

AIRLINE MANAGEMENT SYSTEM

COLLEGE NAME:AVANTHI INSTITUTE OF ENGINEERING AND
TECHNOLOGY[AVEV]

TEAM ID:LTVIP2025TMID29185

TEAM MEMBERS: KANDREGULA BANDHAVI [22Q71A0564]
ASHA BAIPILLI [22Q71A0511]
G RAMA CHANDRARAO [22Q71A0561]

AIRLINES MANAGEMENT SYSTEM

Phase1: Ideation Phase

[Brainstorm]

Date	1 June 2025
Team ID	LTVIP2025TMID29185
Project Name	AIRLINES MANAGEMENT SYSTEM
Maximum Marks	4 Marks

OBJECTIVE:

To automate and optimize various operational, administrative, and customer-facing functions of an airline such as flight booking, crew management, maintenance scheduling, and more.

KEY FEATURES:

- Real-time seat availability
- Mobile/web check-in
- Dynamic pricing based on demand
- Airport/airline alerts (delays, gate changes)
- Multilingual support
- Weather integration
- Code-share and partner airline sync
- Frequent flyer tracking and points redemption

USER ROLES & ACCESS:

- **Admin:** Full control over all modules
- **Airline Staff:** Booking agents, ground staff, engineers, etc.
- **Pilots/Crew:** Schedule, shift details, leave requests
- **Customers:** Search flights, book tickets, manage trips
- **Airport Authorities:** Verify manifests, security, air traffic updates

STAKEHOLDERS:

- Airlines and their operational teams
- Customers/passengers
- Travel agencies and aggregators
- Government and aviation regulators
- Partner airlines and service providers (e.g., hotels, transport)

STEPS REQUIRED:

1. Requirement Gathering

- Identify all stakeholders (airline staff, passengers, admin, maintenance team)
- Conduct interviews or workshops to understand system needs
- Define system scope: modules (e.g., booking, scheduling, crew, maintenance)
- Analyze existing systems (if any) for gaps and improvement areas

2. System Design

- **Architecture Design:**
 - Decide on architecture (Monolithic vs Microservices, Cloud vs On-Prem)
 - Choose technology stack (e.g., Java/Python backend, React frontend)
- **Database Design:**
 - Design ER diagrams for flights, passengers, bookings, payments, etc.
- **UI/UX Design:**
 - Create mockups/wireframes for customer and admin interfaces
 - Ensure accessibility and responsive design for all devices

3. Module Development

Each module is developed based on design specs:

- **Flight Scheduling System**
- **Passenger Booking System**
- **Crew and Staff Management**
- **Aircraft Maintenance Tracker**
- **Payment Integration**
- **Notifications & Alerts**

- **Security & Authentication System**

4. Integration

- Integrate external systems:
 - Payment gateways (e.g., Razorpay, PayPal)
 - Airport APIs for gate info, weather, air traffic data
 - Partner airlines (for code-share flights)
 - CRM tools or chatbots

5. Testing

- **Unit Testing** – for individual components
- **Integration Testing** – for module communication
- **System Testing** – for the entire system functionality
- **User Acceptance Testing (UAT)** – by airline staff and end-users
- **Performance Testing** – to handle high load during peak seasons

6. Deployment

- Set up production environment (cloud or physical servers)
- Use CI/CD pipelines for automated deployment
- Ensure rollback strategies and backup systems
- Monitor deployment logs for errors

7. Training & Documentation

- Train airline staff on using the system
- Provide help manuals and support materials
- Set up knowledge base for customers (FAQs, chatbot)

8. Launch & Go-Live

- Perform soft launch with limited users
- Fix early bugs or feedback
- Full-scale rollout with marketing and support readiness

9. Maintenance & Support

- Regular system monitoring and logs review
- Provide customer support (24/7 helpline, email/chat support)
- Fix bugs, release updates, and security patches
- Ensure compliance with aviation regulations (e.g., DGCA, FAA)

10. Continuous Improvement

- Collect user feedback (from passengers, staff)
- Analyze data for improvement (e.g., frequent delays, booking drop-offs)
- Implement new features (AI, analytics, loyalty programs)
- Scale infrastructure as users grow

Ideation Phase

[Empathize and Discover]

Date	1 June 2025
Team ID	LTVIP2025TMID29185
Project Name	AIRLINES MANAGEMENT SYSTEM
Maximum Marks	4 Marks

Objective:

To design and implement a comprehensive Airline Management System that streamlines flight operations, enhances customer experience, automates ticketing, and supports airline staff in day-to-day operations.

2. Stakeholder Identification (Empathize Stage)

Stakeholder	Needs / Goals	Pain Points
Passengers	Easy ticket booking, flight updates, flexible cancellation	Delays, complicated UI, no real-time updates
Airline Booking Agents	Fast booking, manage seats, access flight data	Manual work, errors in manifests
Flight Crew	View schedules, request time-off, receive notifications	Last-minute changes, lack of mobile access
Maintenance Engineers	Track maintenance tasks, receive alerts	Manual tracking, missing logs
Admin/Managers	Dashboard view of operations, reports, manage users	Lack of visibility, no automation
IT/Support Team	System reliability, error logs, backend access	Scalability issues, poor error handling

3. Problem Statement (Empathy to Definition Transition)

Passengers and airline staff face operational inefficiencies and fragmented systems for booking, flight management, and crew coordination, leading to delays, reduced satisfaction, and poor resource optimization.

4. Business Goals

- Increase ticket sales by improving the customer experience
- Reduce flight delays by automating crew and maintenance schedules
- Minimize operational costs through efficient resource management
- Ensure compliance with aviation regulations and improve security
- Enable data-driven decisions via analytics and reporting

5. Key Features to Discover (High-Level Scope)

Functional Area	Features to Explore
Flight Scheduling	Route planning, aircraft assignment, time zone handling
Passenger Booking	Seat selection, multi-leg journeys, loyalty programs
Crew Management	Shift rosters, certifications, performance tracking
Baggage Tracking	RFID/barcode integration, lost baggage workflows
Ticketing & Payment	Secure payment gateways, invoice generation
Maintenance Management	Aircraft service logs, scheduled maintenance alerts
Reporting & Analytics	Revenue, occupancy, delay analysis, compliance metrics
Notifications & Alerts	SMS/email push for delays, gate changes, cancellations

6. Discovery Activities

Activity	Purpose
Stakeholder Interviews	Understand pain points and goals
Journey Mapping	Visualize customer and staff interactions
Competitive Benchmarking	Study systems used by top global airlines

Brainstorming Workshops	Ideate possible features, UI/UX solutions
Data Review	Analyze ticket sales, delays, complaints, etc.

7. Initial Use Case Scenarios

- Passenger books a one-way ticket and receives email confirmation
- Airline manager schedules aircraft and assigns crew
- Crew member checks daily schedule on mobile app
- Maintenance engineer gets alert for a due inspection
- Passenger receives real-time delay notification via app

8. Success Criteria

- System handles **100K+ concurrent users** without performance drop
- **Ticket booking time < 3 minutes**
- **99.99% system uptime**
- **Automated maintenance alerts reduce unexpected delays by 30%**
- **Customer satisfaction score above 90%**

9. Tools & Technologies to Consider

- **Frontend:** React, Angular, Flutter
- **Backend:** Java, Node.js, Python (Django/Flask)
- **Database:** PostgreSQL, MongoDB, MySQL
- **Cloud:** AWS, Azure (for scalability and uptime)
- **Security:** OAuth 2.0, HTTPS, data encryption
- **Testing Tools:** Selenium, JMeter, Postman

10. Next Steps

1. Conduct stakeholder interviews & create user personas

2. Finalize feature list and prioritize with MoSCoW method
3. Develop system architecture diagram
4. Prepare wireframes for core modules
5. Create MVP backlog for Agile development

Ideation Phase [Problem Statements]

Date	1 June 2025
Team ID	LTVIP2025TMID29185
Project Name	AIRLINES MANAGEMENT SYSTEM
Maximum Marks	4 Marks

PROBLEM STATEMENT:

In the current airline industry, **manual processes, fragmented systems, and limited automation** are causing significant inefficiencies in managing core operations such as **flight scheduling, passenger booking, crew assignment, baggage handling, and maintenance tracking**.

Passengers often experience **delays, confusing interfaces, and lack of real-time updates**, while airline staff face **operational overload, poor coordination, and limited visibility** into flight and crew status. These issues lead to **lower customer satisfaction, increased operational costs, and compliance risks**.

Therefore, there is a critical need for an **integrated, user-friendly, and scalable Airline Management System** that:

- Automates end-to-end airline operations,
- Improves real-time communication,
- Enhances the passenger experience,

- And ensures regulatory compliance and operational efficiency.

REQUIREMENT ANALYSIS

[CUSTOMER JOURNEY MAP]

Creating a **Customer Journey Map** for an **Airline Management System Project** in **Salesforce** involves outlining the experience a customer has from first learning about the airline to post-flight engagement, while integrating how **Salesforce** tools support and enhance each step.

Airline Management System – Customer Journey Map

Stage	Customer Actions	Touchpoints	Salesforce Capabilities Used	Opportunities for Enhancement
1. Awareness	Searches flights, visits airline website, sees ads	Social media, Google ads, partner sites, mobile app	<i>Marketing Cloud, Advertising Studio, Web-to-Lead</i>	Personalized ad targeting, capture leads for follow-up
2. Consideration	Compares airlines, reads reviews, signs up for newsletter	Website, blog, email campaigns	<i>Marketing Cloud, Email Studio, Lead Scoring</i>	Lead nurturing with personalized offers
3. Booking	Selects flight, enters preferences, makes payment	Website, mobile app, call center	<i>Salesforce Experience Cloud, Service Cloud, Commerce Cloud</i>	Seamless booking flow with AI-based fare suggestions
4. Pre-Travel	Gets updates, checks-in online, chooses seat	Email, SMS, app notifications, chatbot	<i>Journey Builder, Einstein Bots, Mobile Studio</i>	Proactive flight updates, upsell upgrades (luggage, seat)
5. At Airport	Uses e-boarding pass, contacts support for queries	Airport kiosk, mobile app, live chat	<i>Field Service Lightning, Service Cloud Voice</i>	AI routing to reduce wait time, sentiment detection

6. In-Flight	Onboard services, feedback forms, entertainment	Inflight app/portal, email survey	<i>Service Cloud, Feedback Management, IoT integration</i>	Real-time service tracking, personalized content
7. Post-Flight	Feedback, loyalty program engagement, future bookings	Email, loyalty app, survey forms	<i>Salesforce Loyalty Management, Marketing Cloud, CDP</i>	Reward points, next-flight promo emails
8. Advocacy	Shares experience, refers friends	Social media, reviews, referral links	<i>Social Studio, Referral Management, CRM Analytics</i>	Track NPS, incentivize referrals

Salesforce Tools Breakdown

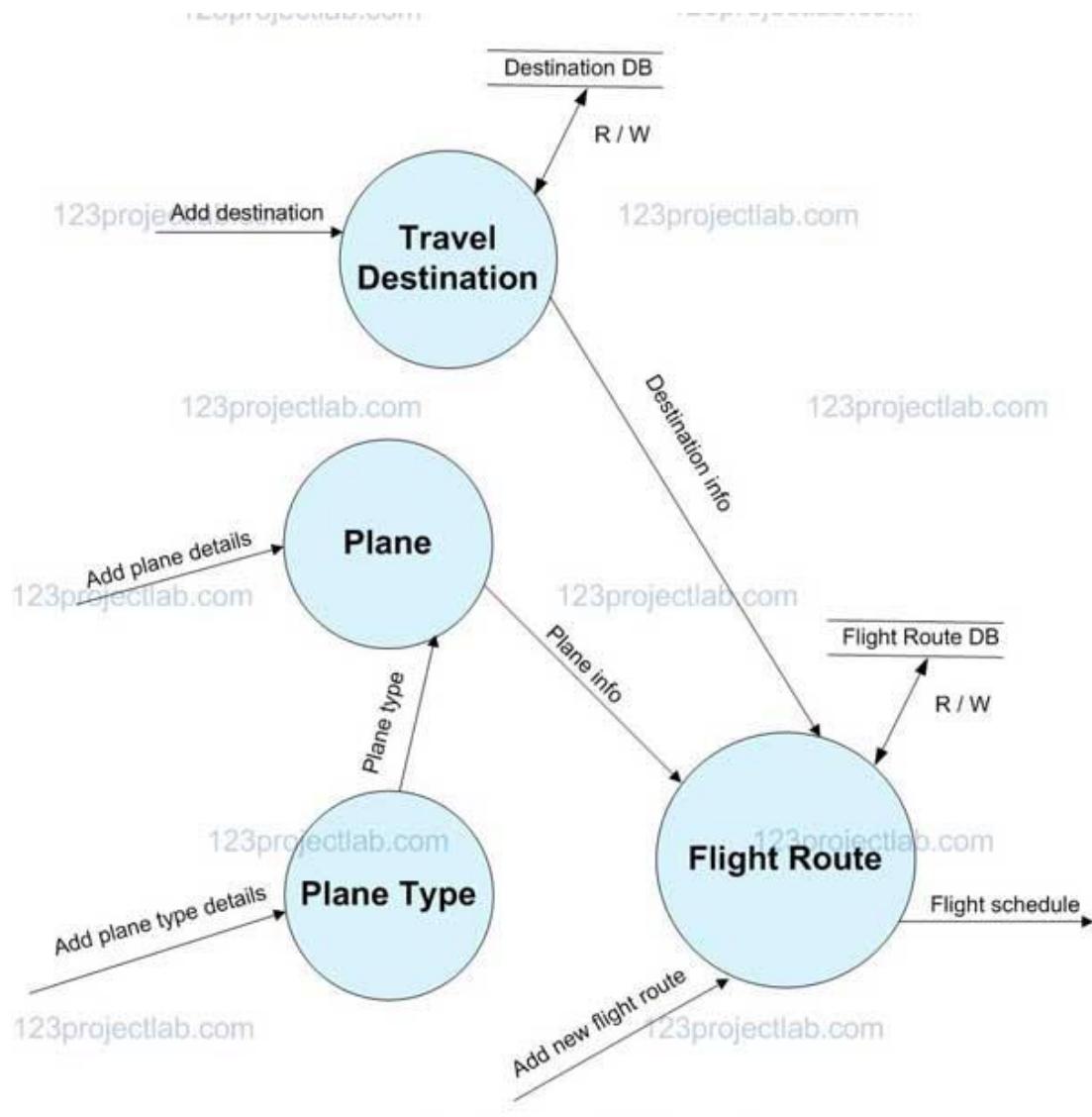
- **Marketing Cloud** – Automate email/SMS journeys, personalize communication.
- **Service Cloud** – Omnichannel support (chat, voice, email).
- **Experience Cloud** – Build branded customer portals.
- **Loyalty Management** – Build/customize loyalty tiers and promotions.
- **Salesforce CDP (Customer Data Platform)** – Unified 360° customer profile.
- **Einstein AI** – Predictive recommendations, AI chatbots.
- **CRM Analytics** – Visualize customer satisfaction and booking trends.

Tips for Implementation

- **Data Integration:** Integrate Salesforce with flight operations databases.
- **Mobile First:** Optimize the journey for mobile users.
- **Omnichannel Service:** Use Salesforce for consistent support across all touchpoints.
- **Feedback Loops:** Capture insights at every stage using Salesforce Surveys.

REQUIREMENT ANALYSIS

[DATA FLOW DIAGRAM]



Level 2.1 DFD for Airlines Reservation System

REQUIREMENT ANALYSIS

[SOLUTION REQUIREMENTS]

Solution Requirements for Airline Management System in Salesforce

1. Functional Requirements

These are the core features the system must support.

◊ *Customer Management*

- Register and manage customer profiles
- View customer travel history and preferences
- Manage frequent flyer status and loyalty points
- Link multiple passenger profiles under one customer account (e.g. family)

◊ *Flight Booking & Management*

- Search for flights based on origin, destination, dates
- Book and manage flight reservations (one-way, round-trip, multi-city)
- Modify or cancel bookings
- Display dynamic pricing and seat availability

◊ *Check-In and Boarding*

- Enable online/mobile check-in
- Assign or change seats during check-in
- Generate e-boarding passes and notify customers

◊ Loyalty & Rewards

- Enroll customers in loyalty programs
- Accumulate and redeem points for flights or services
- View tier benefits, history, and upcoming tier upgrades

◊ Customer Support

- Support via chatbot, email, phone, and in-app messaging
- Escalate to live agents via Service Cloud
- Log and track support tickets and feedback

◊ Notifications & Alerts

- Notify customers via SMS/email for booking confirmations, flight delays, cancellations
- Pre-boarding and check-in reminders

◊ Payments & Invoicing

- Integrate with secure third-party payment gateways
- Generate invoices and send payment confirmations
- Handle refunds and cancellations

◊ Feedback & Surveys

- Send post-flight surveys
- Collect NPS (Net Promoter Score) and satisfaction metrics

2. Non-Functional Requirements

Performance

- System should support concurrent users during peak times (e.g., holiday travel)
- Response time < 3 seconds for search and booking

Scalability

- Ability to scale up to support new markets or regions
- Modular design for future feature expansion

Security

- Role-based access control for internal users
- GDPR and PCI-DSS compliance
- Encrypted data at rest and in transit

Availability

- 99.9% system uptime with Salesforce SLA
- Real-time sync of customer and booking data

Integration

- Integrate with third-party systems:
 - Global Distribution Systems (GDS)
 - Payment gateways (e.g. Stripe, PayPal)
 - Baggage handling or airport systems
 - Airline ERP systems

3. Salesforce-Specific Requirements

Salesforce Clouds to Use

- **Service Cloud:** Omnichannel customer support and case management
- **Marketing Cloud:** Personalized journey automation and promotions
- **Experience Cloud:** Self-service customer portal and mobile site
- **Loyalty Management:** Loyalty tiers, point tracking, and rewards
- **CRM Analytics:** Track KPIs like churn, revenue, customer satisfaction
- **Salesforce CDP:** Unified customer profile across departments

Automation & Intelligence

- Use **Einstein AI** for:

- Predictive offers
- Chatbot support
- Intelligent case routing
- Use **Journey Builder** for email/SMS workflows based on triggers

Data Sync & APIs

- REST/SOAP APIs for syncing flight schedules and inventory
- Real-time sync with external GDS (e.g., Sabre, Amadeus)

Audit & Compliance

- Use Salesforce Shield for audit trails, data retention, and encryption

REQUIREMENT ANALYSIS [TECHNOLOGY STACK]

Technology Stack for Airline Management System

1. Salesforce Core Stack

Component	Technology	Purpose
-----------	------------	---------

CRM Platform	Salesforce Core (Sales, Service, Experience Cloud)	Customer data, booking & support management
Marketing Automation	Salesforce Marketing Cloud	Email/SMS campaigns, journey automation
Customer Service	Salesforce Service Cloud + Omnichannel	Ticketing, chatbots, agent routing
Self-Service Portal	Salesforce Experience Cloud	Customer-facing portal for booking & account
Loyalty Management	Salesforce Loyalty Management	Tier system, rewards, redemptions
Analytics	CRM Analytics (Tableau CRM)	Dashboards, customer insights, trends
Data Unification	Salesforce CDP	Real-time unified customer profile
Security & Auditing	Salesforce Shield	Field-level encryption, audit trail, monitoring

2. Front-End Technologies

Component	Technology	Purpose
Web Portal	Lightning Web Components (LWC), HTML5, CSS3, JS	Responsive booking and check-in interfaces
Mobile App	React Native or Flutter (with Salesforce Mobile SDK)	Flight booking, loyalty, and real-time updates
CMS	Salesforce CMS or Headless CMS (e.g., Contentful)	Content delivery for promotions and pages
Integration		

3. Integration & Middleware

Purpose	Technology	Use Case
API Integration	MuleSoft Anypoint Platform	Connect to GDS (Sabre, Amadeus), payment gateways, airport systems
Real-Time Sync	Platform Events, Change Data Capture	Update flight schedules, seat availability
Legacy System Integration	REST/SOAP APIs, ESB	Connect with older airline ERP or reservation systems

External Payment	Stripe, PayPal, Razorpay (via MuleSoft or native API)	Secure customer transactions
-------------------------	---	------------------------------

4. Data & Storage

Type	Technology	Purpose
Primary Data Store	Salesforce Data Cloud / Platform	Customer data, bookings, preferences
Data Warehouse	Snowflake, AWS Redshift (optional external DW)	Analytical data at scale
Backup & Archival	AWS S3, Salesforce Shield, MuleSoft connectors	Long-term retention and auditing

5. Security & Identity Management

Purpose	Technology	Use Case
SSO & Identity	Salesforce Identity + SAML/OAuth2	Customer and agent login with SSO
Field-Level Security	Salesforce Shield	PII and payment info protection
Access Control	Salesforce Profiles & Roles	Internal and external access management

6. AI & Automation

Purpose	Technology	Use Case
AI Predictions	Salesforce Einstein	Upsell recommendations, case deflection
Chatbots	Einstein Bots	24/7 automated customer support
Workflow Automation	Flow Builder, Process Builder	Booking confirmations, alerts, case triggers
Journey Automation	Journey Builder	Marketing flows based on user behavior

7. Monitoring & DevOps

Tool	Purpose
Salesforce DevOps Center / Copado	CI/CD pipelines for metadata and code
DataDog / New Relic	App performance and real-time monitoring
Salesforce Event Monitoring	Track user interactions, security incidents

Optional Enhancements

- **IoT Integration** (e.g., for baggage tracking): Salesforce IoT Cloud or external platform
- **Voice Support**: Amazon Connect + Salesforce Service Cloud Voice
- **BI Tool Integration**: Tableau, Power BI (via CRM Analytics connector)

PROJECT DESIGN PHASE

[PROBLEM-SOLUTION FIT]

Problem (Customer or Operational Pain Points):

- 1. Poor Customer Experience:**
 - a. Disconnected channels (call center, website, app).
 - b. Delays in support and resolution (e.g., lost baggage, flight changes).
 - c. No personalized offers or communication.
- 2. Manual, Fragmented Operations:**
 - a. Reservation and support systems not integrated.
 - b. Flight, crew, and passenger data scattered across silos.
 - c. Lack of real-time status updates for staff and passengers.
- 3. Limited Visibility & Analytics:**
 - a. Inability to track customer journey end-to-end.
 - b. Poor insights into sales, cancellations, service trends.
- 4. High Customer Attrition:**
 - a. Low loyalty due to generic service.
 - b. Loyalty programs underutilized or not data-driven.

Solution (Using Salesforce Cloud Products):

- 1. Customer 360 View with Salesforce CRM:**
 - a. Integrate reservation, support, and loyalty program data to form a unified profile.
 - b. Deliver consistent support and marketing across all channels (phone, email, SMS, app).
- 2. Service Cloud for Passenger Support:**
 - a. Omnichannel support with AI-powered chatbots and case routing.
 - b. Automate common requests: rebookings, refunds, baggage claims.
- 3. Sales Cloud for Revenue Optimization:**
 - a. Manage ticket sales, upselling (seat upgrades, meals), and partner offers.
 - b. Integrate with travel agents and B2B booking platforms.

4. **Marketing Cloud for Personalization:**
 - a. Use passenger data to tailor promotions, re-engagement campaigns, and loyalty offers.
 - b. Automate travel reminders, gate changes, or feedback surveys.
5. **Tableau CRM / Einstein Analytics:**
 - a. Track KPIs like on-time performance, service levels, customer lifetime value.
 - b. Predict cancellations, no-shows, and satisfaction trends using AI.
6. **Integration with Flight Systems via MuleSoft:**
 - a. Connect legacy flight systems (e.g., SABRE, Amadeus) to Salesforce for real-time updates.

Fit (How This Delivers Business Value):

- **Boosts customer satisfaction (CSAT, NPS)** with faster, personalized support.
- **Increases loyalty and repeat bookings** through tailored engagement.
- **Reduces operational costs** by automating service and integrating data.
- **Improves decision-making** through real-time insights across business units.
- **Accelerates digital transformation** with scalable cloud architecture.

PROJECT DESIGN PHASE

[PROPOSED SOLUTION]

Objective:

To modernize airline operations and customer engagement by implementing a centralized, intelligent, and scalable management system on Salesforce—enhancing customer experience, streamlining services, and increasing operational efficiency.

Solution Overview

We propose a **modular solution** built on Salesforce, integrating the following key components:

1. Passenger Relationship Management (PRM) – Salesforce Sales Cloud

- Manage passenger data (personal details, booking history, preferences).
- Enable sales automation for ticketing, upgrades, and ancillary services.
- Integration with booking engines (Amadeus, SABRE, etc.) via APIs.

2. Customer Support System – Salesforce Service Cloud

- **Omnichannel Case Management:** Phone, email, SMS, social, WhatsApp.
- **AI Chatbot (Einstein Bots)** to handle common queries (e.g., baggage status, rebooking).
- Automated SLA tracking, escalations, and workflow approvals.

3. Flight Disruption Handling & Rebooking Workflow

- Proactive alerts via SMS/email in case of flight delays/cancellations.
- Self-service rebooking via Salesforce Experience Cloud (passenger portal).
- Case automation for compensation claims and refunds.

4. Marketing & Loyalty Engagement – Salesforce Marketing Cloud

- Personalized offers based on travel history, preferences, and status.
- Loyalty program automation: tier upgrades, point tracking, member campaigns.
- Customer journeys for abandoned bookings, frequent travelers, VIPs.

5. Analytics & Reporting – Tableau CRM (Einstein Analytics)

- Real-time dashboards for:
 - Bookings and cancellations
 - Customer service metrics (CSAT, resolution time)
 - Flight performance and passenger load factors
- Predictive insights: customer churn, demand forecasting.

6. System Integration – MuleSoft Anypoint Platform

- Integrate Salesforce with core airline systems (PSS, DCS, GDS).
- Ensure data flow between booking, crew scheduling, baggage handling, and CRM.
- Enable real-time updates for staff and passengers.

Additional Capabilities

- **Mobile Accessibility:** Salesforce mobile app for agents and crew.
- **Experience Cloud Portal:** For passengers to manage bookings and support tickets.
- **Compliance & Security:** Adhere to GDPR, PCI DSS, and IATA data standards.

Business Benefits

-  **Customer Satisfaction:** Faster, smarter support with personalized experiences.

-  **Operational Efficiency:** Reduce manual tasks, unify systems, automate workflows.
-  **Revenue Growth:** Upselling, cross-selling, and retention of loyal customers.
-  **Data-Driven Decisions:** Leverage AI and analytics to drive continuous improvement.

PROJECT DESIGN PHASE

[SOLUTION ARCHITECTURE]

1. Core Architectural Layers

◊ *Presentation Layer (User Interfaces)*

- **Customer Interfaces:**
 - Salesforce Experience Cloud (Passenger Portal)
 - Mobile App / Airline Website
- **Employee Interfaces:**
 - Salesforce Console (Support Agents, Sales Reps)
 - Mobile App (Flight Crew, Ground Staff)

Application Layer (Salesforce Clouds & Logic)

- **Sales Cloud:**
 - Manage passenger profiles
 - Ticket sales, upgrades, partner deals
- **Service Cloud:**
 - Customer support (cases, live chat, email, phone)
 - Omni-Channel routing & SLA tracking
 - AI Chatbot (Einstein Bots) for self-service
- **Marketing Cloud:**
 - Targeted campaigns (email/SMS/push)
 - Journey Builder for lifecycle marketing
 - Loyalty program automation
- **Experience Cloud:**
 - Passenger self-service portal (check booking, raise requests, update info)
- **Tableau CRM (Einstein Analytics):**
 - Reports and dashboards (revenue, service KPIs, flight metrics)
 - Predictive analytics (churn, delays, demand)
- **MuleSoft Anypoint Platform:**

- API integration with external systems

Integration Layer

Using **MuleSoft**, connect Salesforce to:

External System	Integration Purpose
Passenger Service System (PSS)	Booking, check-in, seat assignment
Global Distribution System (GDS)	Flight availability, pricing, itinerary sync
Departure Control System (DCS)	Gate info, boarding, flight manifests
ERP / Billing System	Ticketing, invoices, refunds, revenue tracking
Baggage Tracking System	Real-time baggage status updates
Loyalty Program Engine	Points balance, tier upgrades

APIs: REST/SOAP, real-time (webhooks) + batch sync

Data Layer

- **Salesforce Data Model:**
 - Standard Objects: Accounts (Passengers), Contacts, Cases, Opportunities (Bookings), Products (Flights/Upgrades)
 - Custom Objects: Flight Schedules, Baggage Claims, Frequent Flyer Records, Crew Assignments
- **External Systems / Data Lakes:**
 - For historical flight data, large-scale analytics, and regulatory reporting

Security Architecture

- Role-based access (e.g., agents, crew, marketing team)
- Two-Factor Authentication (2FA) for internal users
- Shield Platform Encryption for sensitive data (passport, payment info)

- GDPR and PCI-DSS compliant data handling
- Audit Trails and Field History Tracking

Workflow & Automation Examples

- **Case Automation:** Delayed baggage auto-opens a case with PSS integration
- **Flight Cancellation:** Trigger email/SMS + self-service rebooking via portal
- **Loyalty Upsell:** Frequent flyers auto-added to upgrade campaigns in Marketing Cloud
- **Agent Assist:** Suggest next best action using Einstein AI during live support

PROJECT DESIGN PHASE

[PROBLEM-SOLUTION FIT]

Problem (Customer or Operational Pain Points):

- 1. Poor Customer Experience:**
 - a. Disconnected channels (call center, website, app).
 - b. Delays in support and resolution (e.g., lost baggage, flight changes).
 - c. No personalized offers or communication.
- 2. Manual, Fragmented Operations:**
 - a. Reservation and support systems not integrated.
 - b. Flight, crew, and passenger data scattered across silos.
 - c. Lack of real-time status updates for staff and passengers.
- 3. Limited Visibility & Analytics:**
 - a. Inability to track customer journey end-to-end.
 - b. Poor insights into sales, cancellations, service trends.
- 4. High Customer Attrition:**
 - a. Low loyalty due to generic service.
 - b. Loyalty programs underutilized or not data-driven.

Solution (Using Salesforce Cloud Products):

- 1. Customer 360 View with Salesforce CRM:**
 - a. Integrate reservation, support, and loyalty program data to form a unified profile.
 - b. Deliver consistent support and marketing across all channels (phone, email, SMS, app).
- 2. Service Cloud for Passenger Support:**
 - a. Omnichannel support with AI-powered chatbots and case routing.
 - b. Automate common requests: rebookings, refunds, baggage claims.
- 3. Sales Cloud for Revenue Optimization:**
 - a. Manage ticket sales, upselling (seat upgrades, meals), and partner offers.
 - b. Integrate with travel agents and B2B booking platforms.

4. **Marketing Cloud for Personalization:**
 - a. Use passenger data to tailor promotions, re-engagement campaigns, and loyalty offers.
 - b. Automate travel reminders, gate changes, or feedback surveys.
5. **Tableau CRM / Einstein Analytics:**
 - a. Track KPIs like on-time performance, service levels, customer lifetime value.
 - b. Predict cancellations, no-shows, and satisfaction trends using AI.
6. **Integration with Flight Systems via MuleSoft:**
 - a. Connect legacy flight systems (e.g., SABRE, Amadeus) to Salesforce for real-time updates.

Fit (How This Delivers Business Value):

- **Boosts customer satisfaction (CSAT, NPS)** with faster, personalized support.
- **Increases loyalty and repeat bookings** through tailored engagement.
- **Reduces operational costs** by automating service and integrating data.
- **Improves decision-making** through real-time insights across business units.
- **Accelerates digital transformation** with scalable cloud architecture.

PROJECT DESIGN PHASE

[PROPOSED SOLUTION]

Objective:

To modernize airline operations and customer engagement by implementing a centralized, intelligent, and scalable management system on Salesforce—enhancing customer experience, streamlining services, and increasing operational efficiency.

Solution Overview

We propose a **modular solution** built on Salesforce, integrating the following key components:

1. Passenger Relationship Management (PRM) – Salesforce Sales Cloud

- Manage passenger data (personal details, booking history, preferences).
- Enable sales automation for ticketing, upgrades, and ancillary services.
- Integration with booking engines (Amadeus, SABRE, etc.) via APIs.

2. Customer Support System – Salesforce Service Cloud

- **Omnichannel Case Management:** Phone, email, SMS, social, WhatsApp.
- **AI Chatbot (Einstein Bots)** to handle common queries (e.g., baggage status, rebooking).
- Automated SLA tracking, escalations, and workflow approvals.

3. Flight Disruption Handling & Rebooking Workflow

- Proactive alerts via SMS/email in case of flight delays/cancellations.
- Self-service rebooking via Salesforce Experience Cloud (passenger portal).
- Case automation for compensation claims and refunds.

4. Marketing & Loyalty Engagement – Salesforce Marketing Cloud

- Personalized offers based on travel history, preferences, and status.
- Loyalty program automation: tier upgrades, point tracking, member campaigns.
- Customer journeys for abandoned bookings, frequent travelers, VIPs.

5. Analytics & Reporting – Tableau CRM (Einstein Analytics)

- Real-time dashboards for:
 - Bookings and cancellations
 - Customer service metrics (CSAT, resolution time)
 - Flight performance and passenger load factors
- Predictive insights: customer churn, demand forecasting.

6. System Integration – MuleSoft Anypoint Platform

- Integrate Salesforce with core airline systems (PSS, DCS, GDS).
- Ensure data flow between booking, crew scheduling, baggage handling, and CRM.
- Enable real-time updates for staff and passengers.

Additional Capabilities

- **Mobile Accessibility:** Salesforce mobile app for agents and crew.
- **Experience Cloud Portal:** For passengers to manage bookings and support tickets.
- **Compliance & Security:** Adhere to GDPR, PCI DSS, and IATA data standards.

Business Benefits

-  **Customer Satisfaction:** Faster, smarter support with personalized experiences.

-  **Operational Efficiency:** Reduce manual tasks, unify systems, automate workflows.
-  **Revenue Growth:** Upselling, cross-selling, and retention of loyal customers.
-  **Data-Driven Decisions:** Leverage AI and analytics to drive continuous improvement.

PROJECT DESIGN PHASE

[SOLUTION ARCHITECTURE]

1. Core Architectural Layers

◊ *Presentation Layer (User Interfaces)*

- **Customer Interfaces:**
 - Salesforce Experience Cloud (Passenger Portal)
 - Mobile App / Airline Website
- **Employee Interfaces:**
 - Salesforce Console (Support Agents, Sales Reps)
 - Mobile App (Flight Crew, Ground Staff)

Application Layer (Salesforce Clouds & Logic)

- **Sales Cloud:**
 - Manage passenger profiles
 - Ticket sales, upgrades, partner deals
- **Service Cloud:**
 - Customer support (cases, live chat, email, phone)
 - Omni-Channel routing & SLA tracking
 - AI Chatbot (Einstein Bots) for self-service
- **Marketing Cloud:**
 - Targeted campaigns (email/SMS/push)
 - Journey Builder for lifecycle marketing
 - Loyalty program automation
- **Experience Cloud:**
 - Passenger self-service portal (check booking, raise requests, update info)
- **Tableau CRM (Einstein Analytics):**
 - Reports and dashboards (revenue, service KPIs, flight metrics)
 - Predictive analytics (churn, delays, demand)
- **MuleSoft Anypoint Platform:**

- API integration with external systems

Integration Layer

Using **MuleSoft**, connect Salesforce to:

External System	Integration Purpose
Passenger Service System (PSS)	Booking, check-in, seat assignment
Global Distribution System (GDS)	Flight availability, pricing, itinerary sync
Departure Control System (DCS)	Gate info, boarding, flight manifests
ERP / Billing System	Ticketing, invoices, refunds, revenue tracking
Baggage Tracking System	Real-time baggage status updates
Loyalty Program Engine	Points balance, tier upgrades

APIs: REST/SOAP, real-time (webhooks) + batch sync

Data Layer

- **Salesforce Data Model:**
 - Standard Objects: Accounts (Passengers), Contacts, Cases, Opportunities (Bookings), Products (Flights/Upgrades)
 - Custom Objects: Flight Schedules, Baggage Claims, Frequent Flyer Records, Crew Assignments
- **External Systems / Data Lakes:**
 - For historical flight data, large-scale analytics, and regulatory reporting

Security Architecture

- Role-based access (e.g., agents, crew, marketing team)
- Two-Factor Authentication (2FA) for internal users
- Shield Platform Encryption for sensitive data (passport, payment info)

- GDPR and PCI-DSS compliant data handling
- Audit Trails and Field History Tracking

Workflow & Automation Examples

- **Case Automation:** Delayed baggage auto-opens a case with PSS integration
- **Flight Cancellation:** Trigger email/SMS + self-service rebooking via portal
- **Loyalty Upsell:** Frequent flyers auto-added to upgrade campaigns in Marketing Cloud
- **Agent Assist:** Suggest next best action using Einstein AI during live support

PROJECT EXECUTABLE FILES

FINAL PROJECT FILES

Hardware Required:

A laptop or computer with internet connection

Software Required:

Salesforce account or org

This project aims to enhance the efficiency and effectiveness of managing flights, reservations, and passenger information. The system enables airlines to manage their fleet, schedule flights, allocate seats, and handle bookings seamlessly. It provides functionalities for ticket reservations, seat availability checks, passenger check-ins, and baggage handling. Additionally, the system facilitates communication between airlines, airports, and passengers through automated notifications and alerts. With its user-friendly interface and robust database management, the Airlines Management System optimizes workflow, improves customer satisfaction, and ensures smooth operations for the entire airline industry.

Use Case:

As a new Administrator, you perform user management tasks like creating and editing users, resetting passwords, granting permissions, configuring data access, and much more. In this unit, you will learn about users and how you add users to your Salesforce org.

A profile is a group/collection of settings and permissions that define what a user can do in Salesforce. Profile controls "Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges. You can define profiles by the user's job function. For example System Administrator, Developer, Sales Representative.

Types of profiles in Salesforce

1. Standard profiles:

By default salesforce provides below standard profiles.

- Contract Manager
- Read Only
- Marketing User
- Solutions Manager
- Standard User
- System Administrator.

We cannot deleted standard ones

Each of these standard ones includes a default set of permissions for all of the standard objects available on the platform.

2. Custom Profiles:

Custom ones defined by us.

They can be deleted if there are no users assigned with that particular one.

Use Case:

Great work Admin, you have done so good till now. The CEO wants you to differentiate the users based on their functionalities, position and based on this those users need to have the minimum access to the database object in the organization. Now it's time to use your Admin skills to focus on the users, their functionality and position in the organization in order to achieve the CEO requirements.

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

Use Case:

You have successfully fulfilled the 1st requirement i.e., differentiating the users based on the functionality. Now comes the 2nd task of differentiating the users based on their position, using your excellent admin skills and expanding the custom roles for the organization and assigning it to the users.

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access. Each user account contains at least the following:

- Username
- Email Address
- User's First Name (optional)
- User's Last Name
- Alias
- Nickname
- License
- Profile
- Role (optional)

Use Case:

TheSmartBridge is all set to move with the Salesforce platform. As this platform is very new to the employees in the organization it's up to you to enlight every employee in it.

REPORTS

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

Types of Reports in Salesforce

1. Tabular
2. Summary
3. Matrix
4. Joined Reports

Use Case:

The CEO of an organization wants to have a brief data of all the 4 objects. So he can have a clear picture of his organization and be able to make any decisions required based on this data. So he calls you on this task and wants you to represent the data in an appropriate way.

Let's create a Report.

DASHBOARDS

Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you've gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

Use Case:

As an Admin for the organization you keep pushing yourself to reach out the business requirements to take the organization to peak heights and all your superiors are very much impressed with your efforts and work dedication. In addition with reports you make an ease for the CEO in viewing the reports with data visualization. So he doesn't have to search for the data he wants during the meetings.

APEX

Apex Overview

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Lightning platform server in conjunction with calls to the Lightning Platform API. Using syntax that looks like Java and acts like database stored procedures, Apex enables developers to add business logic to most system events, including button clicks, related record updates, and Visualforce pages. Apex code can be initiated by Web service requests and from triggers on objects.

It is as similar as java i.e, it also supports OOP(Object oriented programming) like Classes, objects, methods.

Use Case 1:

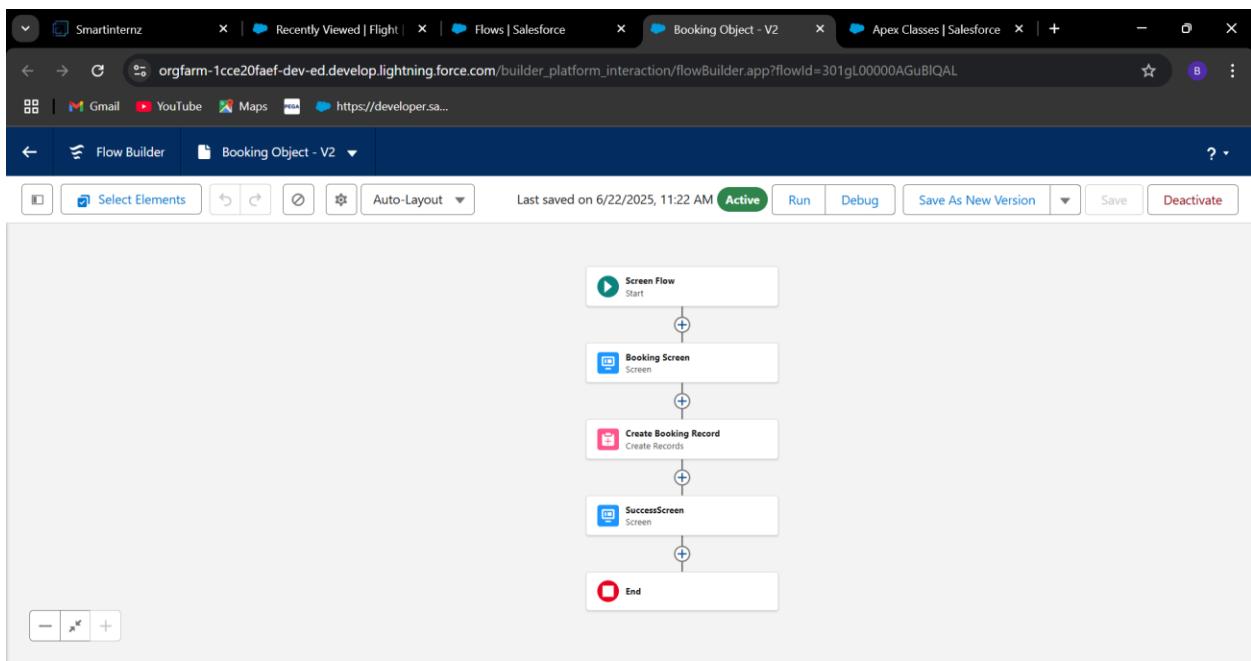
Scenario: The Airline Management wants to make the passengers phone field as a mandatory field. So whenever a record gets inserted in a passenger's object that record should not get saved into the database, if the user missed the phone field.

FLOWS

Use Case :

A sales representative needs to create a new booking for a customer who has expressed interest in a specific flight service. The representative wants to ensure that all relevant information is captured accurately and consistently.

OUTPUT:



FUNCTIONAL AND PERFORMANCE TESTING

1. Introduction

1.1 Purpose

This document outlines the performance testing strategy for the Airline Management System built on the Salesforce platform. The purpose is to ensure the system performs optimally under expected and peak load conditions, delivering a seamless experience to users including airline staff, travel agents, and customers.

1.2 Scope

Performance testing will cover major modules:

- Flight Booking
- Ticket Cancellation
- Flight Search
- Loyalty Program Management
- Check-in and Boarding Pass Generation
- Admin Reporting Dashboards

Salesforce components under test:

- Lightning Web Components (LWC)
- Apex Controllers
- Salesforce APIs
- Third-party Integrations

2. Performance Testing Objectives

- Validate system response times for critical transactions.
- Identify bottlenecks under load.
- Ensure system stability during high traffic.
- Confirm scalability across regions and user roles.

3. Test Environment

Component	Description
Platform	Salesforce Lightning Experience
Edition	Salesforce Enterprise
Tools Used	JMeter, BlazeMeter, Salesforce Debug Logs, Developer Console
Test Data	100k+ mock users, flight records, and booking scenarios
Integration Services	Payment Gateway, Airline Scheduling APIs, Email Services

4. Types of Performance Testing

Type	Purpose
Load Testing	Check performance under expected user load
Stress Testing	Evaluate system behavior under extreme load
Soak Testing	Test system stability over extended periods
Spike Testing	Analyze reaction to sudden traffic spikes
Scalability Testing	Verify response with increasing user base

5. Key Performance Metrics

Metric	Target
Response Time	< 3 seconds for 95% of transactions
Throughput	1000 transactions/min under peak load

Error Rate	< 1%
CPU/Memory Usage	Within Salesforce org limits
API Latency	< 2 seconds per call (on average)
Transaction Success Rate	≥ 99.5%

6. Test Scenarios

Scenario	Description	Load
Flight Search	500 concurrent users search flights	Load
Booking Transaction	300 users book flights with payment	Load
Cancellation Process	100 users cancel bookings	Stress
Loyalty Redemption	200 users redeem points	Spike
Dashboard Load	50 admins access reports	Soak

7. Test Data Requirements

- Users: Regular passengers, Frequent Flyers, Admins
- Flights: 1,000+ records across various time zones
- Bookings: Historical and upcoming booking data
- Rewards: Loyalty balances and tier statuses

8. Test Execution Plan

8.1 Load Testing

- Simulate 1,000 users using JMeter or BlazeMeter
- Monitor page load and API response time

8.2 Stress Testing

- Gradually increase user load until system failure
- Identify breaking point and memory usage

8.3 Soak Testing

- Run normal load for 8+ hours
- Check for performance degradation

9. Monitoring and Reporting Tools

Tool	Purpose
Salesforce Debug Logs	Backend performance monitoring
JMeter/BlazeMeter	Simulate load, analyze response
Developer Console	Real-time performance monitoring
Salesforce Optimizer	Org performance recommendations

10. Risks and Mitigation

Risk	Mitigation
Governor Limits Breach	Optimize Apex and SOQL queries
API Rate Limits	Implement caching and retry logic
Integration Timeout	Use asynchronous calls or queues
Data Skew	Test with realistic, distributed data sets

11. Exit Criteria

- $\geq 95\%$ test cases passed under expected load
- Response time under 3 seconds for key operations
- No critical or high defects remain open
- Performance reports signed off by QA Lead

12. Conclusion

This performance testing document aims to ensure the Airline Management System on Salesforce can handle large volumes of users and data with reliability and speed, maintaining high user satisfaction and operational efficiency.

To demonstrate testing for an **Airlines Management System** built on **Salesforce**, you should follow a structured approach that covers the **functional, integration, UI, security, and performance aspects** of the application. Here's a step-by-step guide tailored for **Salesforce-based testing**, with examples relevant to airline operations (like bookings, check-ins, loyalty, etc.).

Define the Testing Scope

Focus on key modules of the airline system:

- **Booking & Ticketing**
- **Flight Scheduling**
- **Customer Profiles / Loyalty Program**
- **Check-in Management**
- **Payment Integration**
- **CRM / Customer Support**

Prepare the Test Environment

- Use **Salesforce Sandbox** for testing.
- Set up **test data**: dummy passengers, flight schedules, fare rules.
- Ensure integration with external systems (payment gateway, airport systems, etc.) is mocked or sandboxed.

Write and Execute Test Cases

❖ Functional Test Cases

Booking Module

- TC001: Verify that a customer can book a flight from city A to city B.
- TC002: Verify error message when booking with invalid credit card.
- TC003: Verify loyalty points are credited after successful booking.

Check-in Module

- TC004: Verify online check-in is available 24 hrs before flight.
- TC005: Ensure seat selection is saved and updated in customer profile.

❖ UI Testing (Lightning Experience)

Use **Salesforce Lightning UI** testing tools like:

- **Salesforce Inspector**
- **Selenium with WebDriver**
- **Provar (Salesforce testing tool)**

Test Case:

- TC006: Ensure the "Book Now" button is visible and responsive.
- TC007: On mobile, verify responsive design for the Check-in screen.

❖ Apex Unit Testing

If there's custom logic written in Apex, write **@isTest** annotated methods to ensure coverage:

```
@isTest
private class BookingTest {
    static testMethod void testCreateBooking() {
        Booking__c booking = new Booking__c(
            Passenger__c = 'Test Passenger',
            Flight__c = 'Test Flight'
        );
        insert booking;
```

```
        System.assertEquals(null, booking.Id);  
    }  
}
```

Salesforce requires **75%+ code coverage** to deploy Apex code.

- Verify users with “Ground Staff” role cannot cancel flights.
- Ensure PII like passport numbers are encrypted at rest.

Performance Testing

Test performance using:

- Salesforce Performance Assistant
- Lightning Page Load Times

Check:

- TC010: Page load time for booking screen is < 2 seconds.
- TC011: System handles 1000 bookings/hour without timeout.
- orce)

Reporting & Validation

- Use **Salesforce Test Execution Reports**
- Track bugs via **JIRA** or built-in Salesforce Case Management
- Conduct **UAT (User Acceptance Testing)** with airline staff

OUTPUT:

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

PhnValid_PassengerObj.apxc

Code Coverage: None ▾ API Version: 58 ▾ Go To

```
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__caddError('Please Enter Phone Number');  
6             }  
7         }  
8     }  
9 }
```

Logs Tests Checkpoints Query Editor View State Progress **Problems**

Name Line Problem

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

PhnValid_PassengerObj.apxc PhnValidTrigger.apxt

Code Coverage: None API Version: 58 Go To

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

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

PhnValid_PassengerObj.apxc PhnValidTrigger.apxt PhnValid_TestClass.apxc

Code Coverage: None API Version: 58 Run Test Go To

```
1 @isTest
2 private 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 = '123456789';
8         varlis.add(var);
9         insert varlis;
10        PhnValid_PassengerObj.valMethod(varlis);
11    }
12 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

PhnValid_PassengerObj.apxc PhnValidTrigger.apxt PhnValid_TestClass.apxc

Code Coverage: None API Version: 58

Run Test Go To

```
1 @isTest
2 private 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 = '123456789';
8         varlis.add(var);
9         insert varlis;
10        PhnValid_PassengerObj.valMethod(varlis);
11    }
12 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage
✓	TestRun @ 2:15:50 pm			0	1	Class Overall 83%
✓	TestRun @ 3:00:33 pm			0	1	Percent PhnValidTrigger 100% 2/2
✓	TestRun @ 3:01:07 pm			0	1	Lines PhnValid_PassengerObj 75% 3/4
✓	TestRun @ 3:02:27 pm			0	1	
✓	TestRun @ 4:02:37 pm			0	1	