

# Airline Management System

<b>Date</b>	<b>25-06-2025</b>
<b>Team ID</b>	<b>LTVIP2025TMID31533</b>
<b>Project Name</b>	<b>Air Line management System</b>
<b>College Name</b>	<b>Ideal Institute Of Technology</b>

## TEAM MEMBERS

- **Team Leader:** Vishal Manibabu Kasukurthi  
**Email:** [vishalkasukurthi38@gmail.com](mailto:vishalkasukurthi38@gmail.com)
- **Team Member:** Sankar Danduprolu  
**Email:** [sankardanduprolu2004@gmail.com](mailto:sankardanduprolu2004@gmail.com)
- **Team Member:** Prem Sagar  
**Email:** [premsagar6754@gmail.com](mailto:premsagar6754@gmail.com)
- **Team Member:** Sathi Grishmanjana  
**Email:** [anjanaram219@gmail.com](mailto:anjanaram219@gmail.com)

# 1. INTRODUCTION

This Salesforce-based Airline Management System (AMS) was developed as a practical project during the SmartInternz Virtual Internship Program, aimed at giving students hands-on experience with cloud-based CRM platforms. The project involves building a centralized solution using Salesforce to streamline airline operations, improve customer service, and enhance data-driven decision-making.

## 1.1 Project Overview

Designed to streamline and automate key airline operations such as flight scheduling, ticket booking, passenger management, and customer service.

The project utilizes Salesforce tools like custom objects, flows, reports, and dashboards to build a user-friendly system. It also includes modules for complaint handling, notifications, and analytics, helping airlines improve efficiency and customer experience.

This project gives interns practical experience in CRM-based application development, teaching how real-world airline systems can be managed using cloud technology.

## 1.2 Project Purpose

The main purpose of the Airline Management Project is to develop a centralized, automated

system on the Salesforce platform to improve the efficiency, accuracy, and customer experience in airline operations.

---

# 2. IDEATION PHASE

## 2.1 Problem Statement

- Manual Processes – Airlines often rely on outdated or manual systems for booking, scheduling, and customer service, leading to inefficiencies.
- Poor Customer Experience – Delayed responses, lack of timely notifications, and service issues affect passenger satisfaction.
- Limited Data Insights – Without proper reporting tools, airlines struggle to analyze performance and make informed decisions.
- Inefficient Complaint Handling – Tracking and resolving passenger issues is time-consuming due to lack of automation and real-time status updates.

## 2.2 Proposed Solution

- Centralized System – Develop a unified platform on Salesforce to manage all airline operations like bookings, schedules, and customer data.
- Process Automation – Use Salesforce automation tools (Flows, Process Builder) to handle ticketing, cancellations, notifications, and issue tracking.
- Real-Time Data Access – Enable instant access to flight details, passenger information, and service updates for both staff and customers.

## 2.3 Objectives of the Project

- Automate Airline Operations – Streamline processes like flight scheduling, bookings, and cancellations using Salesforce tools.
- Improve Customer Experience – Provide timely updates, easy access to services, and efficient complaint resolution.
- Enhance Learning and Skill Development – Provide interns hands-on experience in CRM, cloud computing, and enterprise application development.

## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey Map

1. Open App
2. Search Flights
3. Select Flight
4. Enter Passenger and Booking Details
5. Confirm and Book Ticket
6. Receive Confirmation via Email/SMS

This journey outlines the end-to-end process a passenger follows within the AMS to complete a flight booking, highlighting critical user touchpoints.

### 3.2 Solution Requirement

#### Functional Requirements

1. Flight Management – Create and manage flight schedules, destinations, and available seats.
2. Passenger Management – Capture and validate passenger information, including passport details and contact information.
3. Booking System – Enable passengers to book flights, select seats, and receive confirmations.

4. Crew Management – Assign crew to flights and prevent duplicate assignments through validation.
5. Dashboard and Reporting – Real-time analytics and reports for bookings, revenue, and flight occupancy.
6. Lightning App Home Page – Centralized, user-friendly home page for navigation and quick access to modules.

### **Non-Functional Requirements**

- Data Accuracy – Enforced using Apex triggers and validation rules to ensure reliable and complete records.
- Ease of Use – Intuitive Lightning UI with simplified navigation and minimal training needs.
- Scalability – Designed to accommodate additional features like payment integration and mobile access.
- Security – Role-based access control (RBAC) ensures only authorized users can view/edit sensitive data.

## **3.3 Data Flow Diagram (DFD)**

User Input → Booking System → Salesforce Database → Confirmation Output (Email/SMS)

This high-level DFD shows how a passenger's action flows through the system, leading to a successful booking and notification.

## **4. Project Design**

### **4.1 Problem Solution Fit**

Airlines often struggle with fragmented data, manual booking processes, and operational inefficiencies. This project solves those problems by providing a cloud-based, automated solution that consolidates all core airline operations.

### **4.2 Proposed Solution**

The solution is a robust Airline Management System built on the Salesforce platform.

It enables end-to-end management of flights, passengers, bookings, and crew.

Features include real-time validations, automated reminders, centralized dashboards, and a clean Lightning Experience interface.

## 4.3 Solution Architecture

The AMS architecture utilizes Salesforce native tools and custom development:

- Custom Objects: Flight, Passenger, Booking, Crew
- Apex Triggers: Used for business rule enforcement (e.g., passport expiry, phone validation)
- Lightning App Builder: For building a responsive and role-based home page
- Dashboards & Reports: Provide visual insights on booking trends, flight loads, and revenue
- Process Automation: Workflow rules and scheduled Apex jobs handle routine alerts and notifications

Together, these components form a scalable, centralized, and secure platform for efficient airline operations.

## 5. PROJECT PLANNING & SCHEDULING

### 5.1 PROJECT PLANINNG

#### → Week 1: Project Initiation & Requirement Analysis

- Understand project scope and objectives
- Identify user roles (Admin, Passenger, Support Staff)
- Gather and document functional and non-functional requirements
- Research Salesforce tools and features relevant to the project
- Prepare initial project documentation

#### → Week 2: System Design

- Define custom objects (Flight, Passenger, Booking, Complaint)
- Design object relationships (lookup, master-detail)
- Plan field-level structure and data model
- Design user roles and permissions
- Draft entity-relationship (ER) diagrams

#### → Week 3: Salesforce Setup & Object Creation

- Set up Salesforce developer org
- Create custom objects, fields, and relationships
- Configure record types, page layouts, and validation rules
- Define profiles and permission sets for different user roles

#### → **Week 4: Automation & Workflow Setup**

- Use Flows and Process Builder to automate:
  - Booking confirmations
  - Flight updates
  - Complaint status notifications
- Set up email templates and alert systems
- Implement basic triggers (if needed)

#### → **Week 5: UI Design with Lightning App Builder**

- Design user interfaces using Lightning pages
- Create tabbed views for each object (Flights, Bookings, etc.)
- Add quick actions, list views, and related record components
- Test navigation and user experience

#### → **Week 6: Reporting & Dashboards**

- Create reports for:
  - Total bookings
  - Complaint resolution rate
  - Flight occupancy trends
- Build dashboards for Admin to view system metrics

#### → **Week 7: Testing & Feedback**

- Conduct system testing with sample data
- Perform unit and integration testing for objects and automation
- Collect feedback from test users
- Debug and fix issues

#### → **Week 8: Final Review & Documentation**

- Finalize the system and test for deployment readiness
- Prepare complete project documentation including:
  - User manual
  - Technical report
  - ER diagrams and screen designs
- Submit project for review and presentation

## 6. DEVELOPMENT PHASE

The development phase is completed when the airline management system is functionally complete, user-ready, and meets all defined requirements, ensuring it can now move to the testing, review, or submission stage.

It begins with setting up the Salesforce Developer Org and enabling the Lightning Experience.

Custom objects such as Flight, Passenger, Booking, and Complaint are created with appropriate relationships to manage airline operations. The user interface is designed using Lightning App

Builder, with customized page layouts for different user roles. Automation is implemented through Flows and Process Builder to handle tasks like booking confirmations and complaint status

updates. Role-based access is configured using profiles and permission sets to ensure secure data handling. Reports and dashboards are created to monitor key metrics such as booking trends


and complaint resolution. Finally, thorough testing is conducted using sample data to validate the system's functionality and fix any issues.

This phase ensures the transformation of design into a practical, user-ready airline management system.

# PROJECT DEVELOPMENT PHASE

Date	2-06-2025
Team Id	LTVIP2025TMID31533
Project Name	AirLine Management System
College Name	Ideal Institute Of Technology


## 1.Created developer org and explored platform features



### Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs



#### Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name

Vishal Manibabu Kasu

Last name

Team

Job title

student

Work email

anjanaram219@gmail

Company

Ideal Institute of Tech

Country/Region

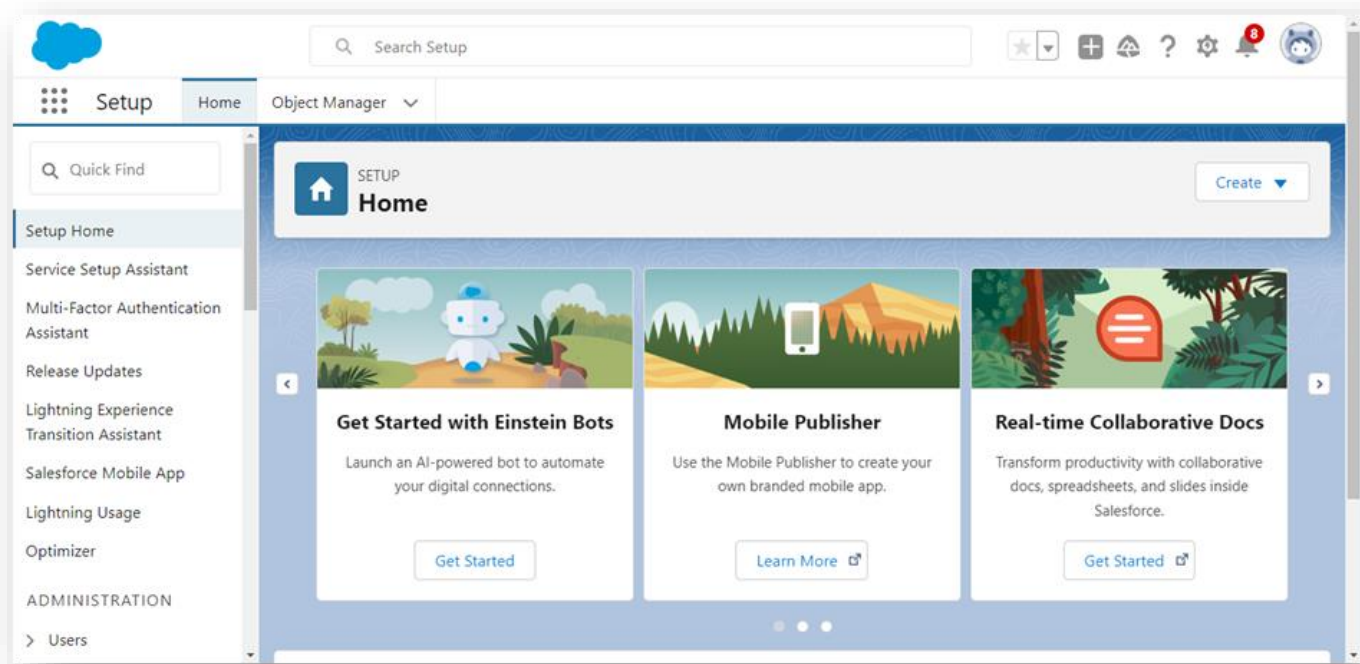
India

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud Infrastructure.

☒ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

We value your privacy. To learn more, visit our Privacy Statement.





## 2.Created flight object

Edit Custom Object  
**Flight**

Custom Object Definition Edit Save Save & New Cancel

Custom Object Information Required Information

The singular and plural labels are used in tabs, page layouts, and reports.  
Be careful when changing the name or label as it may affect existing integrations and merge templates.

Label  Example: Account

Plural Label  Example: Accounts

Starts with vowel sound ☐

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

Object Name  Example: Account

Description

Context-Sensitive Help Setting

☒ Open the standard Salesforce.com Help & Training window

☐ Open a window using a Visualforce page

Contact Name

Enter Record Name Label and Format

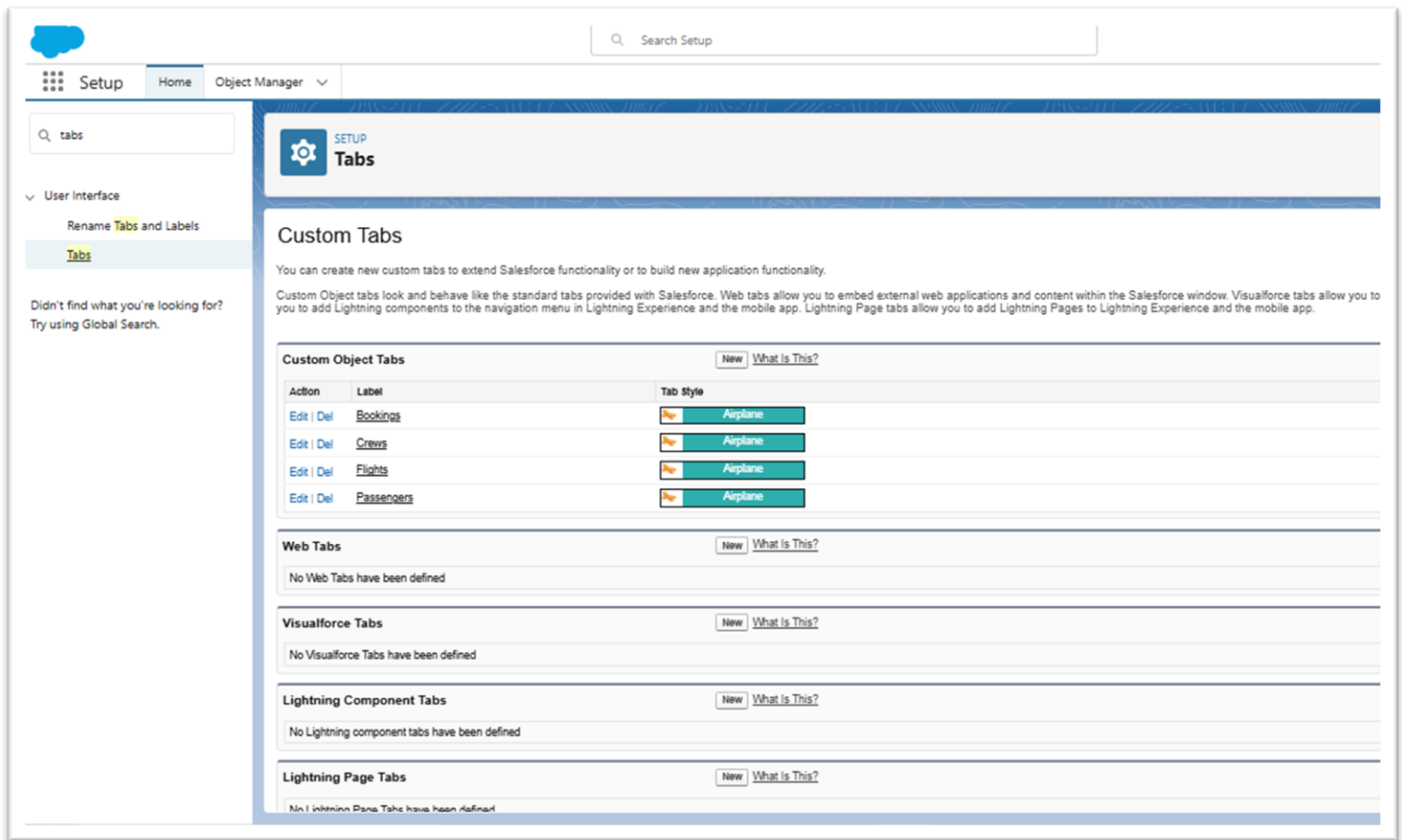
The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name  Example: Account Name

Data Type  Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Optional Features

### 3.Created core objects and tabs for Passenger, Booking, Flight, and Crew.



The screenshot shows the Salesforce Setup interface for Custom Tabs. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area is titled "Custom Tabs" and includes a search bar and a "Search Setup" button. Below the title, there is a section for "Custom Object Tabs" with a table listing tabs for Bookings, Crews, Flights, and Passengers, all using the "Airplane" tab style. Below this, there are sections for "Web Tabs", "Visualforce Tabs", "Lightning Component Tabs", and "Lightning Page Tabs", each with a "New" button and a "What Is This?" link.

**Custom Object Tabs**

Action	Label	Tab Style
Edit   Del	Bookings	Airplane
Edit   Del	Crews	Airplane
Edit   Del	Flights	Airplane
Edit   Del	Passengers	Airplane

**Web Tabs**

No Web Tabs have been defined

**Visualforce Tabs**

No Visualforce Tabs have been defined

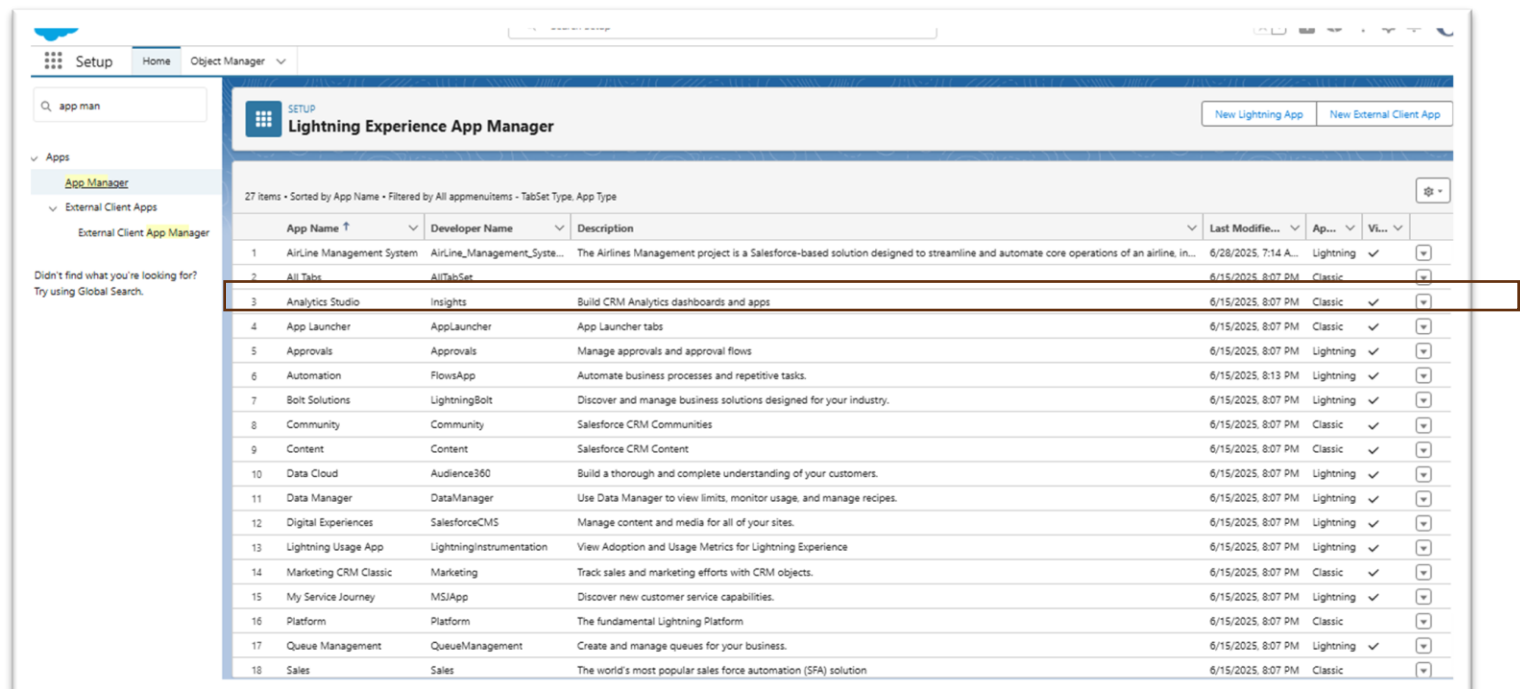
**Lightning Component Tabs**

No Lightning component tabs have been defined

**Lightning Page Tabs**

No Lightning Page Tabs have been defined

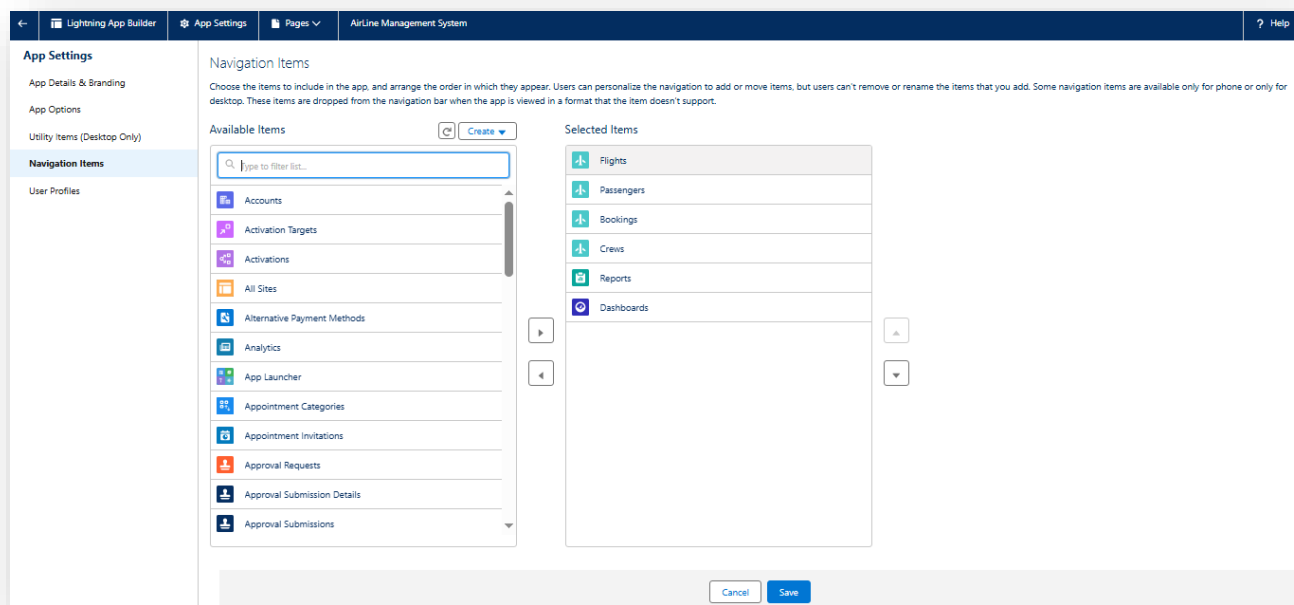
### 4.Developed a Lightning App to manage Passenger, Booking, Flight, and Crew modules from a single interface



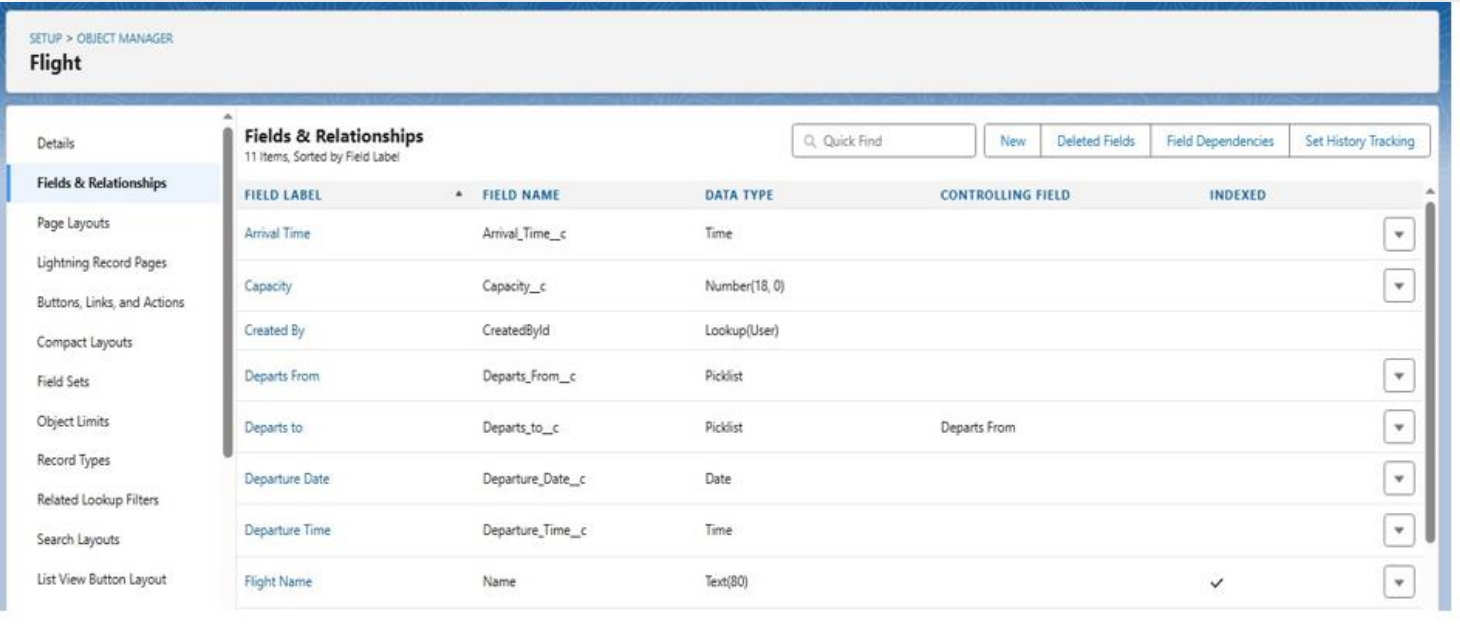
The screenshot shows the Salesforce Lightning Experience App Manager interface. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area is titled "Lightning Experience App Manager" and includes a search bar and a "Search Setup" button. Below the title, there is a table listing 27 items, sorted by App Name. The table has columns for App Name, Developer Name, Description, Last Modified, App Type, and Visibility. The first item is "AirLine Management System" with a description: "The Airlines Management project is a Salesforce-based solution designed to streamline and automate core operations of an airline. in...". The second item is "All Tabs" with a description: "Build CRM Analytics dashboards and apps". The third item is "Analytics Studio" with a description: "Build CRM Analytics dashboards and apps". The fourth item is "App Launcher" with a description: "App Launcher tabs". The fifth item is "Approvals" with a description: "Manage approvals and approval flows". The sixth item is "Automation" with a description: "Automate business processes and repetitive tasks". The seventh item is "Bolt Solutions" with a description: "Discover and manage business solutions designed for your industry". The eighth item is "Community" with a description: "Salesforce CRM Communities". The ninth item is "Content" with a description: "Salesforce CRM Content". The tenth item is "Data Cloud" with a description: "Build a thorough and complete understanding of your customers". The eleventh item is "Data Manager" with a description: "Use Data Manager to view limits, monitor usage, and manage recipes". The twelfth item is "Digital Experiences" with a description: "Manage content and media for all of your sites". The thirteenth item is "Lightning Usage App" with a description: "View Adoption and Usage Metrics for Lightning Experience". The fourteenth item is "Marketing CRM Classic" with a description: "Track sales and marketing efforts with CRM objects". The fifteenth item is "My Service Journey" with a description: "Discover new customer service capabilities". The sixteenth item is "Platform" with a description: "The fundamental Lightning Platform". The seventeenth item is "Queue Management" with a description: "Create and manage queues for your business". The eighteenth item is "Sales" with a description: "The world's most popular sales force automation (SFA) solution".

App Name	Developer Name	Description	Last Modified	App Type	Visibility
1 AirLine Management System	AirLine_Management_Syste...	The Airlines Management project is a Salesforce-based solution designed to streamline and automate core operations of an airline. in...	6/28/2025, 7:14 A...	Lightning	✓
2 All Tabs	AllTabs	Build CRM Analytics dashboards and apps	6/15/2025, 8:07 PM	Classic	✓
3 Analytics Studio	Insights	Build CRM Analytics dashboards and apps	6/15/2025, 8:07 PM	Classic	✓
4 App Launcher	AppLauncher	App Launcher tabs	6/15/2025, 8:07 PM	Classic	✓
5 Approvals	Approvals	Manage approvals and approval flows	6/15/2025, 8:07 PM	Lightning	✓
6 Automation	FlowsApp	Automate business processes and repetitive tasks.	6/15/2025, 8:13 PM	Lightning	✓
7 Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	6/15/2025, 8:07 PM	Lightning	✓
8 Community	Community	Salesforce CRM Communities	6/15/2025, 8:07 PM	Classic	✓
9 Content	Content	Salesforce CRM Content	6/15/2025, 8:07 PM	Classic	✓
10 Data Cloud	Audience360	Build a thorough and complete understanding of your customers.	6/15/2025, 8:07 PM	Lightning	✓
11 Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage recipes.	6/15/2025, 8:07 PM	Lightning	✓
12 Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	6/15/2025, 8:07 PM	Lightning	✓
13 Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	6/15/2025, 8:07 PM	Lightning	✓
14 Marketing CRM Classic	Marketing	Track sales and marketing efforts with CRM objects.	6/15/2025, 8:07 PM	Classic	✓
15 My Service Journey	MSJApp	Discover new customer service capabilities.	6/15/2025, 8:07 PM	Lightning	✓
16 Platform	Platform	The fundamental Lightning Platform	6/15/2025, 8:07 PM	Classic	✓
17 Queue Management	QueueManagement	Create and manage queues for your business.	6/15/2025, 8:07 PM	Lightning	✓
18 Sales	Sales	The world's most popular sales force automation (SFA) solution	6/15/2025, 8:07 PM	Classic	✓

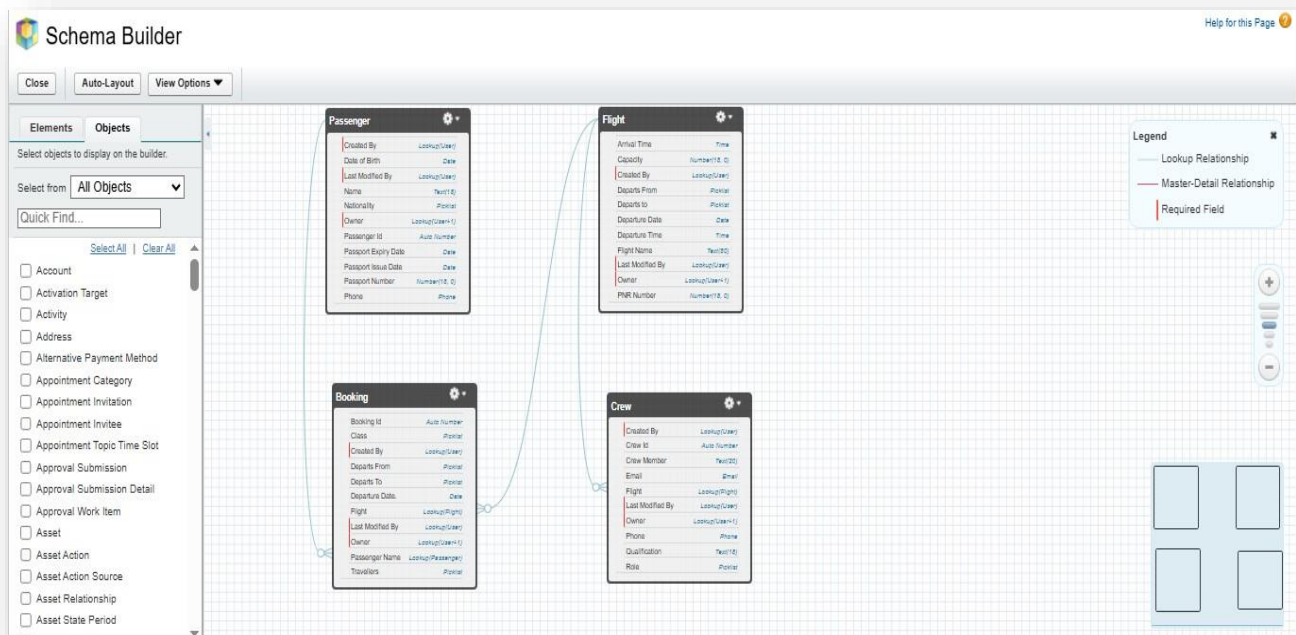
5.Created the 'Airline Management System' Lightning App with navigation tabs for Passenger, Booking, Flight, and Crew



6.Added custom fields, picklists, lookup relationships, and field dependencies to support data integrity and user input control



## 7.Used Salesforce Schema Builder to visualize and verify object relationships



## 8.Created 10 sample records for each object using the Salesforce Lightning interface flights, passengers,bookings,crews

The screenshot shows the Salesforce Lightning interface for the "AirLine Manage..." app. The "Flights" tab is selected, and the "Recently Viewed" list is displayed. The list contains 10 items, each with a checkbox and a flight name.

	Flight Name
1	<input type="checkbox"/> Delta Air Lines
2	<input type="checkbox"/> United Airlines
3	<input type="checkbox"/> SpiceJet
4	<input type="checkbox"/> IndiGo
5	<input type="checkbox"/> Qatar Airways
6	<input type="checkbox"/> Air France
7	<input type="checkbox"/> Lufthansa
8	<input type="checkbox"/> Emirates
9	<input type="checkbox"/> Singapore Airlines
10	<input type="checkbox"/> Air India



	Crew Id	
<input checked="" type="checkbox"/>	Bk-0001	
<input type="checkbox"/>	Bk-0010	
<input type="checkbox"/>	Bk-0009	
<input type="checkbox"/>	Bk-0008	
<input type="checkbox"/>	Bk-0007	
<input type="checkbox"/>	Bk-0006	
<input type="checkbox"/>	Bk-0005	
<input type="checkbox"/>	Bk-0004	
<input type="checkbox"/>	Bk-0003	
<input type="checkbox"/>	Bk-0002	

9.Configured user profiles with appropriate object permissions and field-level security.

**Clone Profile**

Enter the name of the new profile.

You must select an existing profile to clone from.

Existing Profile	Standard Platform User
User License	Salesforce Platform
Profile Name	Management Admin

Save Cancel

**Clone Profile**

Enter the name of the new profile.

You must select an existing profile to clone from.

Existing Profile	System Administrator
User License	Salesforce
Profile Name	Senior Admin

Save Cancel

## 10. Set up role hierarchy to define data access levels based on user roles within the organization

## 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](#)

 Ideal Institute of Technology

Add Role

 **CEO** Edit | Del | Assign

[Add Role](#)

 **CFO** Edit | Del | Assign

... [Add Role](#)

 COO Edit | Del | Assign

... [Add Role](#)

 Senior Admi

[Add Role](#)

 **General Admin** [Edit](#) | [Del](#) | [Assign](#)

[Add Role](#)

 **Assistant Admin** [Edit](#) | [Del](#) | [Assign](#)

... [Add Role](#)


Management Admin	Edit	Del	Assign

Add Role

**Crew Member** [Edit](#) | [Del](#) | [Assign](#)

[Add Role](#)

[Ground Station](#)

 **Shift Supervisor** Edit | Del | Archive

Add Role

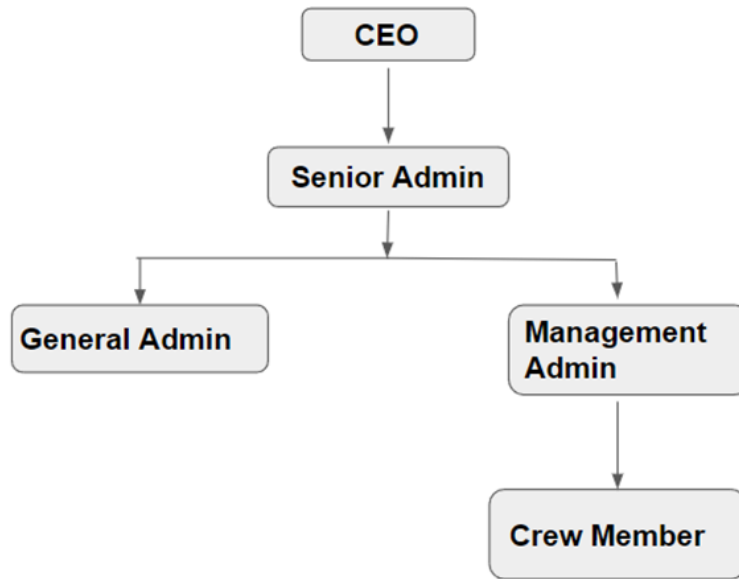
SVP Customer Service &amp; Support Edit | Del | Archive

Add Role

Customer Support International Edit Del Assign

Add Role

Customer Support: North America Edit | Del | Assign



**Role Hierarchy:** The above diagram represents which role reports to which one.

## 11.Created and managed users by assigning appropriate roles and profiles

User Edit  
Niklaus Mikaelson

Save Save & New Cancel

General Information

First Name	Niklaus	Role	Senior Admin
Last Name	Mikaelson	User License	Salesforce
Alias	nmika	Profile	Senior Admin
Email	anjanaram219@gmail.com	Active	<input checked="" type="checkbox"/>
Username	niki@mike.com	Marketing User	<input type="checkbox"/>
Nickname	niki	Offline User	<input type="checkbox"/>
Title		Knowledge User	<input type="checkbox"/>
Company		Flow User	<input type="checkbox"/>
Department		Service Cloud User	<input type="checkbox"/>
Division		Site.com Contributor User	<input type="checkbox"/>
		Site.com Publisher User	<input type="checkbox"/>
		WDC User	<input type="checkbox"/>
		Data.com User Type	None



SETUP

Users

User Edit

Kol Mikaelson

User Edit

Save Save & New Cancel

General Information

First Name

Kol

Last Name

Mikaelson

Alias

kmika

Email

anjanaram219@gmail.com

Username

kol@mika.com

Nickname

kolson

Title

Company

Department

Division

Role

General Admin

User License

Salesforce Platform

Profile

General Admin

Active

☐

Marketing User

☐

Offline User

☐

Knowledge User

☐

Flow User

☐

Service Cloud User

☐

Site.com Contributor User

☐

Site.com Publisher User

☐

WDC User

☐

Data.com User Type

--None--

Data.com Monthly Addition Limit

300

Accessibility Mode (Classic Only)

☐

High-Contrast Palette on Charts

☐

User Edit

Damon Salvatore

User Edit

Save Save & New Cancel

General Information

First Name

Damon

Last Name

Salvatore

Alias

dsalv

Email

anjanaram219@gmail.com

Username

damon@sal.com

Nickname

damon

Title

Company

Department

Division

Role

Management Admin

User License

Salesforce Platform

Profile

Management Admin

Active

☒

Marketing User

☐

Offline User

☐

Knowledge User

☐

Flow User

☐

Service Cloud User

☐

Site.com Contributor User

☐

Site.com Publisher User

☐

WDC User

☐

Data.com User Type

--None--

Data.com Monthly Addition Limit

300

SETUP

Users

User Edit

Alice martin

User Edit

Save Save & New Cancel

General Information

First Name

Alice

Last Name

martin

Alias

amart

Email

anjanaram219@gmail.com

Username

alice@martin1.com

Nickname

User175067051829988068

Title

Company

Department

Division

Role

Crew Member

User License

Salesforce Platform

Profile

Crew Members

Active

☒

Marketing User

☐

Offline User

☐

Knowledge User

☐

Flow User

☐

Service Cloud User

☐

Site.com Contributor User

☐

Site.com Publisher User

☐

WDC User

☐

Data.com User Type

--None--

Data.com Monthly Addition Limit

300

12.Created custom reports to track flight bookings, passenger details, and crew assignments.

AirLine Manage...

Flights

Passengers

Bookings

Crews

Reports

Dashboards

Search...

Reports

Recent

3 items

Search recent reports...

REPORTS	Report Name	Description	Folder	Created By	Created On
Recent	Bookings with Flight Details		Private Reports	Vishal Manibabu Kasukurthi Team	6/25/2025, 2:19 AM
Created by Me	Crew with Flight		Private Reports	Vishal Manibabu Kasukurthi Team	6/25/2025, 3:08 AM
Private Reports	Booking with Passenger Name		Private Reports	Vishal Manibabu Kasukurthi Team	6/25/2025, 3:04 AM
Public Reports					
All Reports					
FOLDERS					
All Folders					
Created by Me					
Shared with Me					
FAVORITES					
All Favorites					

13. Developed visual dashboards (e.g., class-wise bookings

AirLine Manage...

Flights

Passengers

Bookings

Crews

Reports

Dashboards

Search...

Star

Grid

Share

Help

Settings

Notifications

Profile

Dashboards

Recent

3 items

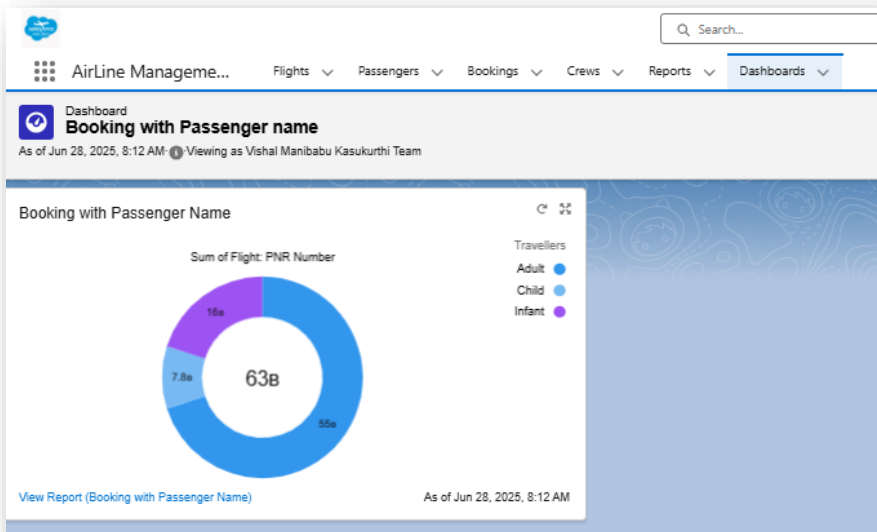
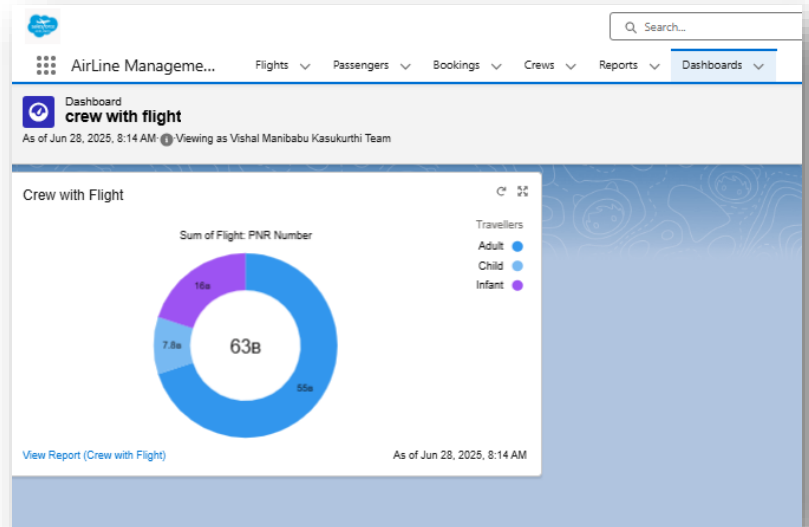
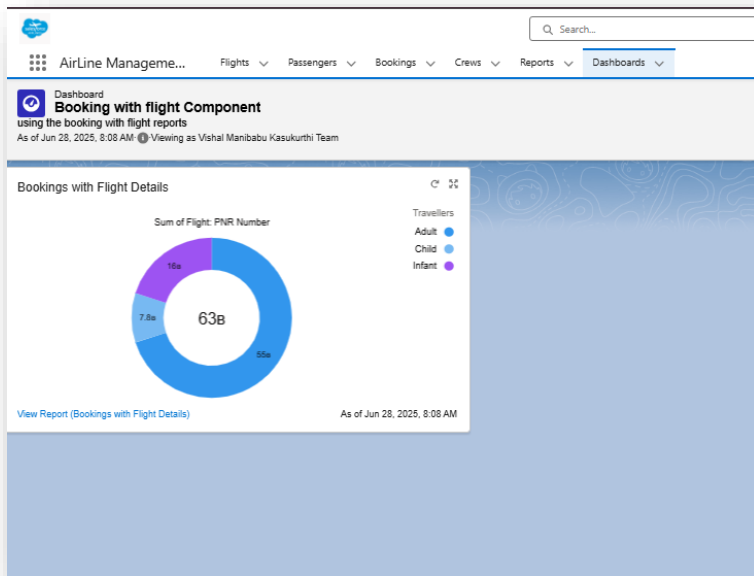
Search recent dashboards...

New Dashboard

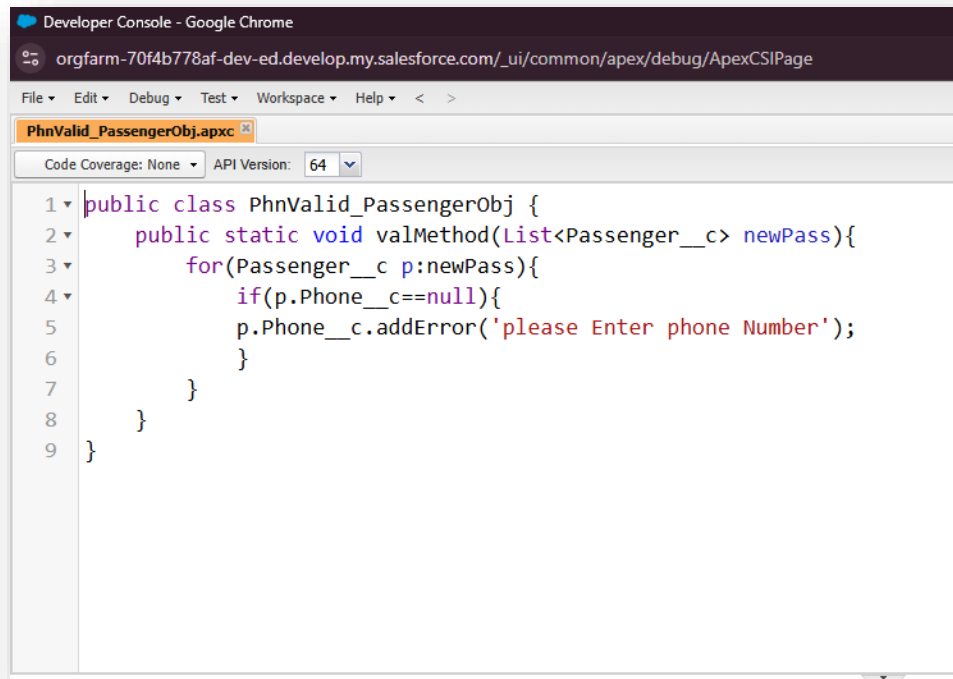
New Folder

Settings

DASHBOARDS	Dashboard Name	Description	Folder	Created By	Created On	Subscribed
Recent	Booking with flight Component	using the booking with flight reports	Private Dashboards	Vishal Manibabu Kasukurthi Team	6/25/2025, 2:57 AM	<input type="checkbox"/>
Created by Me	Booking with Passenger name		Private Dashboards	Vishal Manibabu Kasukurthi Team	6/28/2025, 8:08 AM	<input type="checkbox"/>
Private Dashboards	crew with flight		Private Dashboards	Vishal Manibabu Kasukurthi Team	6/28/2025, 8:10 AM	<input type="checkbox"/>
All Dashboards						
FOLDERS						
All Folders						
Created by Me						
Shared with Me						
FAVORITES						
All Favorites						

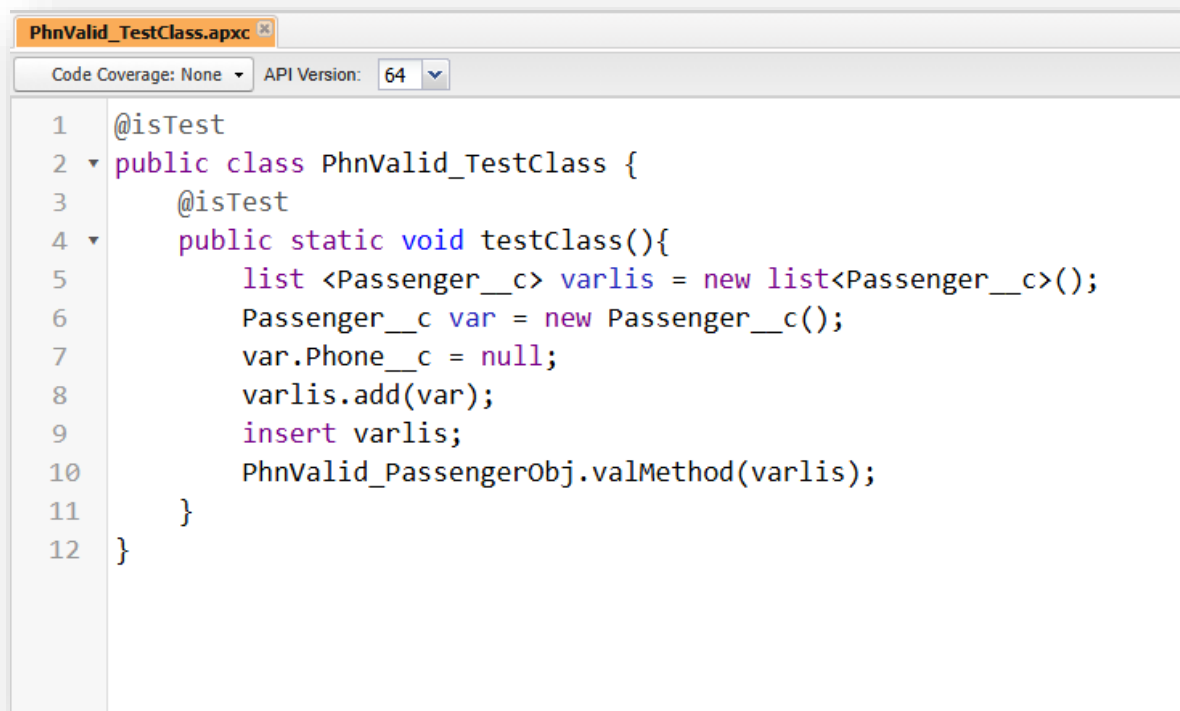


#### 14. Developed Apex class, trigger, and test class to validate Passenger phone input during record creation



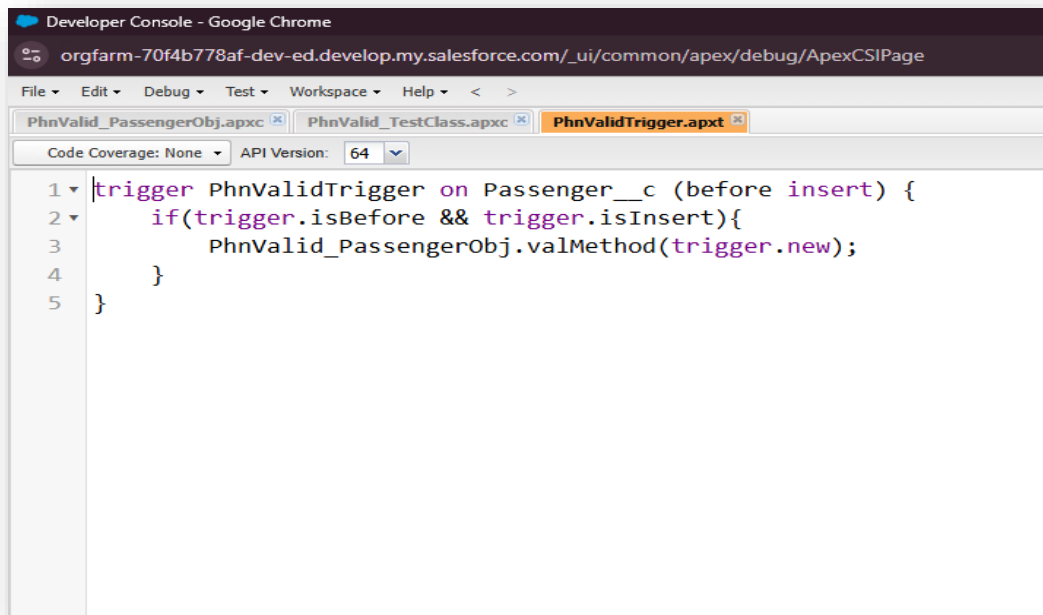
The screenshot shows the Salesforce Developer Console with the Apex class `PhnValid_PassengerObj.apxc` open. The class contains a static method `valMethod` that iterates through a list of `Passenger__c` objects and adds an error message if the phone number is null.

```
1 public class PhnValid_PassengerObj {
2     public static void valMethod(List<Passenger__c> newPass){
3         for(Passenger__c p:newPass){
4             if(p.Phone__c==null){
5                 p.Phone__c.addError('please Enter phone Number');
6             }
7         }
8     }
9 }
```



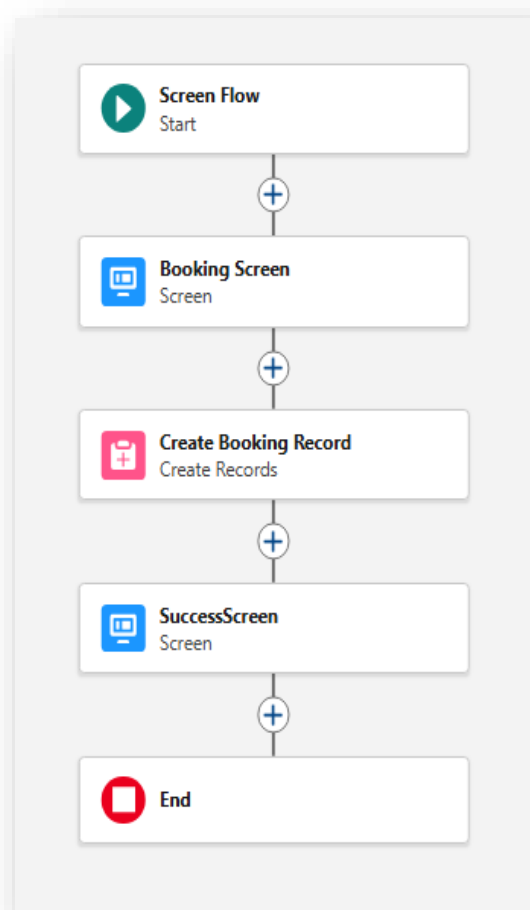
The screenshot shows the Salesforce Developer Console with the Apex test class `PhnValid_TestClass.apxc` open. The test class includes a static test method `testClass` that creates a `Passenger__c` object with a null phone number, adds it to a list, and calls the `valMethod` from the `PhnValid_PassengerObj` class to validate the input.

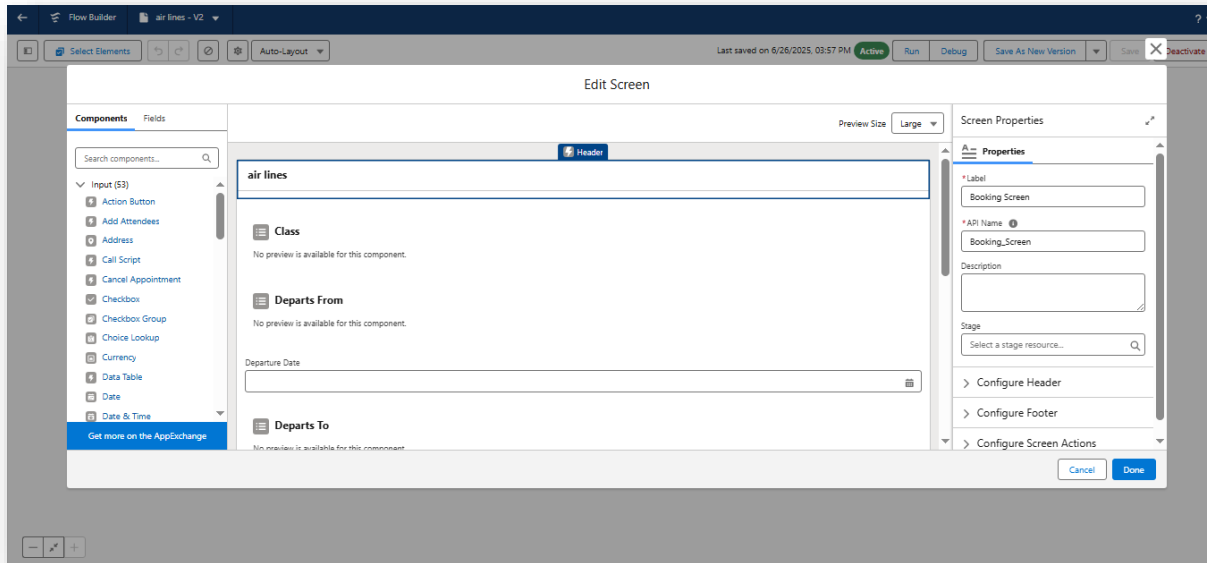
```
1 @isTest
2 public class PhnValid_TestClass {
3     @isTest
4     public static void testClass(){
5         list <Passenger__c> varlis = new list<Passenger__c>();
6         Passenger__c var = new Passenger__c();
7         var.Phone__c = null;
8         varlis.add(var);
9         insert varlis;
10        PhnValid_PassengerObj.valMethod(varlis);
11    }
12 }
```



```
1 trigger PhnValidTrigger on Passenger__c (before insert) {
2     if(trigger.isBefore && trigger.isInsert){
3         PhnValid_PassengerObj.valMethod(trigger.new);
4     }
5 }
```

15. Built a Booking Flow with a confirmation screen displaying a success message upon completion





Last saved on 6/26/2025, 03:57 PM

Active

Run

Debug

Save As New Version

Save

Deactivate

Create Records

\* Label

Create Booking Record

\* API Name

Create\_Booking\_Record

Description

\* How to set record field values

Manually

Create a Record of This Object

\* Object

Booking

Set Field Values for the Booking

Field	Value
Class	bookingObject > Class
Departs From	bookingObject > Departs From
Departs To	bookingObject > Departs To
Departure Date	bookingObject > Departure Date
Flight	bookingObject > Flight
Passenger Name	bookingObject > Passenger Name

# Edit Screen

Components

Fields

Preview Size

Large

Screen Properties

Search components...

Input (53)

Action Button

Add Attendees

Address

Call Script

Cancel Appointment

Get more on the AppExchange

1. We are happy to inform you that your booking for [tour name] is confirmed! Get ready to create some unforgettable memories. We've made things easy for you and included all of your booking details in this very email. All you need to do is show us this email on the day you arrive, and you'll be good to go!

Date: {!bookingObject.Departure\_Date\_\_c}  
Departs from: {!bookingObject.Departs\_From\_\_c}  
Departs to: {!bookingObject.Departs\_To\_\_c}  
Class: {!bookingObject.Class\_\_c}  
Travellers: {!bookingObject.Travellers\_\_c}

Properties

\* Label

SuccessScreen

\* API Name

SuccessScreen

Description

Cancel

Done

## PROJECT EXECUTABLE FILES

In the development of the Airlines Management System, Salesforce's low-code platform was enhanced using **Apex programming** to implement custom validations and business logic.

These executable files represent the backend logic that enforces data integrity and automates specific actions within the system

The following components were developed as part of the executable logic:

- **Apex Class:** Contains reusable methods that enforce custom validation rules on the **Passenger** object.
- **Apex Trigger:** Automatically calls the validation method before a Passenger record is inserted, ensuring mandatory fields are checked.
- **Apex Test Class:** Validates the logic through unit testing, ensuring that the trigger behaves correctly and achieves over 75% code coverage—a Salesforce deployment requirement

### 1.1 Apex Class: PhnValid\_PassengerObj.cls

```
public class PhnValid_PassengerObj {  
    public static void valMethod(List<Passenger__c> newPass) {  
        for (Passenger__c p : newPass) {  
            if (p.Phone__c == null) {  
                p.Phone__c.addError('Please enter phone number');  
            }  
        }  
    }  
}
```

### 1.2 Apex Trigger: PhnValidTrigger

```
trigger PhnValidTrigger on Passenger__c (before insert) {  
    if (Trigger.isBefore && Trigger.isInsert) {  
        PhnValid_PassengerObj.valMethod(Trigger.new);  
    }  
}
```



```
}  
}
```

### 1.3 Apex Test Class: PhnValid\_TestClass

@isTest

```
public class PhnValid_TestClass {  
    @isTest  
    public static void testClass() {  
        List<Passenger__c> varlis = new List<Passenger__c>();  
        Passenger__c var = new Passenger__c();  
        var.Phone__c = null;  
        varlis.add(var);  
        insert varlis;  
        PhnValid_PassengerObj.valMethod(varlis);  
    }  
}
```

## 2. DATASET

### 2.1 Dataset Creation

To validate and test the Airline Management System, **sample datasets** were manually created for each key custom object. A minimum of **10 records** were entered for the following Salesforce custom objects:

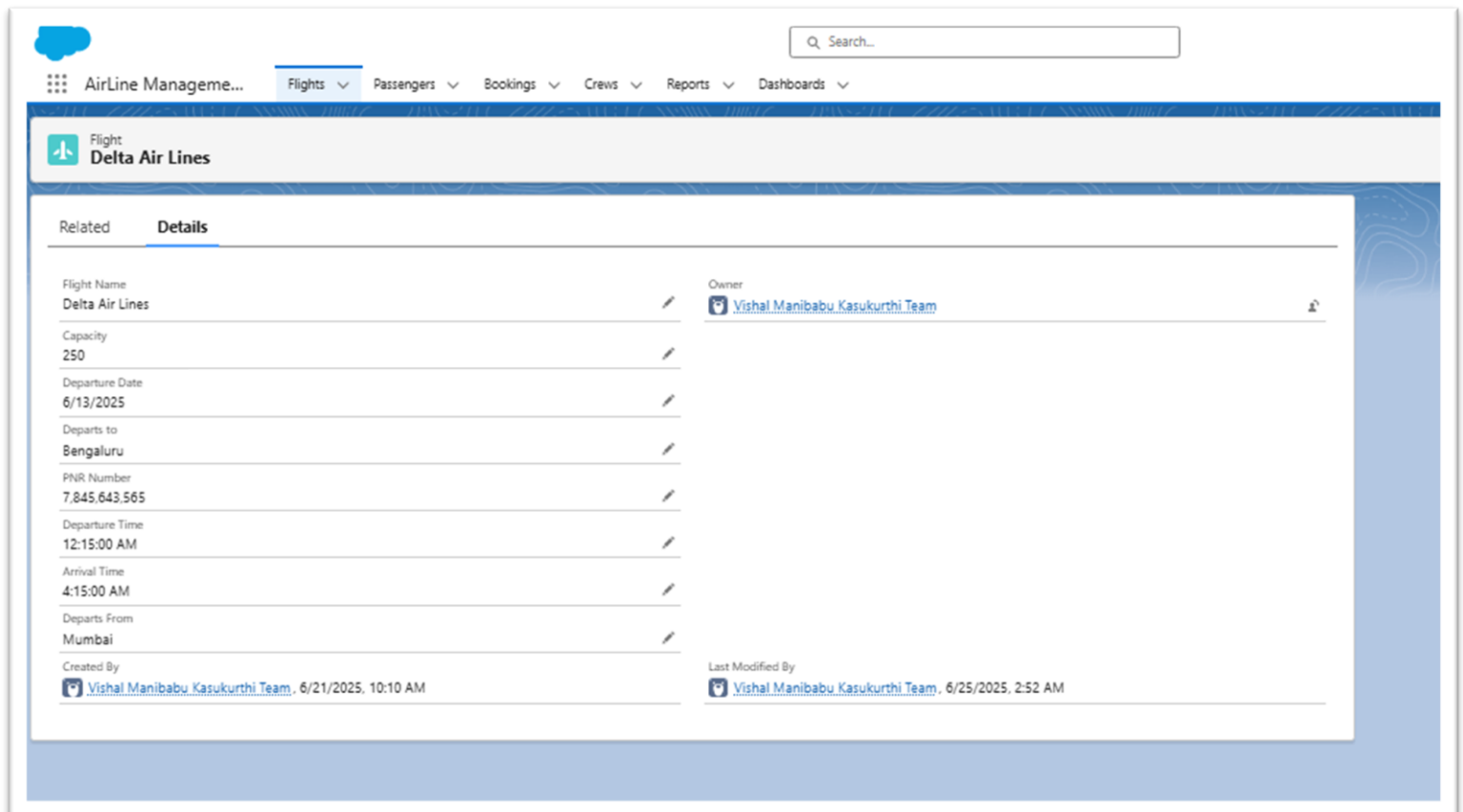
- ✈️ **Flight**
- 🧑 **Passenger**
- 📄 **Booking**
- 👮 **Crew**

### 2.1.1 ✈ Flight Object Dataset

To simulate real-world airline operations, 10 sample flight records were created in the **Salesforce Lightning App**. Each record includes critical information such as:

- **Flight Name**
- **Departs From (Origin)**
- **Departs To (Destination)**
- **Capacity**
- **PNR Number**
- **Departure Time**
- **Arrival Time**

These records help model realistic scheduling, route management, and passenger planning scenarios within the system.




## 2.1.2 Passenger

### Passenger Records Creation

A total of **10 Passenger records** were created within the **Airlines Management System** to represent various travelers booking flights through the platform. Each record includes essential personal and travel-related information such as:

- **Passenger Name**
- **Phone Number**
- **Passport Details**
  - Passport Number
  - Issue Date
  - Expiry Date
- **Date of Birth**
- **Nationality** (*selected from a predefined picklist*)

These records were created via the **Passenger** tab in the **Salesforce Lightning App**, allowing for realistic passenger data simulation for testing and validation purposes.

 Passenger  
Bk-0011

Related

Details

Passenger Id  
Bk-0011

Name  
bhagii

Phone  
(789) 145-2367


Passport Number  
4366,918


Passport Issue Date  
6/4/2025


Passport Expiry Date  
6/12/2036

Date of Birth  
6/2/1989

Nationality  
India

Created By  
 Vishal Manibabu Kasukurthi Team . 6/21/2025, 10:04 AM

Owner  
 Vishal Manibabu Kasukurthi Team

Last Modified By  
 Vishal Manibabu Kasukurthi Team . 6/21/2025, 10:04 AM


## 2.1.3 Booking

### Booking Records Creation

A total of 10 Booking records were created to simulate real-time flight reservation scenarios within the Airline Management System. Each booking represents a passenger reserving a seat on a scheduled flight.

Each Booking record includes the following key details:

- Booking ID
- Passenger Name
- Flight Name
- Booking Date
- Seat Number
- Booking Status
- Payment Status

 Booking  
**Bk-0007**

Related

Details

Booking Id  
Bk-0007

Passenger Name  
[Bk-0006](#)

Departs From  
Bengaluru

Departs To  
Delhi


Travellers  
Child


Class  
Economy


Flight  
[Air India](#)

PNR Number  
7,845,643,565

Departure Date

Owner  
 [Vishal Manibabu Kasukurthi Team](#)

Created By  
 [Vishal Manibabu Kasukurthi Team](#) · 6/21/2025, 10:07 AM

Last Modified By  
 [Vishal Manibabu Kasukurthi Team](#) · 6/25/2025, 3:01 AM

These entries were created using the Booking tab in the Salesforce Lightning App, enabling end-to-end testing of flight reservation workflows and data integrity across related objects.


## 2.1.4. Crew

### Crew Records Creation

A total of **10 Crew records** were created in the **Airline Management System** to represent the personnel responsible for flight operations. These records include pilots, co-pilots, and flight attendants assigned to various flights.

Each Crew record captures the following essential details:

- **Crew Member Name**
- **Role** (e.g., *Pilot, Co-Pilot, Flight Attendant* — from a picklist)
- **Employee ID**
- **Phone Number**
- **Email Address**
- **Assigned Flight** (linked from the *Flight object*)
- **License/Certification Number**
- **Availability Status** (e.g., *Available, On Duty, Off Duty* — from a picklist)

 Crew  
**Bk-0006**

Related

Details

Crew Id  
Bk-0006

Flight  
[Lufthansa](#)


Crew Member  
junnu


Phone  
(789) 145-2367


Email  
airline123@gmail.com

Qualification  
b tech

Role  
Air Stewardesses

Created By  
 Vishal Manibabu Kasukurthi Team · 6/21/2025, 10:18 AM




Owner  
 Vishal Manibabu Kasukurthi Team

Last Modified By  
 Vishal Manibabu Kasukurthi Team · 6/21/2025, 10:18 AM

# OVERVIEW

This Airline Management System (AMS) project is built on the **Salesforce platform** and integrates both backend logic and frontend visualization to deliver a complete airline operations solution.

It includes:

-  **Apex Classes, Triggers, and Test Classes** to implement core business logic, automate processes, and ensure robust data validation.
-  **Dashboard Reports and Analytics** to monitor operations in real-time with visual insights into bookings, flights, and crew allocations.
- ✈️ **Sample Datasets** for key custom objects — *Passenger*, *Flight*, *Booking*, and *Crew* — created to simulate real-world airline scenarios and support testing.
-  **Output Screenshots** from the Salesforce Org, showcasing:
  - Successful record creation and execution
  - Custom error handling via validations and triggers
  - The custom Lightning App interface
  - Report views and charts

This project demonstrates the seamless integration of **business logic**, **automation**, and **visual reporting** using Salesforce Lightning Experience, making it a scalable and user-friendly airline management solution.

## FUNCTIONAL & PERFORMANCE TESTING OUTPUTS

Date	26-06-2025
Team Id	LTVIP2025TMID31533
Project Name	Air Line Management System
College Name	Ideal Institute Of Technology

### 7.1 Output Screenshots

Trigger Error on Passport Expiry date

The screenshot shows a web form for passport application. The fields are: Name (Anjana), Phone (7891452367), Passport Number (4,366,918), Passport Issue Date (6/6/2025), Passport Expiry Date (7/1/2020), Date of Birth (6/23/2025), and Nationality (India). The Passport Expiry Date field is highlighted with a red border and a red error message: "Passport expiry date cannot be in the past." A red modal box with a close button (X) is displayed over the form, containing the text "We hit a snag." and "Review the following fields" with a bullet point: "Passport Expiry Date". At the bottom of the form, there are three buttons: "Cancel", "Save & New", and "Save".

**Trigger Validation Error When Passport Expiry Date Is in the Past**

To ensure data accuracy, I implemented a before insert and before update Apex trigger that prevents a **Passenger** record from being saved if the **Passport Expiry Date** is in the past. This validation was handled using a custom Apex class and tested with both valid and invalid expiry dates.

### **Class Name: PassportExpiryValidator.cls**

```
public class PassportExpiryValidator {  
    public static void validate(List<Passenger__c> passengers) {  
        for (Passenger__c p : passengers) {  
            if (p.Passport_Expiry_Date__c != null && p.Passport_Expiry_Date__c <  
Date.today()) {  
                p.Passport_Expiry_Date__c.addError('Passport expiry date cannot be in  
the past.');            }  
        }  
    }  
}
```

✓ This class checks each record's Passport Expiry Date and adds an error if the date is earlier than today.

### **Step 2: Created Apex Trigger**

```
trigger PassportExpiryTrigger on Passenger__c (before insert, before update) {  
    if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {  
        PassportExpiryValidator.validate(Trigger.new);  
    }  
}
```



## Trigger Validation Error When Mobile Number Is Missing

\* = Required Information

Information

Passenger Id

Name

bhagii

Phone

please Enter phone Number

Passport Number

4,366,918

Passport Issue Date

6/24/2025

Passport Expiry Date

6/12/2025

Date of Birth

6/18/2025

Nationality

India

Owner

Vishal Manibabu Kasukurthi Team

We hit a snag.

Review the following fields

- Phone

Cancel

Save & New

Save

To ensure data accuracy, I implemented a **before-insert Apex trigger** that prevents a **Passenger** record from being created without a mobile number. This validation was handled using a custom **Apex class** and tested with both valid and invalid inputs.

### Step 1: Created a apex class

**Class Name: PhnValid\_PassengerObj.cls**

```
public class PhnValid_PassengerObj {
```

```

public static void valMethod(List<Passenger__c> newPass){
    for(Passenger__c p : newPass){
        if(p.Phone__c == null){
            p.Phone__c.addError('Please enter phone number');
        }
    }
}

```

This class checks every passenger record and throws an error if the phone number is missing.

### Step 2: Created an Apex Trigger

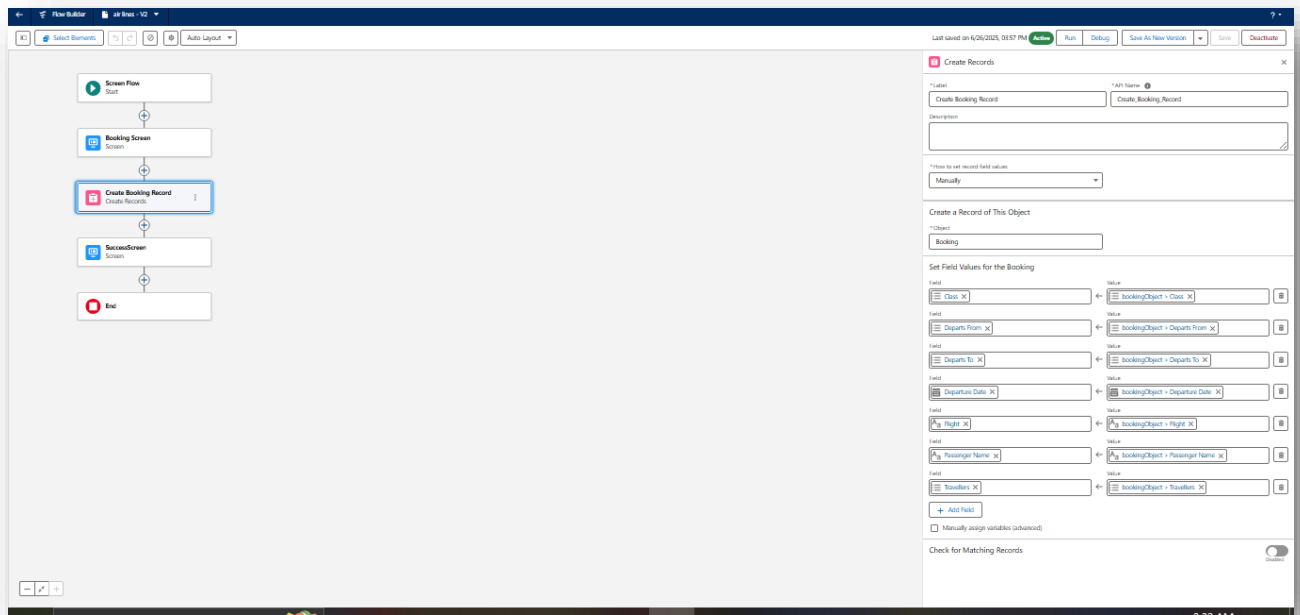
```

trigger PhnValidTrigger on Passenger__c (before insert) {
    if(Trigger.isBefore && Trigger.isInsert){
        PhnValid_PassengerObj.valMethod(Trigger.new);
    }
}


```

This trigger runs **before insert** and calls the validation method.

- **Designed a Booking Flow with input screens and a success message on completion.**




- Created sample Passenger and Booking records to demonstrate system functionality


 Search...

**AirLine Manage...** Flights Passengers **Bookings** Crews Reports Dashboards

**Booking**  
**Bk-0001**

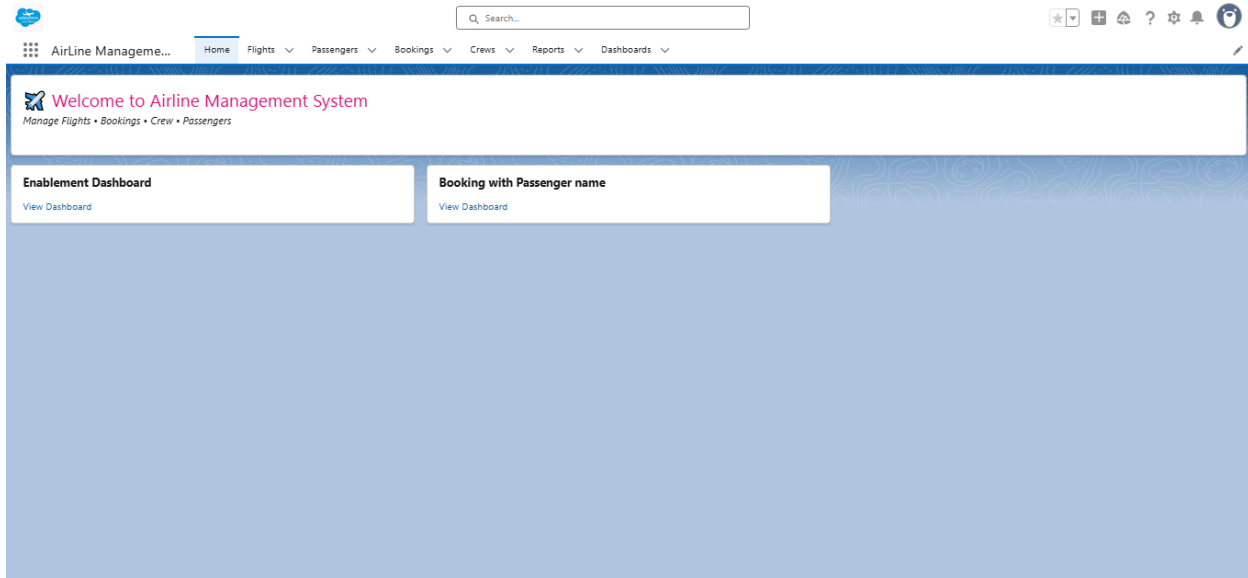
Related	Details
Booking Id	Bk-0001
Passenger Name	<a href="#">Bk-0008</a>
Departs From	Delhi
Departs To	Mumbai
Travellers	Infant
Class	Economy
Flight	<a href="#">Delta Air Lines</a>
PNR Number	7,845,643,565
Departure Date	
Created By	<a href="#">Vishal Manibabu Kasukurthi Team</a> , 6/21/2025, 10:05 AM
Owner	<a href="#">Vishal Manibabu Kasukurthi Team</a>
Last Modified By	<a href="#">Vishal Manibabu Kasukurthi Team</a> , 6/25/2025, 3:02 AM

<div>  <div> <div>Search...</div> </div> </div>							
<div> <div>AirLine Manageme...</div> <div> <div>Flights</div> <div>Passengers</div> <div>Bookings</div> <div>Crews</div> <div>Reports</div> <div>Dashboards</div> </div> </div>							
<div> <div>Report: Bookings with Flight</div> <div>Booking with Passenger Name</div> </div>							
<div> <div>Total Records</div> <div>Total Flight: PNR Number</div> <div>10</div> <div>62,765,148,520</div> </div>							
<div> <div>Travellers</div> <div> <div>Adult (7)</div> </div> </div>	<div> <div>Booking: Booking Id</div> </div>	<div> <div>Flight: Flight Name</div> </div>	<div> <div>Class</div> </div>	<div> <div>Passenger Name</div> </div>	<div> <div>Departs From</div> </div>	<div> <div>Departs To</div> </div>	<div> <div>Flight: PNR Number</div> </div>
<div> <div>Adult (7)</div> </div>	Bk-0005	Lufthansa	Business	Bk-0003	Chennai	Kolkata	7,845,643,565
	Bk-0006	Delta Air Lines	Economy	Bk-0005	Delhi	Kolkata	7,845,643,565
	Bk-0004	Air France	Business	Bk-0002	Chennai	Delhi	7,845,643,565
	Bk-0008	SpiceJet	Economy	Bk-0007	Bengaluru	Kolkata	7,845,643,565
	Bk-0003	Singapore Airlines	First Class	Bk-0010	Chennai	Kolkata	7,845,643,565
	Bk-0010	Qatar Airway	Economy	Bk-0010	Mumbai	Bengaluru	7,845,643,565
	Bk-0009	United Airlines	Economy	Bk-0012	Delhi	Mumbai	7,845,643,565
Subtotal							54,919,504,955
<div> <div>Child (1)</div> </div>	Bk-0007	Air India	Economy	Bk-0006	Bengaluru	Delhi	7,845,643,565
Subtotal							7,845,643,565
<div> <div>Infant (2)</div> </div>	Bk-0001	Delta Air Lines	Economy	Bk-0008	Delhi	Mumbai	7,845,643,565
	Bk-0002	Qatar Airway	Economy	Bk-0009	Mumbai	Bengaluru	7,845,643,565
Subtotal							15,991,287,130
Total (10)							62,765,148,520

<div>  <div> <div>Search...</div> </div> </div>							
<div> <div>AirLine Manageme...</div> <div> <div>Flights</div> <div>Passengers</div> <div>Bookings</div> <div>Crews</div> <div>Reports</div> <div>Dashboards</div> </div> </div>							
<div> <div>Report: Bookings with Flight</div> <div>Crew with Flight</div> </div>							
<div> <div>Total Records</div> <div>Total Flight: PNR Number</div> <div>10</div> <div>62,765,148,520</div> </div>							
<div> <div>Travellers</div> <div> <div>Adult (7)</div> </div> </div>	<div> <div>Booking: Booking Id</div> </div>	<div> <div>Flight: Flight Name</div> </div>	<div> <div>Departs From</div> </div>	<div> <div>Departs To</div> </div>	<div> <div>Flight: PNR Number</div> </div>	<div> <div>Passenger Name</div> </div>	<div> <div>Class</div> </div>
<div> <div>Adult (7)</div> </div>	Bk-0005	Lufthansa	Chennai	Kolkata	7,845,643,565	Bk-0003	Business
	Bk-0006	Delta Air Lines	Delhi	Kolkata	7,845,643,565	Bk-0005	Economy
	Bk-0004	Air France	Chennai	Delhi	7,845,643,565	Bk-0002	Business
	Bk-0008	SpiceJet	Bengaluru	Kolkata	7,845,643,565	Bk-0007	Economy
	Bk-0003	Singapore Airlines	Chennai	Kolkata	7,845,643,565	Bk-0010	First Class
	Bk-0010	Qatar Airway	Mumbai	Bengaluru	7,845,643,565	Bk-0010	Economy
	Bk-0009	United Airlines	Delhi	Mumbai	7,845,643,565	Bk-0012	Economy
Subtotal					54,919,504,955		
<div> <div>Child (1)</div> </div>	Bk-0007	Air India	Bengaluru	Delhi	7,845,643,565	Bk-0006	Economy
Subtotal					7,845,643,565		
<div> <div>Infant (2)</div> </div>	Bk-0001	Delta Air Lines	Delhi	Mumbai	7,845,643,565	Bk-0008	Economy
	Bk-0002	Qatar Airway	Mumbai	Bengaluru	7,845,643,565	Bk-0009	Economy
Subtotal					15,991,287,130		
Total (10)					62,765,148,520		



## ➤ Created a lighting app page



## 8. ADVANTAGES & DISADVANTAGES

### Advantages:

#### 1. Centralized Airline Data

All passenger, flight, booking, and crew records are stored in one platform, improving accessibility and operational efficiency.

#### 2. Automated Flight & Booking Tracking

The system sends automated reminders for upcoming flights, booking confirmations, and schedule changes, reducing manual follow-ups.

#### 3. Enhanced Reporting & Analytics

Real-time dashboards display key metrics such as flight occupancy, passenger trends, and booking revenue, aiding strategic decisions.

#### 4. Improved Passenger Communication

Auto-generated emails and alerts notify passengers about flight schedules, changes, cancellations, or offers—enhancing customer satisfaction.

#### 5. Regulatory Compliance & Audit Readiness

Well-documented passenger data and change logs help meet aviation compliance standards and make audit processes easier.

### Disadvantages:

#### 1. Initial Setup Complexity

Implementing the AMS requires careful data modeling, configuration, and migration, which may take time initially.

#### 2. Training for Airline Staff

Staff such as ground crew or booking agents may need hands-on training to effectively use the system's features.

#### 3. High Customization Cost

Tailoring the Salesforce platform to fit complex airline operations may involve higher development and licensing costs.

#### 4. **Dependence on Internet Access**

As a cloud-based solution, AMS needs stable internet connectivity for uninterrupted access, especially in remote areas.

#### 5. **Data Privacy Concerns**

Handling sensitive passenger information like passport data and travel history requires robust data security practices.

## 9. CONCLUSION

The **Airline Management System (AMS)** built on Salesforce streamlines airline operations by automating passenger handling, booking, flight scheduling, and crew management. It reduces manual errors, improves data accuracy, enhances customer service, and enables real-time reporting, making it a valuable tool for airline administrators and operators.

## 10. FUTURE SCOPE

### **Future Enhancements for AMS:**

#### 1. **Integration with Online Payment Gateways**

Allow passengers to pay for tickets, luggage, and onboard services using gateways like Stripe or Razorpay for real-time receipts and smoother processing.

#### 2. **Advanced Dashboards for Airline Metrics**

Add visual dashboards showing seat occupancy, flight punctuality, revenue per route, etc., using Salesforce Reports and third-party analytics.

#### 3. **SMS & WhatsApp Notifications**

Notify passengers via SMS/WhatsApp about boarding passes, flight delays, check-in windows, and gate changes.

#### 4. **Role-Based User Access**

Assign different access levels for pilots, ground crew, admins, and ticketing agents with secure, role-specific dashboards.



### 5. **Mobile App for Check-In and Status**

Provide a mobile-responsive UI or app for passengers to check in, download tickets, or track flight status.

### 6. **AI-Powered Delay Prediction**

Use AI to analyze weather, air traffic, and history to predict and alert delays before they occur.

### 7. **E-ticket Generation and Digital Signature**

Automatically generate and sign e-tickets and boarding passes with integrations like DocuSign.

### 8. **International Compliance Modules**

Support travel documentation checks (e.g., passport expiry validation, visa compliance) across global routes.

### 9. **IoT for Smart Airport Services**

Integrate smart devices for real-time baggage tracking, automated check-in kiosks, and gate security monitoring.

### 10. **Multi-Airline or Franchise Support**

Scale the system to support multiple airlines or franchise operations, each with their own data models, rules, and reports.

## 11. APPENDIX

### Apex Class: PhnValid\_PassengerObj.cls

```
public class PhnValid_PassengerObj {  
    public static void valMethod(List<Passenger__c> newPass){  
        for(Passenger__c p : newPass){  
            if(p.Phone__c == null){  
                p.Phone__c.addError('please Enter phone Number');  
            }  
        }  
    }  
}
```

```
}
```

### **Apex Trigger: PhnValidTrigger.trigger**

```
trigger PhnValidTrigger on Passenger__c (before insert) {  
    if(trigger.isBefore && trigger.isInsert){  
        PhnValid_PassengerObj.valMethod(trigger.new);  
    }  
}
```

### **Apex Test Class: PhnValid\_TestClass.cls**

```
@isTest  
public class PhnValid_TestClass {  
    @isTest  
    public static void testClass(){  
        List<Passenger__c> varlis = new List<Passenger__c>();  
        Passenger__c var = new Passenger__c();  
        var.Phone__c = null;  
        varlis.add(var);  
        insert varlis;  
        PhnValid_PassengerObj.valMethod(varlis);  
    }  
}
```

**GitHub Link:** <https://github.com/sathigrishmanjana/Airline-Management-System>

**Demo Video Link:** <https://drive.google.com/file/d/1NZZmEwhU9-PkatggvZqm7iwKn21n7LP9/view?usp=sharing>