AIRLINE MANAGEMENT SYSYTEM

Date	20-6-2025
Team ID	LTVIP2025MID29669
Project Name	Airlines Management system
Maximum Marks	

Chapter 4

4. Project Design

1. Data Model Design

Define custom objects and relationships:

Object	Key Fields	Relationships
Flight	FlightID, FlightName, Company	One-to-many with FlightSchedule
FlightSchedule	Source, Destination, Departure/Arrival, Status	Master-detail with Flight
Pilot	Name, DOB, Experience, Contact	Master-detail with Flight
PilotSchedule	Assigned FlightSchedule, Pilot1, Pilot2	Master-detail with FlightSchedule

Validation Rules:

- Source ≠ Destination
- Departure < Arrival
- Age ≥ 18 for pilots

2. User Roles & Access Control

Define hierarchy and permissions:

- **Roles**:
  - CEO (System Admin)
  - Manager (Flight oversight)
  - Operators (Schedule flights, assign pilots)
- **Sharing Rules**:
  - Flights arriving in Mumbai → Shared with Mumbai folder users

    - Flights arriving in Pune → Shared with Pune folder users
- **Public Folders**:
    - Organize flight records by destination city

3. Workflow & Automation

Use Salesforce Flow or Process Builder:

- **Flight Status Automation**:
    - On creation → Status = "Open"
    - On departure → Status = "InProgress"
    - On arrival → Status = "Closed"
- **Email Notifications**:
    - To pilots when assigned
    - To admin on flight creation, departure, arrival, or cancellation

4. Reports & Dashboards

Visualize key metrics:

- **Dashboards**:
    - Daily flight schedules
    - Monthly flight trends
    - Pie chart: Closed vs. Canceled flights
- **Reports**:
    - All flights with schedules
    - Monthly and quarterly summaries

5. Security & Compliance

- **Field-Level Security**: Hide sensitive fields from unauthorized users
- **Shield Encryption**: Protect passenger and pilot data
- **Audit Trail**: Track changes to flight and pilot records

4.1 Existing System (Problem Solution Fit)

Existing System Overview

Most traditional airline systems rely on **manual processes or legacy software**, which leads to:

- **Inefficient flight scheduling**: Manual coordination causes delays and errors
- **Limited pilot assignment tracking**: No centralized view of pilot availability or experience

- **Fragmented reservation systems**: Booking, cancellation, and passenger data are often siloed
- **Poor data visibility**: Managers struggle to generate real-time reports or dashboards
- **Security risks**: Sensitive passenger and pilot data may be exposed due to lack of encryption or access control

These limitations result in **slow response times**, **overburdened staff**, and **loss of customer satisfaction**, especially as passenger volume increases.

✅**Problem-Solution Fit with Salesforce**

Salesforce solves these problems by offering:

- **Centralized Data Model**: Custom objects for flights, pilots, schedules, and reservations
- **Automation**: Flows and workflows for status updates, notifications, and pilot assignments
- **Role-Based Access**: Secure data sharing based on user roles and destination folders
- **Real-Time Dashboards**: Visual insights into flight status, trends, and pilot schedules
- **Scalability**: Easily handles growing data and integrates with external systems like payment gateways or weather APIs3

This solution fits the problem by **streamlining operations**, **reducing manual effort**, and **enhancing customer experience** through automation and visibility.

4.2 Proposed System (Proposed Solution)

□ Key Objectives of the Proposed System

- **Automate flight scheduling and pilot assignment**
- **Streamline passenger reservations and cancellations**
- **Enable real-time notifications and status tracking**
- **Ensure secure, role-based access to sensitive data**
- **Provide actionable insights through dashboards and reports**

□ System Features & Modules

Module	Description
Flight Management	Create, schedule, and track flights with status updates (Open, InProgress, Closed, Canceled)
Pilot Management	Maintain pilot profiles, validate experience, and assign pilots to flights
Reservation System	Search flights, book tickets, manage passenger details and payments
Notification Engine	Send automated emails/SMS for flight events and pilot assignments
Reporting & Dashboards	Visualize flight trends, pilot schedules, and booking statistics

 Salesforce Implementation Highlights

- **Custom Objects**: Flight, FlightSchedule, Pilot, PilotSchedule
- **Automation**: Flows and Workflow Rules for status changes and alerts
- **Validation Rules**: Prevent illogical data entries (e.g., same source/destination)
- **Security**: Role hierarchy, public folders, and sharing rules based on destination
- **User Roles**: CEO, Manager, Operators with tailored permissions

 Benefits Over Existing System

- **Efficiency**: Reduces manual effort and speeds up operations
- **Accuracy**: Validations and automation minimize human errors
- **Visibility**: Real-time dashboards improve decision-making
- **Scalability**: Easily adapts to growing data and user base
- **Security**: Protects sensitive data with encryption and access control

4.3 Solution Architecture

 1. Architectural Layers

Layer	Components	Purpose
Presentation Layer	Salesforce Lightning UI, Mobile App	User interaction and experience
Business Logic Layer	Apex Classes, Flows, Validation Rules	Handles flight scheduling, pilot assignment, reservations
Data Layer	Custom Objects (Flight, Pilot, Reservation), SOQL	Stores and retrieves structured airline data
Integration Layer	REST/SOAP APIs, MuleSoft, Heroku	Connects with external systems (e.g., payment gateways, weather APIs)
Security Layer	Role Hierarchy, Shield Encryption, Sharing Rules	Protects sensitive data and enforces access control

□ 2. Key Salesforce Components

- **Custom Objects**: Flight, FlightSchedule, Pilot, PilotSchedule, Reservation
- **Relationships**: Master-detail and lookup for data integrity
- **Automation**: Flows, Workflow Rules for status updates and notifications
- **Reports & Dashboards**: Real-time insights into flight operations and pilot schedules

□ 3. External Integrations

- **Payment Gateway**: For ticket booking and refunds
- **Weather API**: To adjust flight schedules based on conditions
- **Email/SMS Services**: For passenger and pilot notifications
- **Heroku Services**: Optional microservices for advanced logic or AI-based recommendations

□ 4. Security Architecture

- **Role-Based Access**: CEO, Manager, Operators with tailored permissions
- **Field-Level Security**: Hide sensitive fields from unauthorized users
- **Public Folders & Sharing Rules**: Share flight records based on destination city
- **Audit Trail**: Track changes to flight and pilot records

□ 5. Scalability & Performance

- **Salesforce DX & Sandboxes**: For modular development and testing
- **Bulk API & Data Loader**: Handle large data volumes efficiently
- **Caching & Query Optimization**: Improve performance for frequent operations