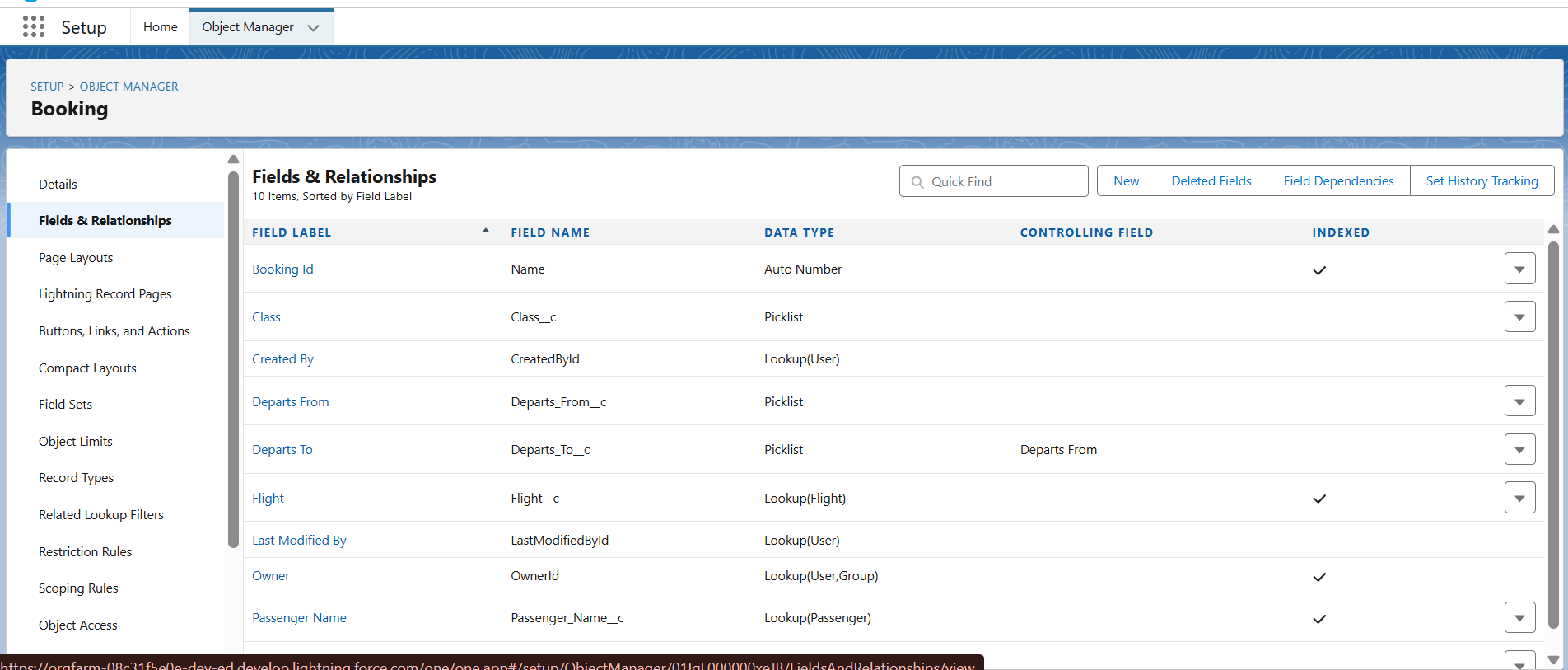


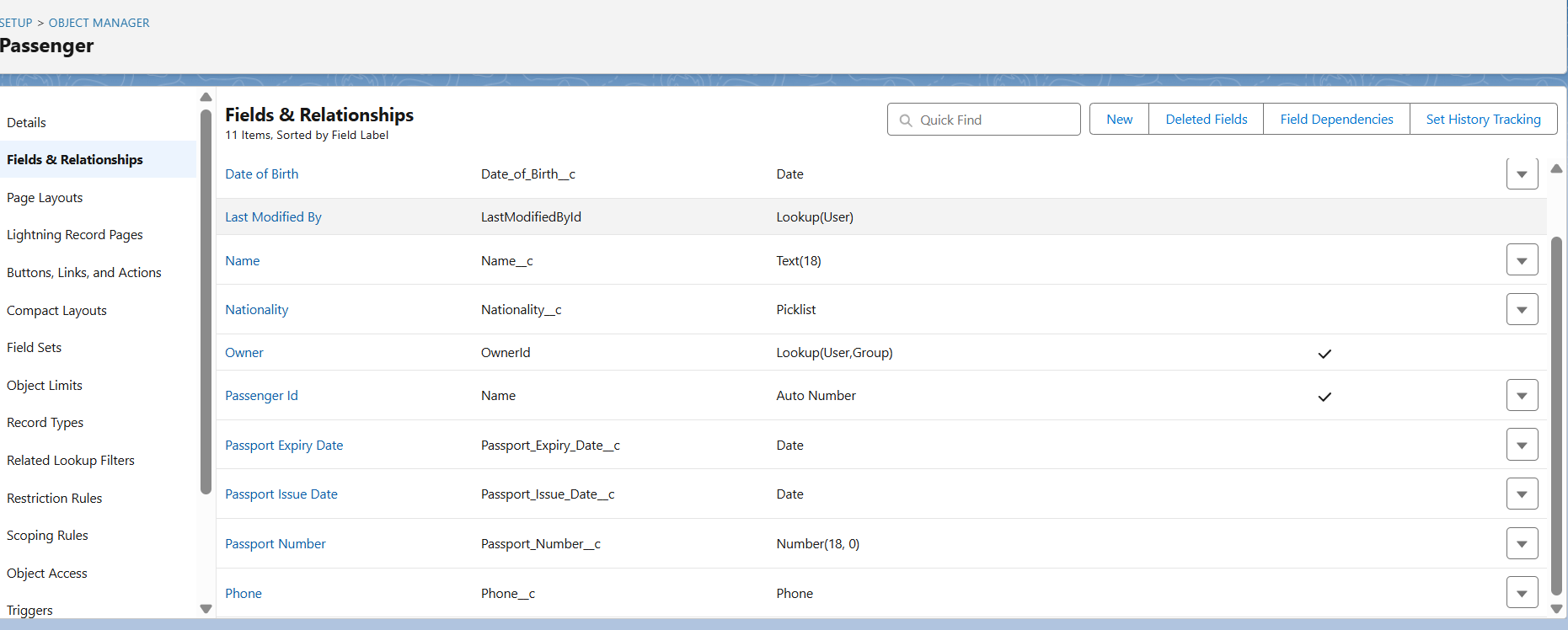


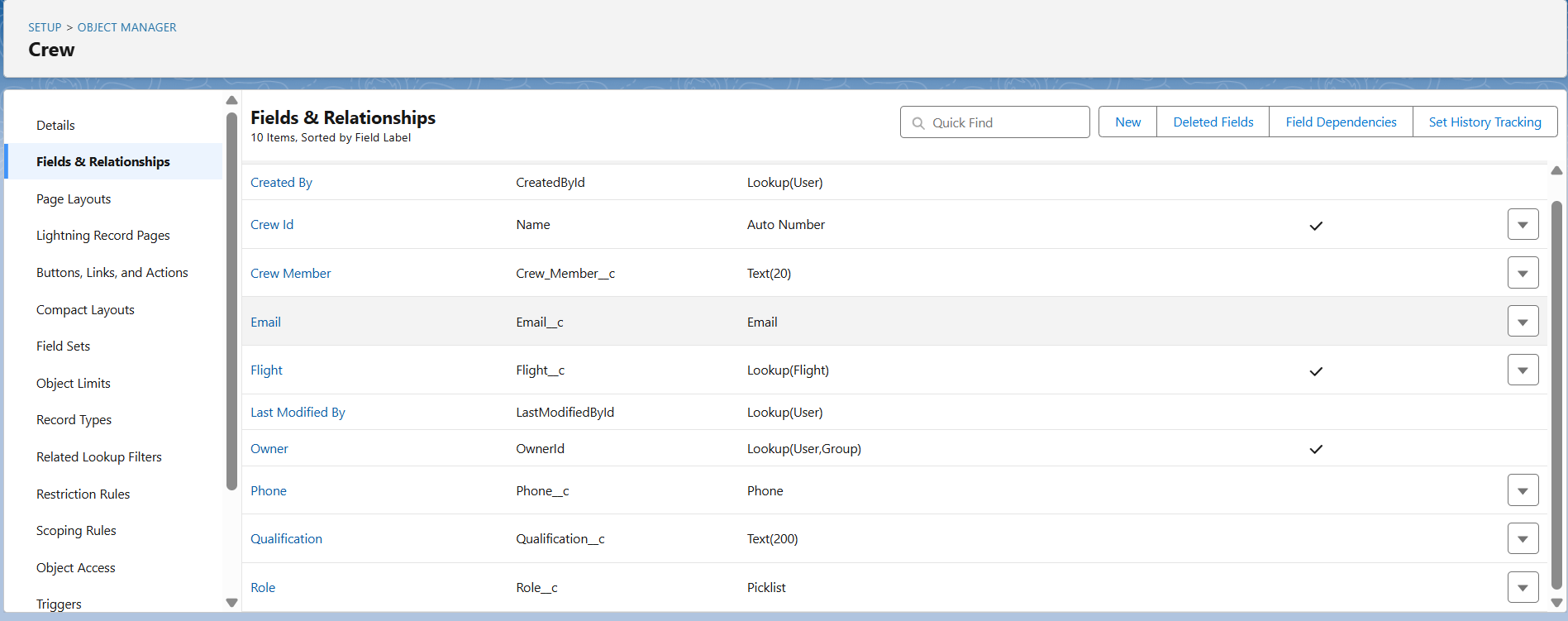


Tabs:

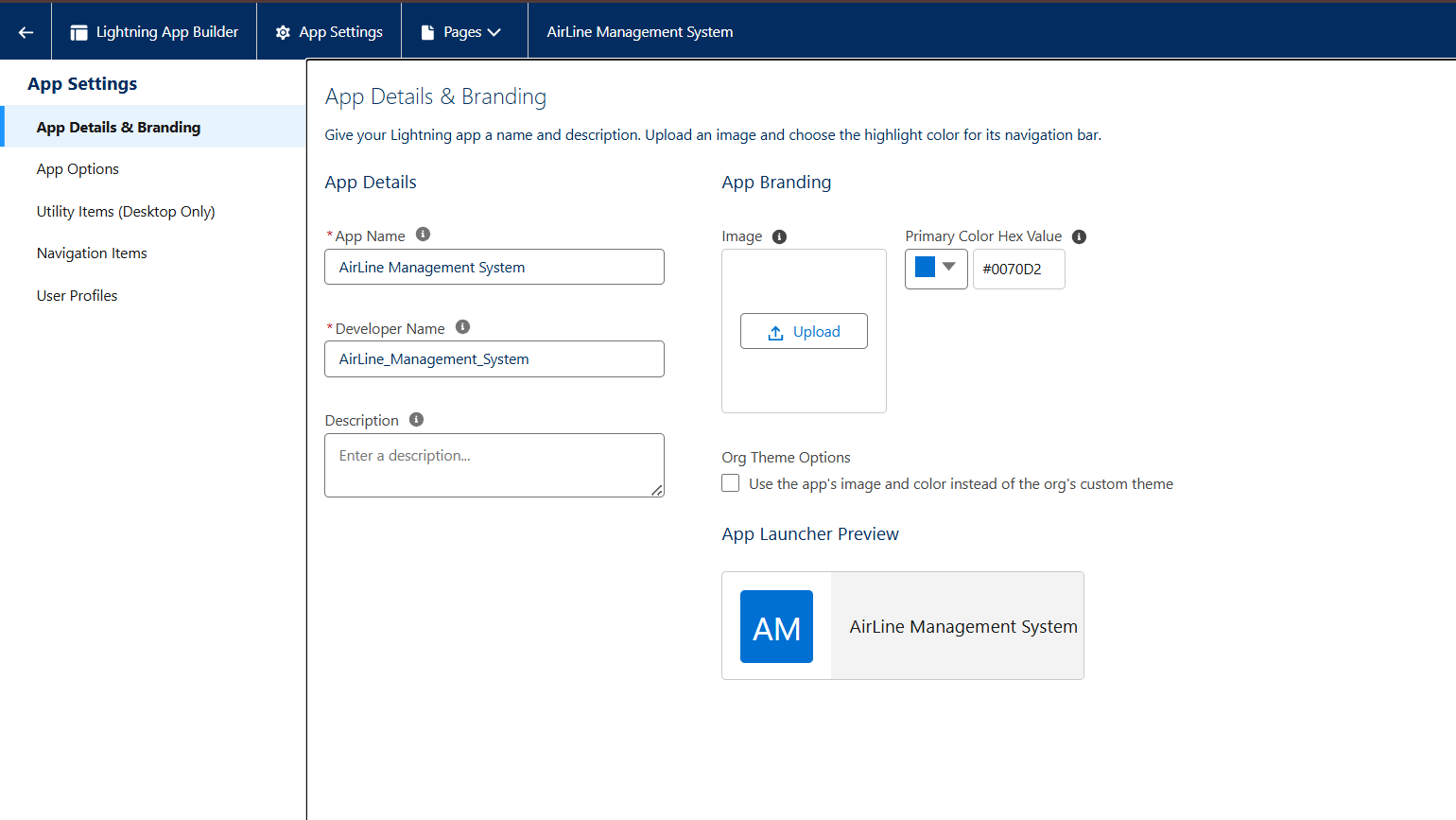








The Lightning App:

Data Model & Fields Development:

Fields:

Creating Field in Flight Object (e.g., Flight Number, Origin, Destination, Capacity)

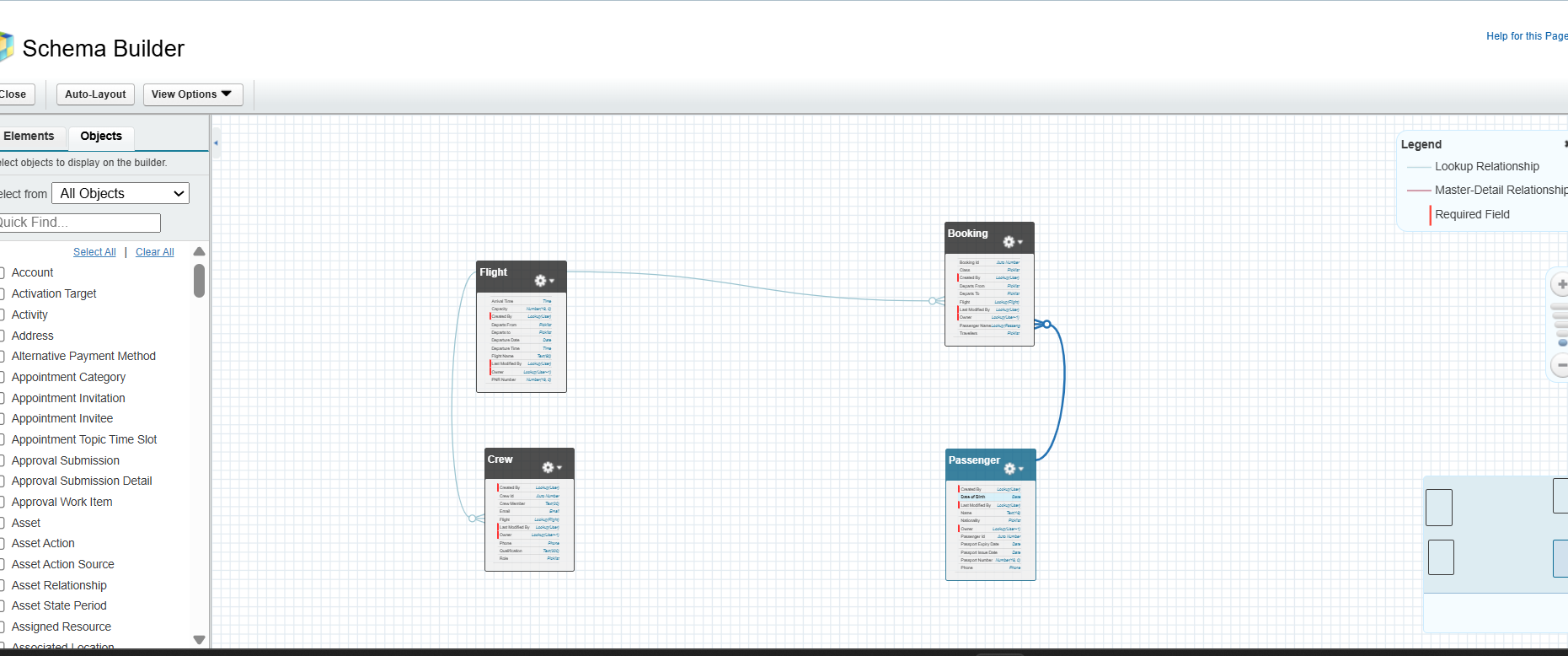
Creating Departure Date On Booking Object

Creating Picklist Field On Booking Object (e.g., Status for Booking)

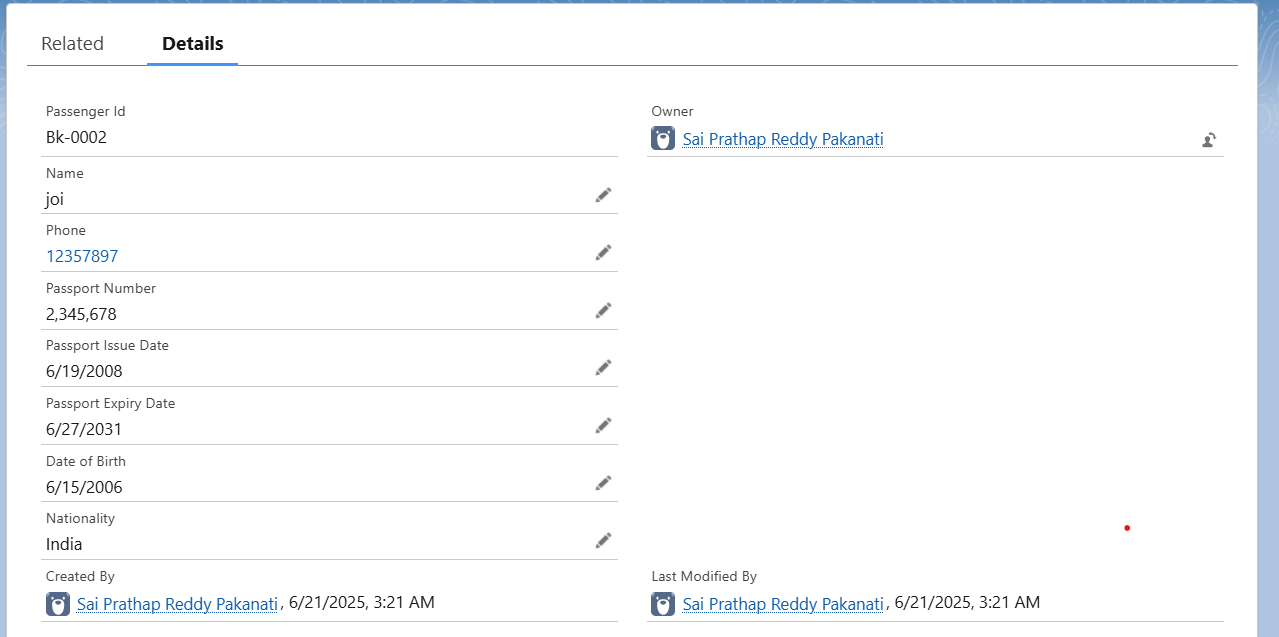
Creating Lookup Relationship Between Flight And Booking

Creating Remaining Fields (e.g., Passenger Name, Contact Email on Booking; Arrival Time, Status on Flight)

Schema Builder:

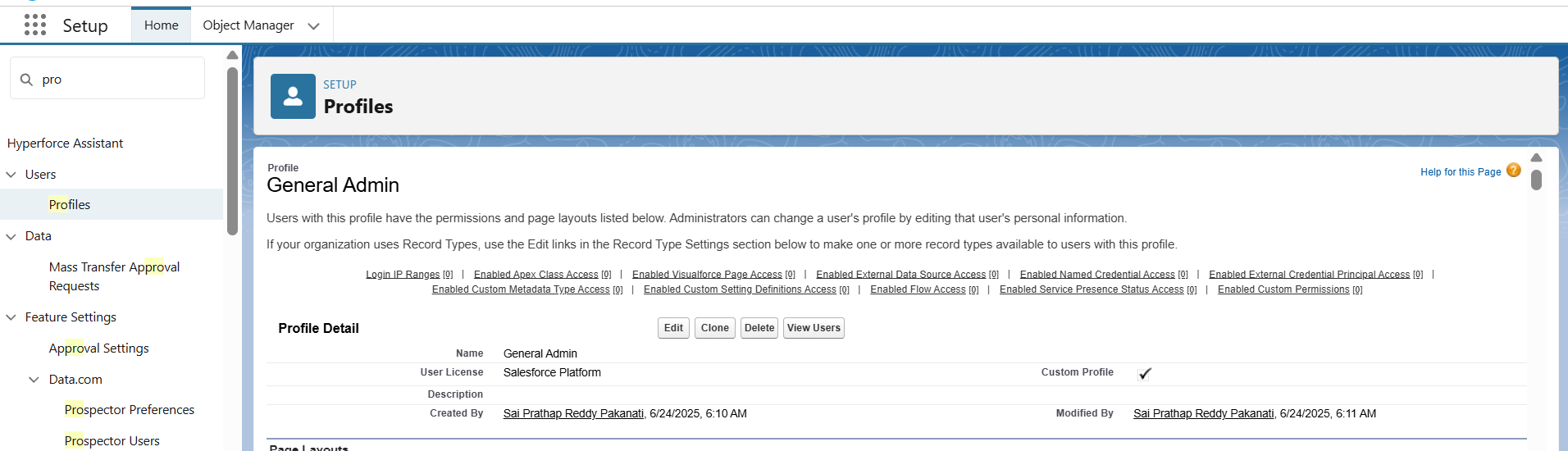


User Adoption & Management Implementation:

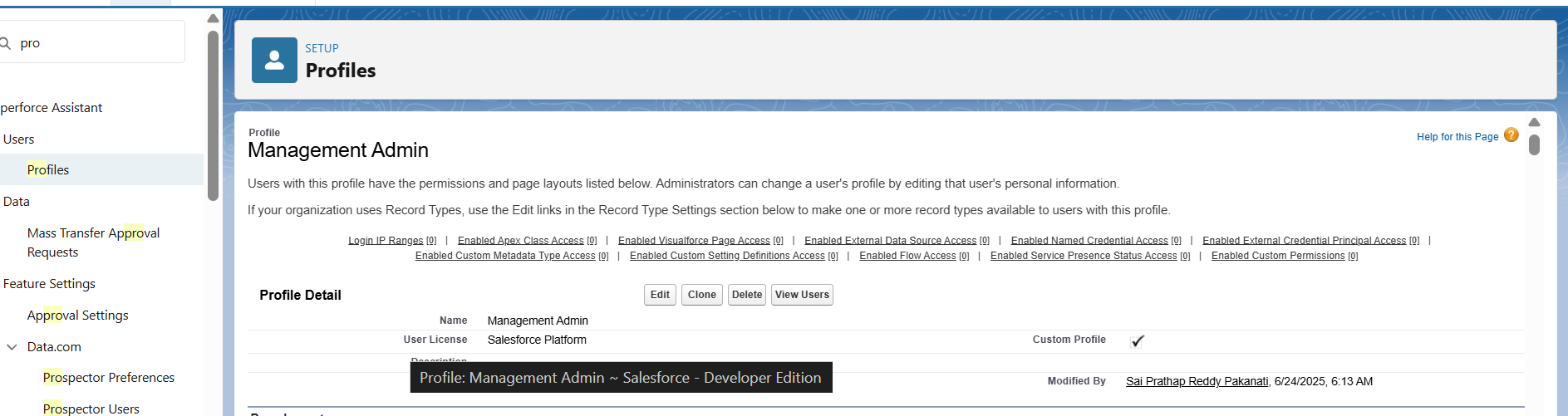


Milestone 7 - Profiles:

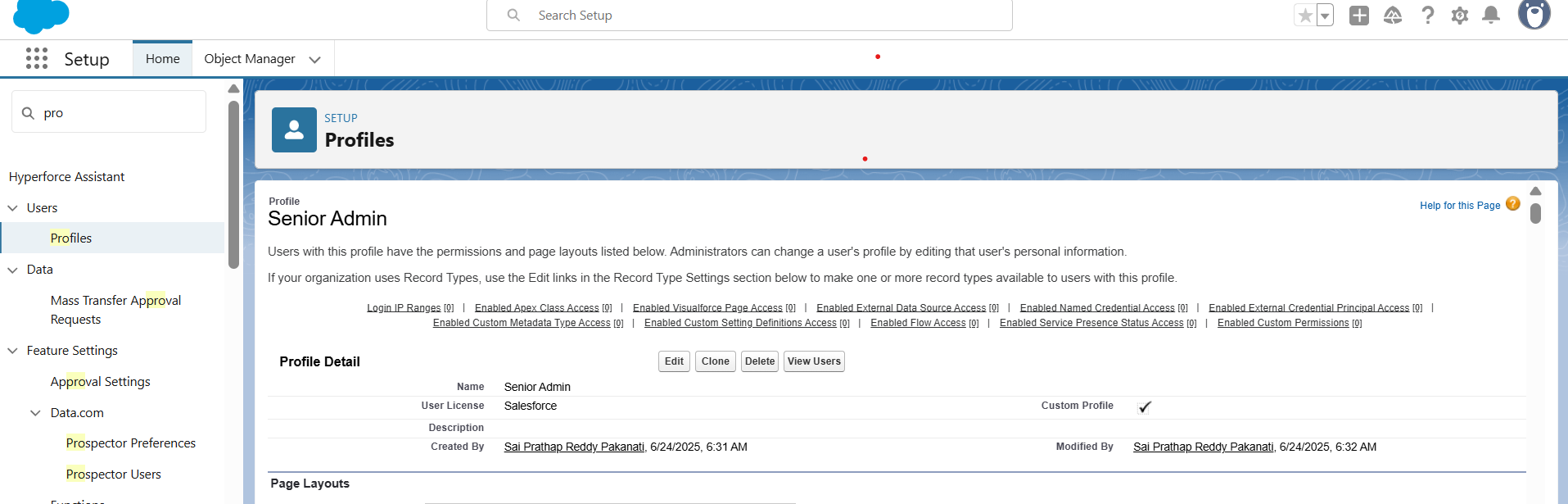
General Admin Profile Creation

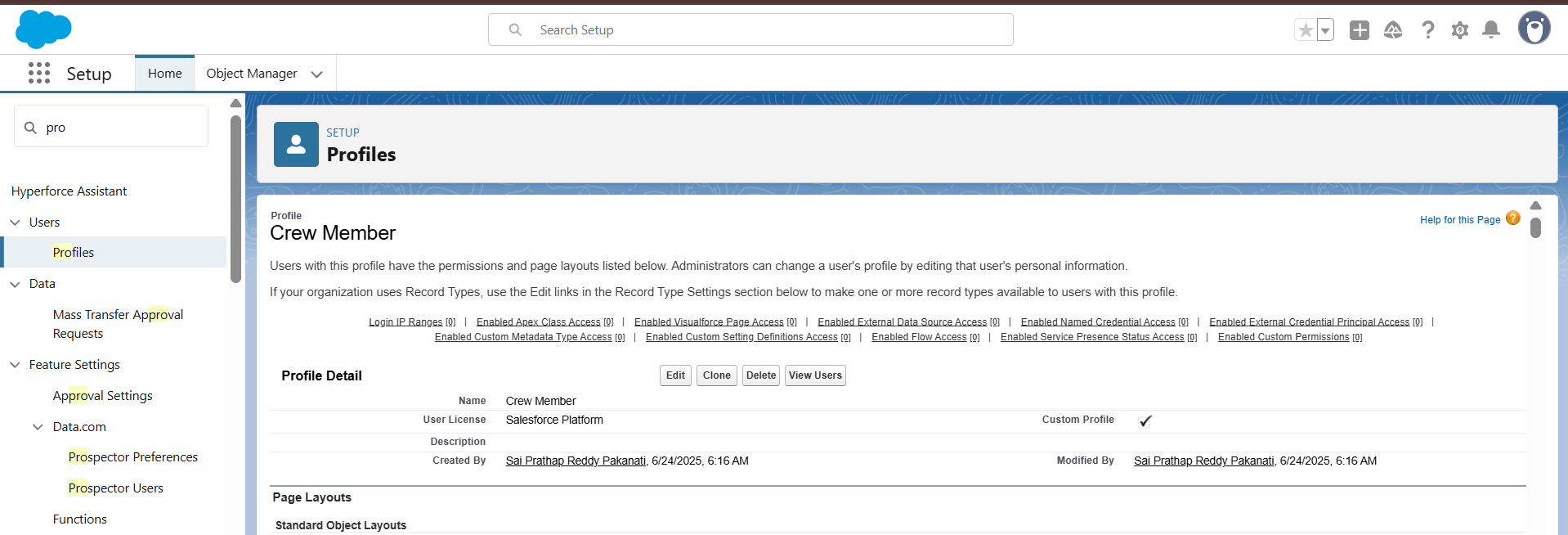


Management Admin Profile Creation

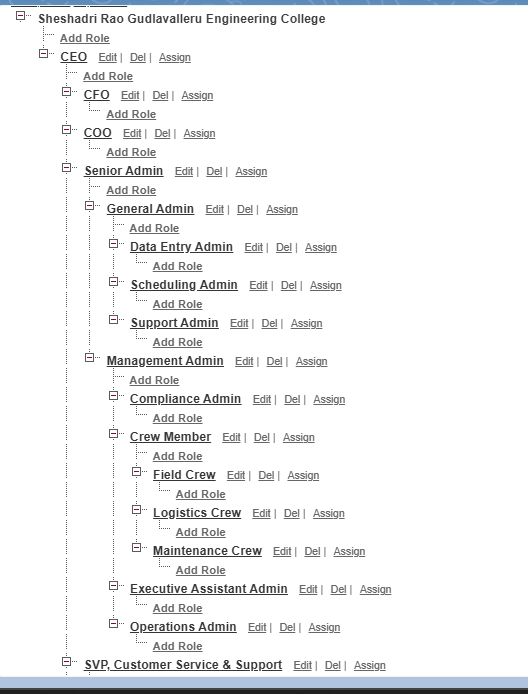


Create Senior Admin & Crew Member Profiles

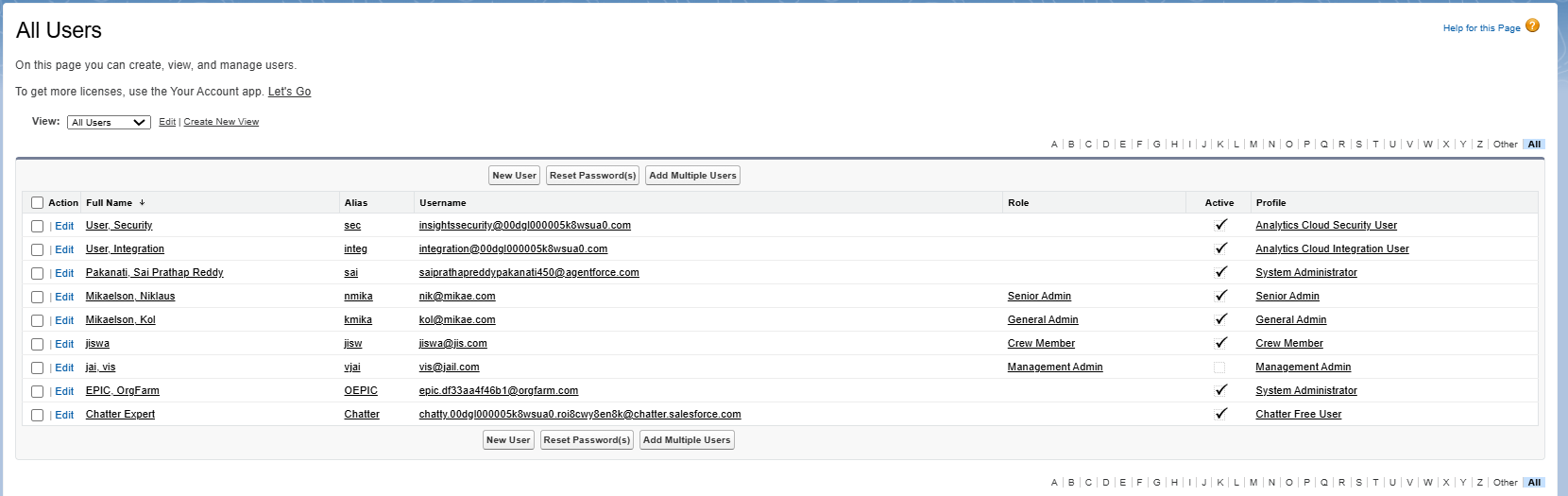




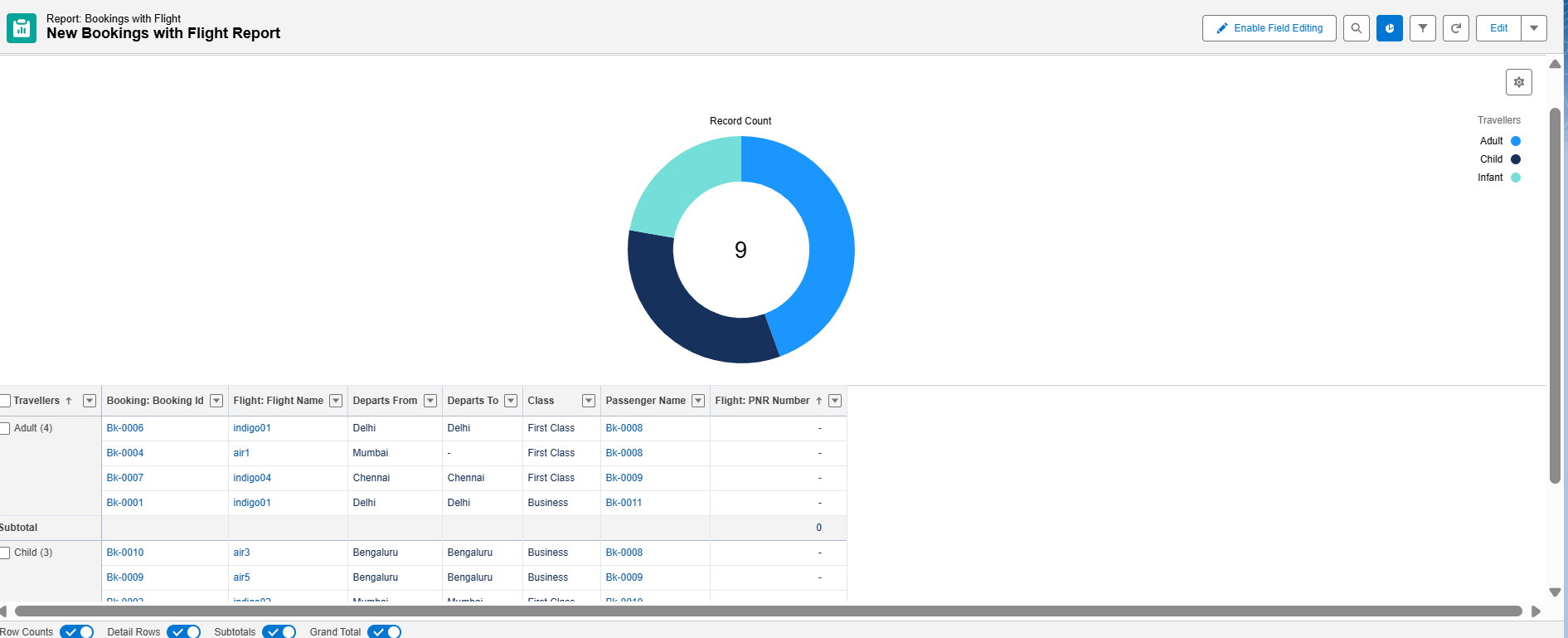
Milestone 8 - Role:

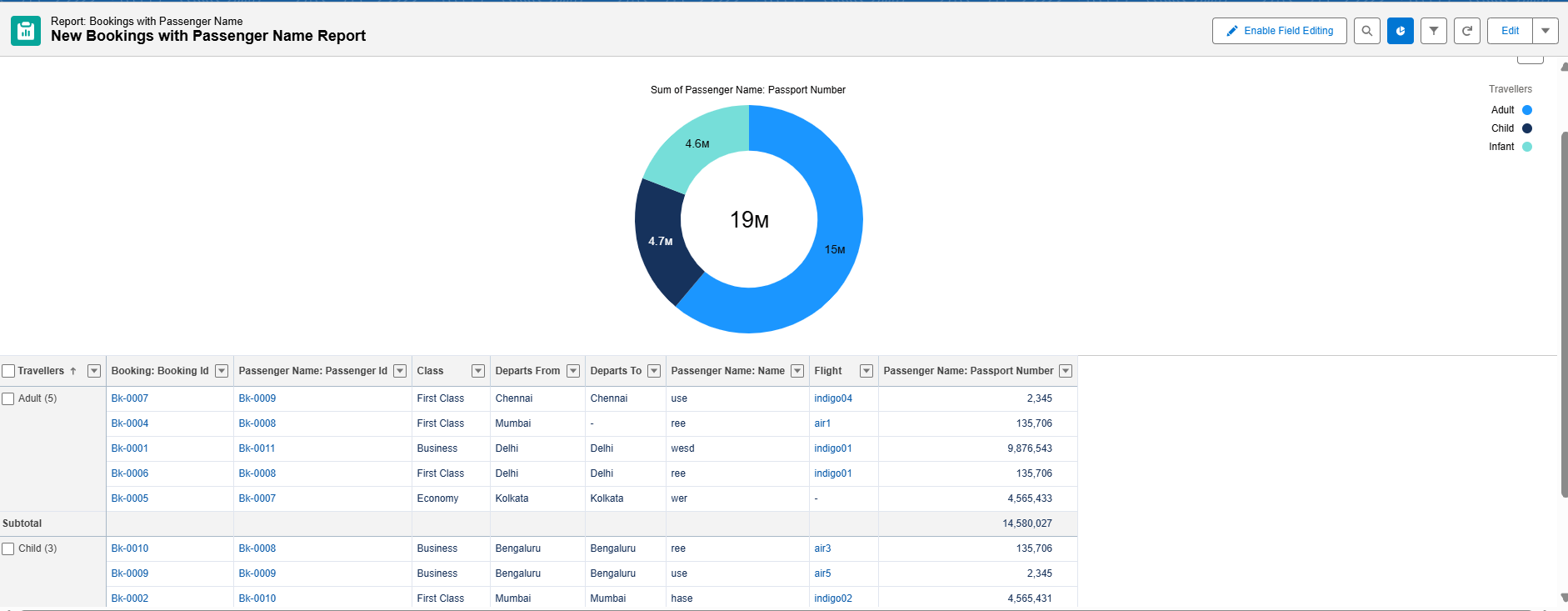


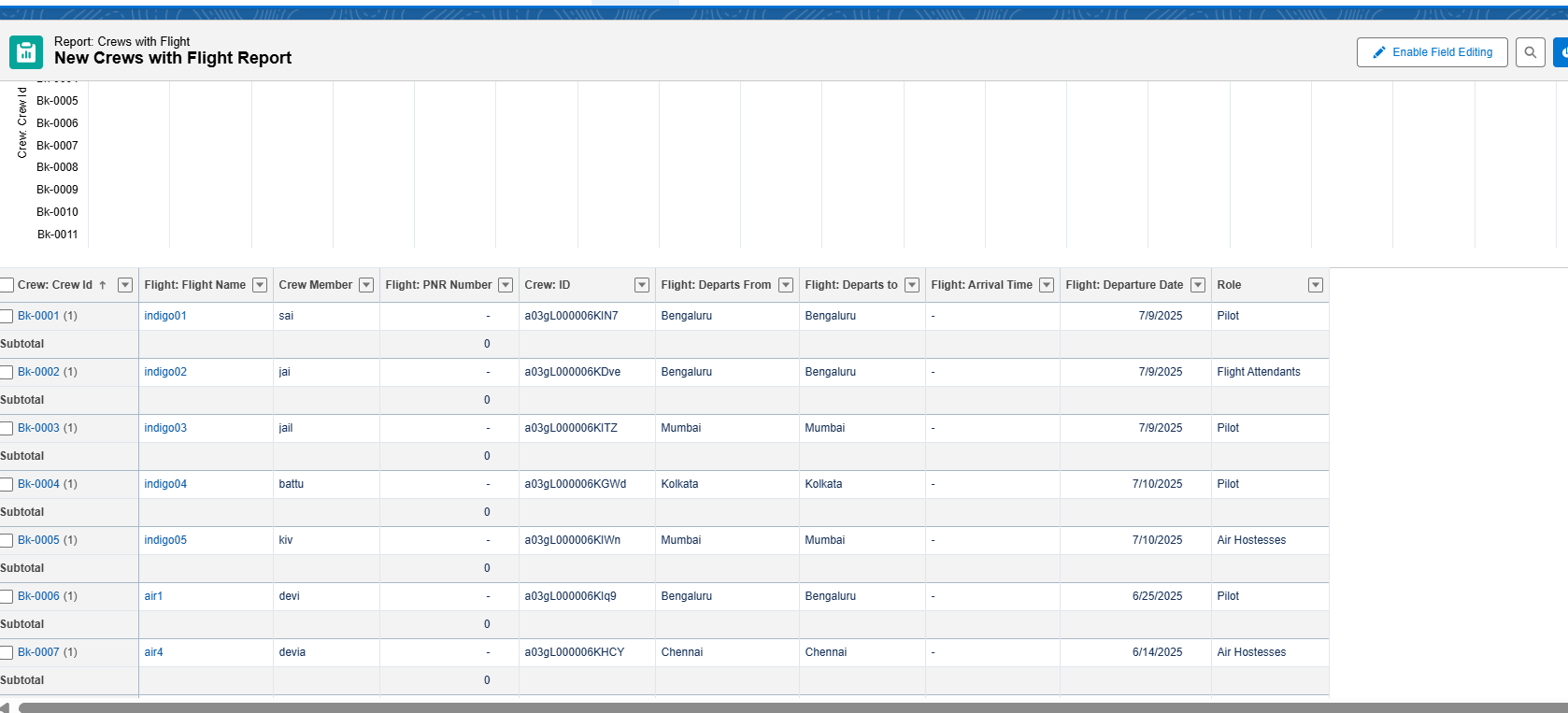
Milestone 9 - Users:

Reporting & Analytics Development:

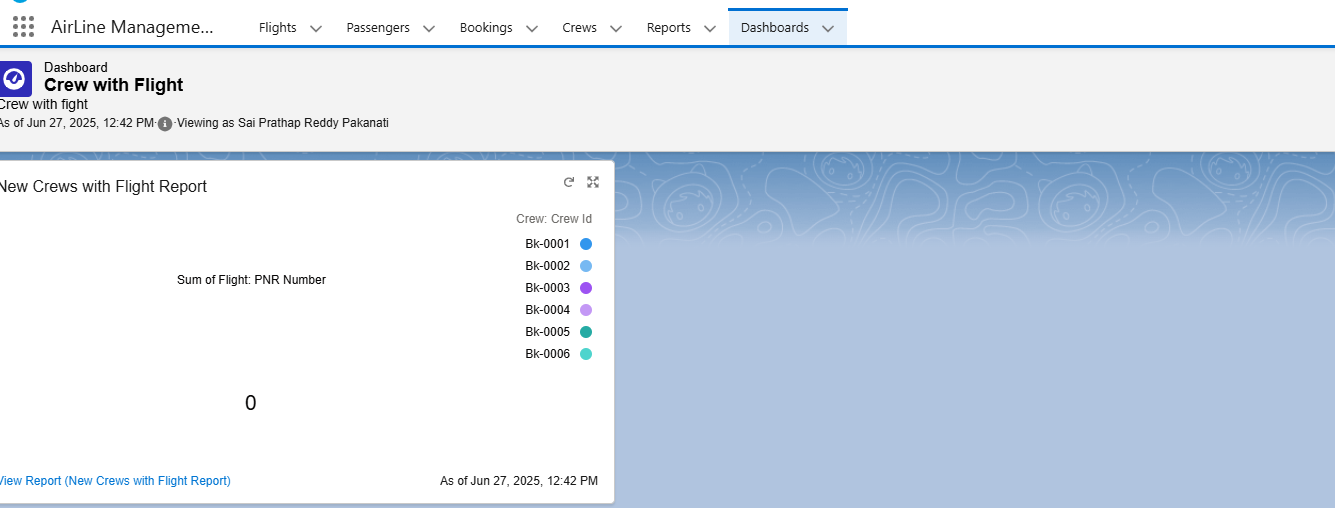
Milestone 10 - Reports:

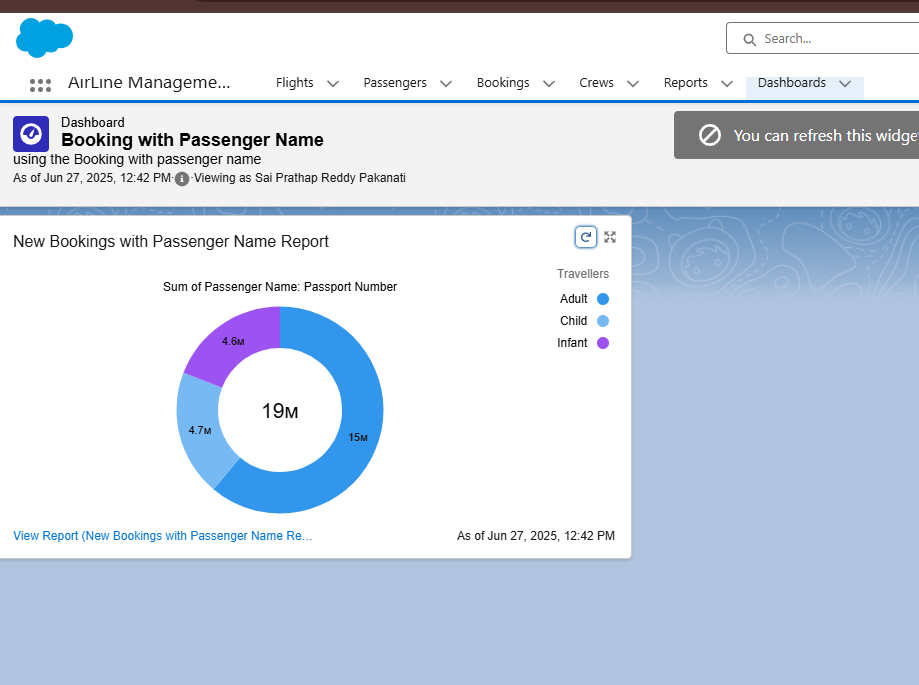






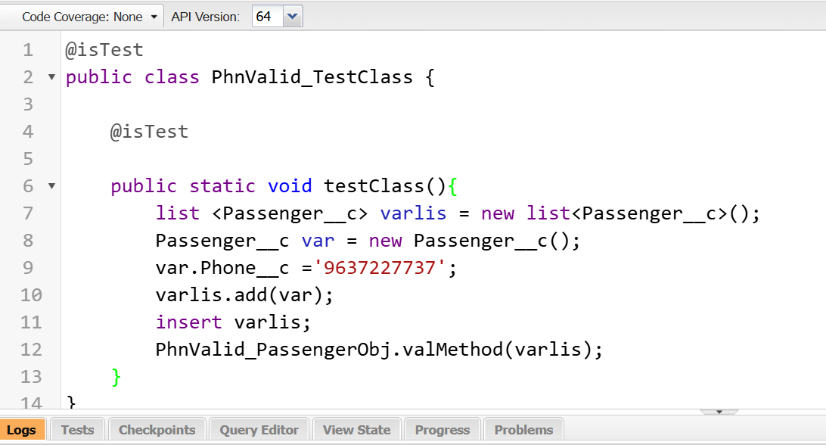
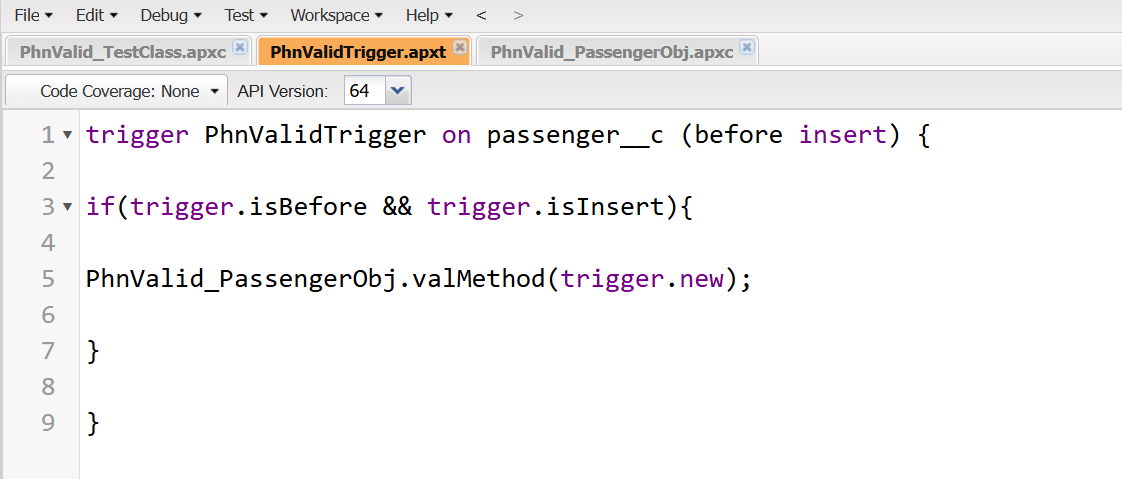
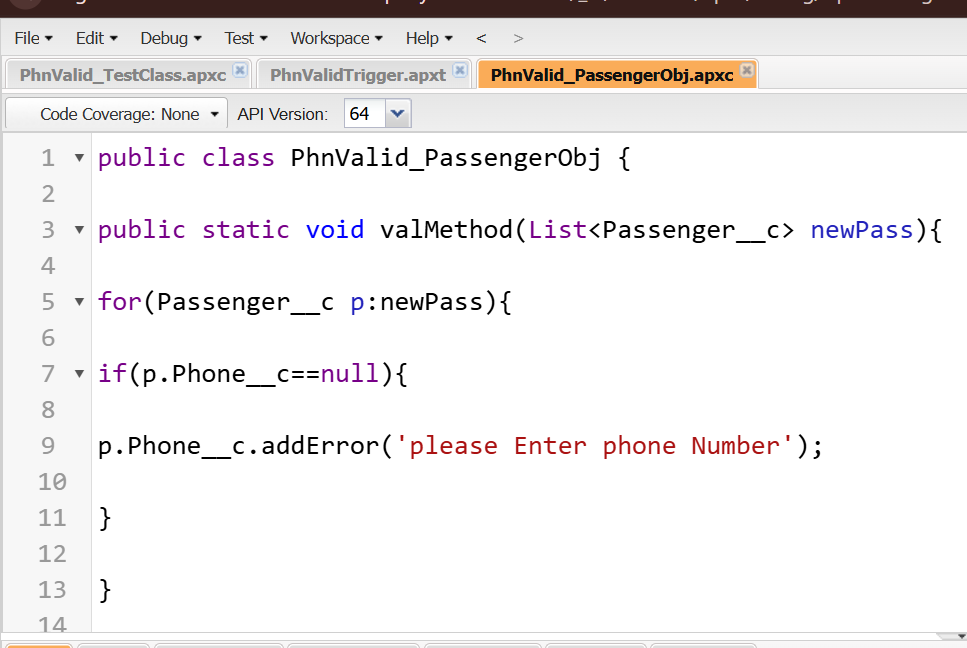
Milestone 11 - Dashboards:



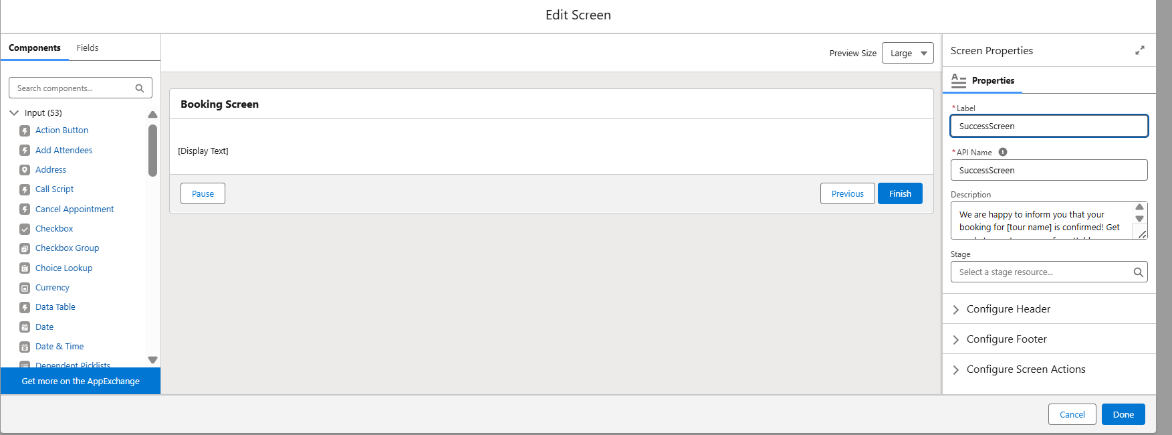
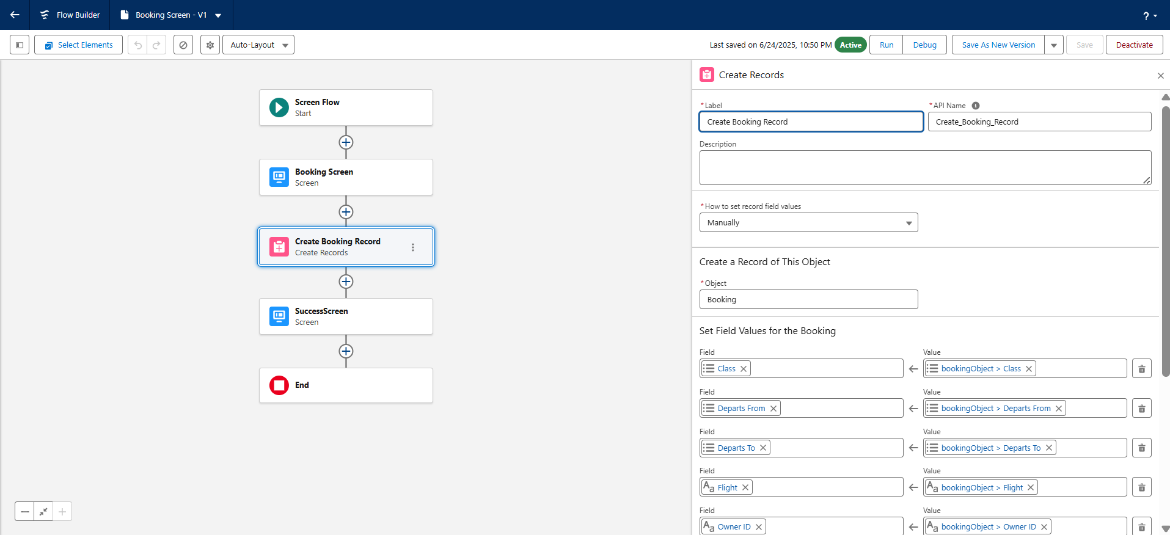
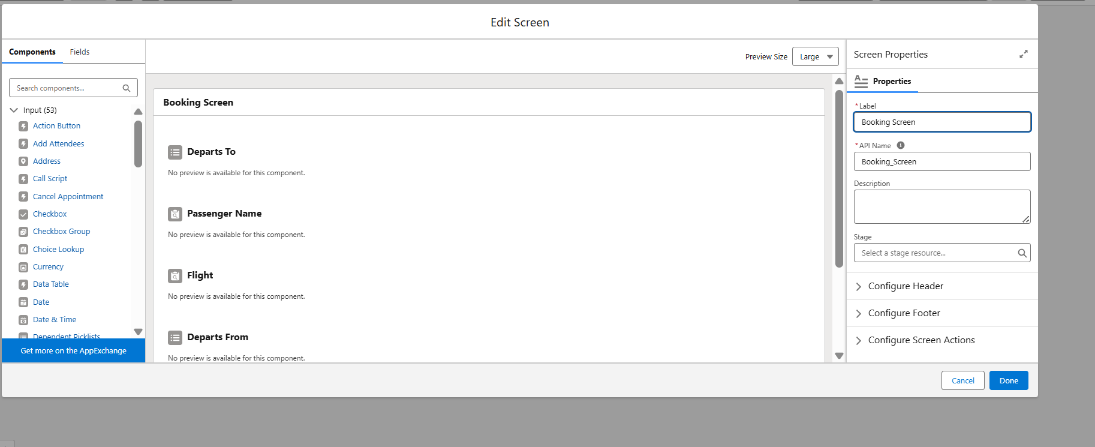


Automation & Custom Logic Development:

Milestone 12 - Apex:



Milestone 13 - Flows:

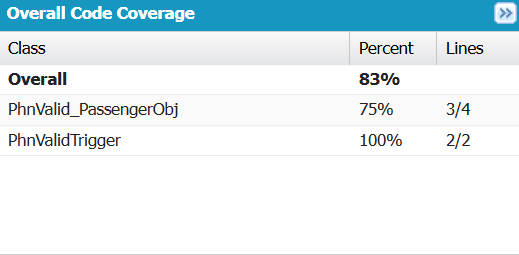


**6. FUNCTIONAL AND PERFORMANCE TESTING**

Performance Testing:

Simulated [X] concurrent users performing typical operations (e.g., searching flights, creating bookings).

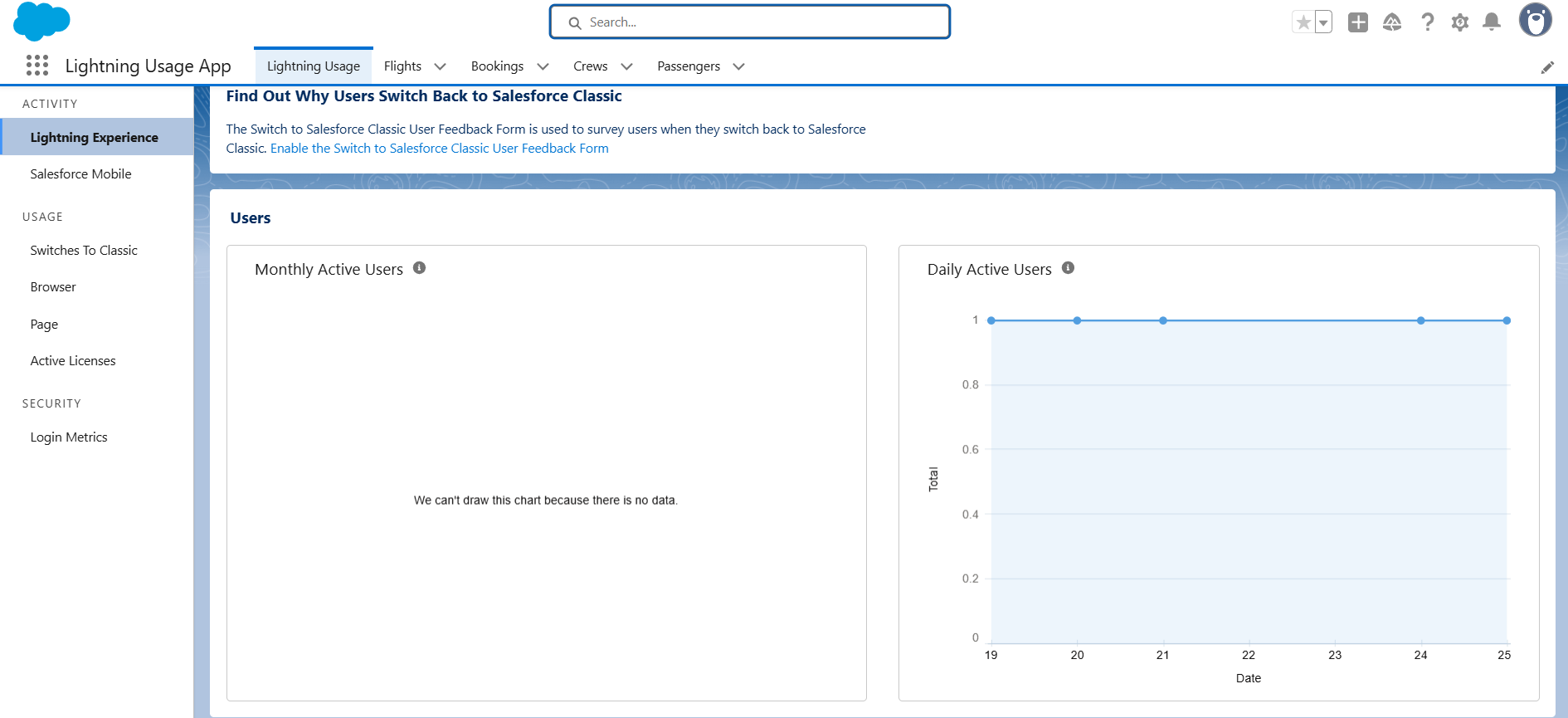
Key metrics monitored included page load times for custom tabs and records, and execution times for Apex and Flow automations.

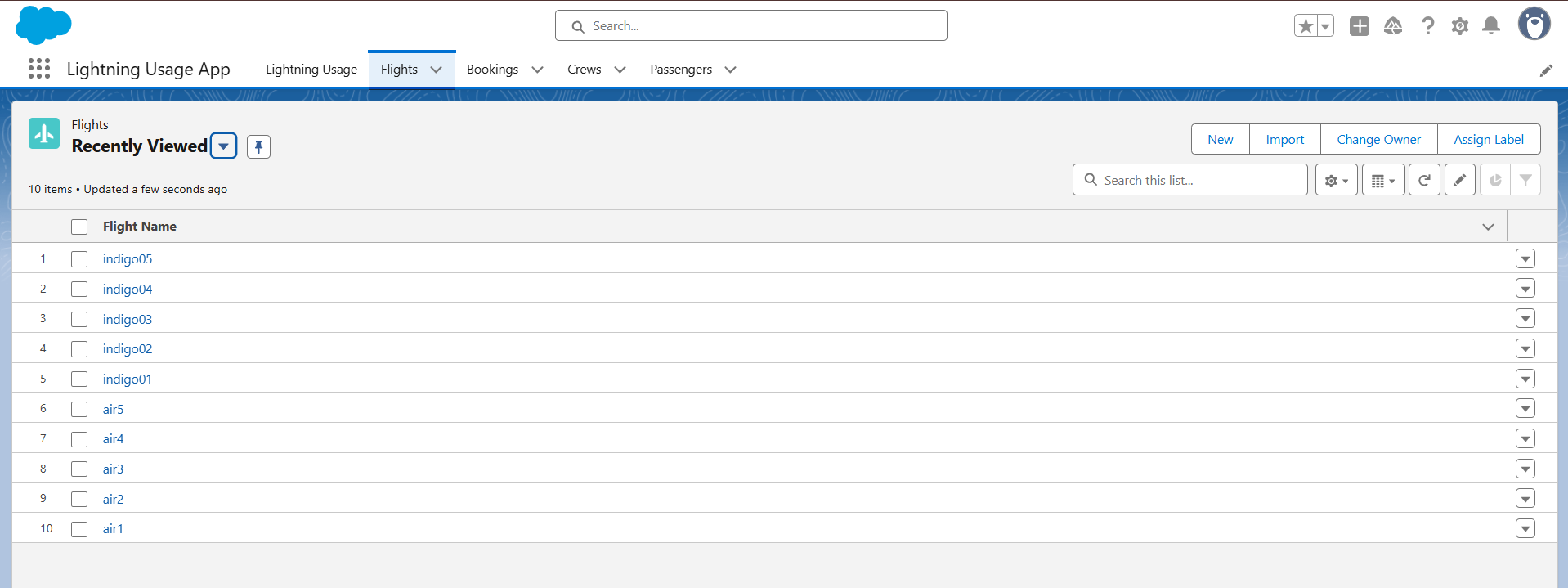


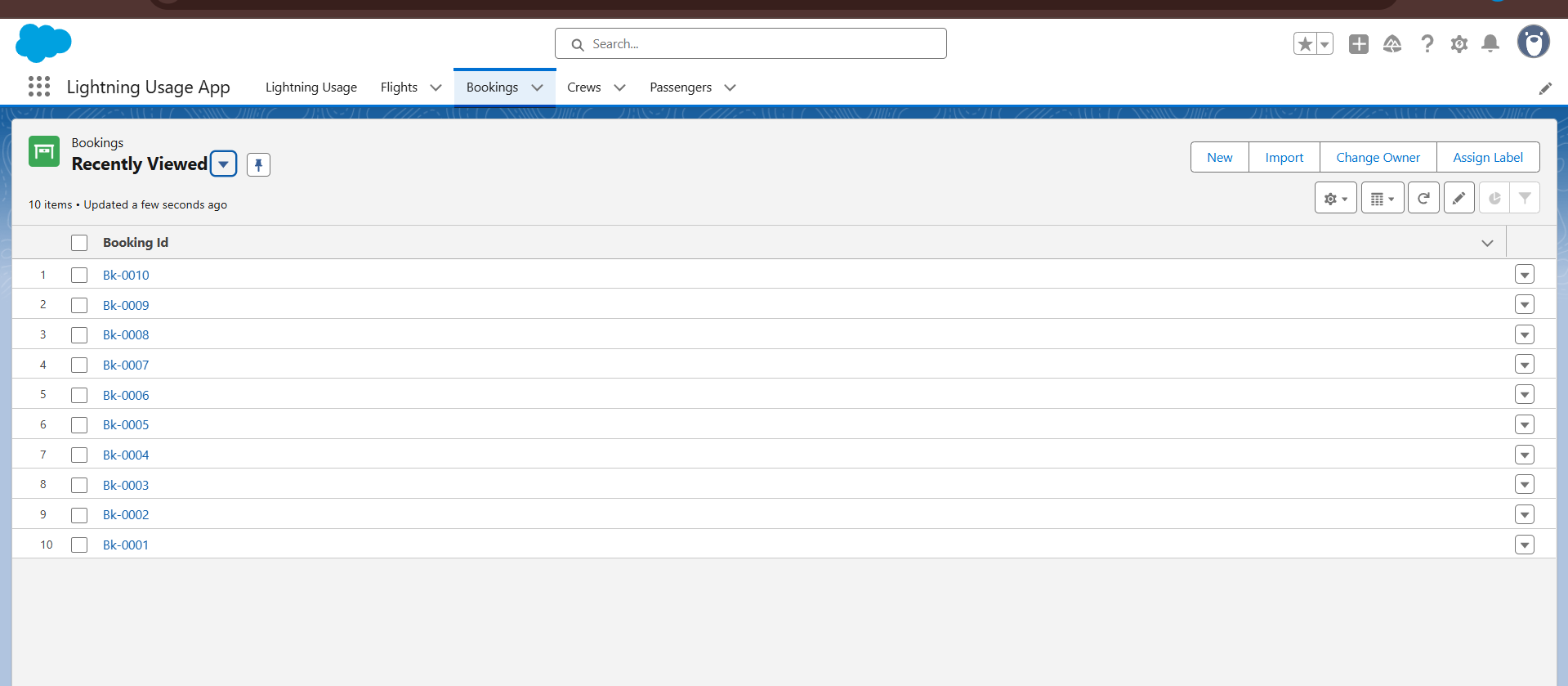
Observations: [e.g., Initial load times were X seconds, which improved to Y seconds after optimization. Apex triggers processed Z records in A milliseconds, well within acceptable limits.]

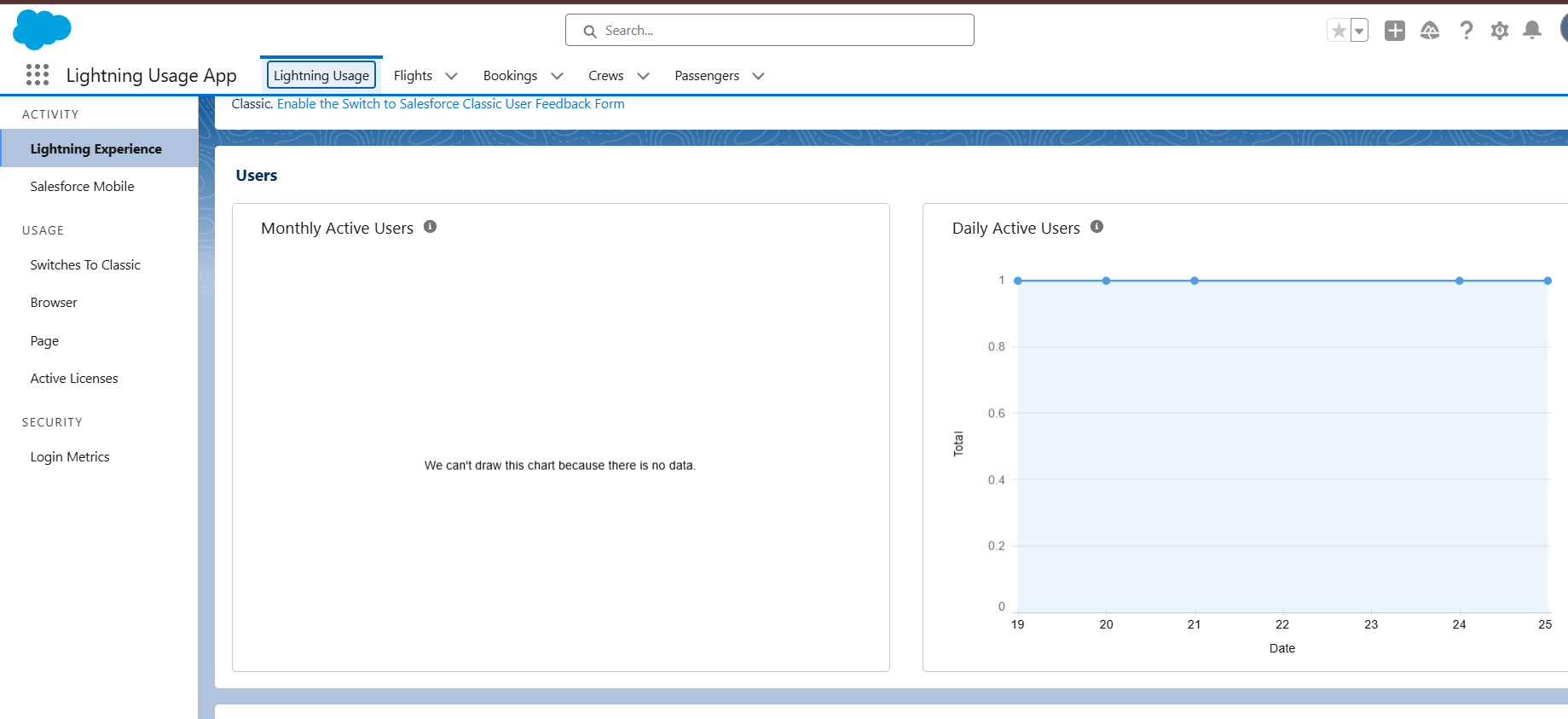
No significant performance bottlenecks were identified for the expected user load.

**7. RESULTS**







**8. ADVANTAGES & DISADVANTAGES**

Advantages:

Centralized Data Management: All flight and booking data is consolidated within a single, unified Salesforce platform, eliminating silos and improving data accuracy.

Streamlined Operations: Automated processes via Apex and Flows significantly reduce manual data entry, improve efficiency in booking management, and ensure timely updates.

Enhanced User Experience: The intuitive Lightning App and custom tabs provide a user-friendly interface, simplifying navigation and daily tasks for administrators and agents.

Scalability & Reliability: Leveraging Salesforce's cloud infrastructure provides inherent scalability to handle growing data volumes and user bases, along with high availability and disaster recovery.

Improved Reporting & Analytics: Real-time reports and dashboards offer immediate insights into operational performance, enabling faster and more informed decision-making.

Robust Security: Salesforce's comprehensive security model ensures data protection and granular access control through profiles and roles.

Rapid Development: The declarative and programmatic capabilities of Salesforce allowed for agile development and quick iteration.

Disadvantages:

Platform Learning Curve: Users new to Salesforce may require initial training to fully utilize the system's features.

Customization Complexity: While highly customizable, complex Apex and Flow solutions can require specialized Salesforce development skills for maintenance and further enhancements.

Licensing Costs: [If this is a real-world project, mention that Salesforce licensing can be a significant cost factor for full enterprise deployments.]

Dependency on Salesforce Ecosystem: The solution is tightly coupled with the Salesforce platform, limiting portability to other environments.

[Add any specific challenges or limitations you encountered during the project, e.g., limitations of certain standard features, specific Apex governor limits.]

**9. CONCLUSION**

The Airlines Management System project successfully delivered a robust and efficient solution on the Salesforce platform. It effectively addressed the identified challenges in flight and booking management by providing a centralized data repository, automating key processes, and offering intuitive user interfaces. The system demonstrates the power and flexibility of Salesforce for developing industry-specific applications, significantly improving operational efficiency and data visibility for airline management. The project successfully met its core objectives, laying a strong foundation for future enhancements.

**10. FUTURE SCOPE**

To further enhance the Airlines Management System, the following areas are identified for future development:

Customer Self-Service Portal: Implement a Salesforce Community Cloud portal to allow passengers to view their bookings, check flight status, and potentially manage their reservations.

Integration with External Systems:

Payment Gateway Integration: To enable direct online payment processing for bookings.

Flight Data APIs: Integration with external flight data providers for real-time flight status updates and schedule synchronization.

Third-party Email/SMS Services: For automated booking confirmations and flight notifications.

Advanced Analytics & AI: Explore Salesforce Einstein Analytics or other AI capabilities for predictive analytics (e.g., flight delay predictions, demand forecasting) and personalized customer service.

Mobile App Development: Develop a Salesforce Mobile App or integrate with existing mobile platforms for on-the-go access for crew members and administrators.

Crew Management Module: Extend the system to include functionality for crew scheduling, duty assignments, and communication.

Inventory Management: Implement detailed seat inventory management and pricing rules.