

Hotel Reservation & Guest Experience System – Full Explanation

Problem Statement

Hotels often face challenges in efficiently managing reservations, guest check-ins, room availability, and customer experiences. Traditional systems are either manual or disconnected, leading to overlapping bookings, long check-in times, poor guest communication, and lack of actionable insights. To remain competitive, hotels need a centralized system built on Salesforce to automate room booking, track availability in real-time, manage guest information, integrate payments, and provide managers with analytics dashboards. This project focuses on building a Hotel Reservation & Guest Experience System using Salesforce core features, automation, and custom development to enhance operational efficiency and customer satisfaction.

Phase 1: Problem Understanding & Industry Analysis

Goal: Understand the business needs of a hotel and translate them into Salesforce requirements.

Requirement Gathering:

- Track all rooms with availability status (Vacant, Occupied, Under Maintenance).
- Allow customers to book rooms online or via reception.
- Prevent overlapping reservations.
- Automate notifications for check-in, check-out, and cancellations.
- Generate revenue and occupancy reports.

Stakeholder Analysis:

- Admin: Manages Salesforce setup and configurations.
- Receptionist: Handles guest bookings and check-ins.
- Hotel Manager: Approves VIP/group reservations and monitors reports.
- Guest: Books rooms and provides feedback.

Business Process Mapping:

Guest requests booking → System checks availability → Reservation created → Payment confirmed → Email sent to guest → Check-in/out managed → Feedback collected.

Industry-specific Use Case Analysis:

Hotels deal with seasonal demand, room categories (Standard, Deluxe, Suite), and amenities (WiFi, Breakfast, Pool). The system should adapt to dynamic pricing and peak booking times.

Phase 2: Org Setup & Configuration

Goal: Prepare Salesforce environment for the hotel system.

- Salesforce Edition: Developer Org.
- Company Profile Setup: Hotel information, time zone, currency.
- Business Hours: 24x7 with defined shifts.
- User Setup: Create Receptionist, Manager roles.
- Profiles: Receptionist (create bookings), Manager (full access).
- Permission Sets: Extra report access.
- OWD: Rooms public read-only, Reservations private.
- Sharing Rules: Managers can see all bookings.
- Login Access Policies: Restrict by business hours.

Phase 3: Data Modeling & Relationships

Goal: Define the data structure for rooms, guests, and reservations.

Objects:

- Room (Custom): Room Number, Type, Rate, Status.
- Reservation (Custom): Check-in Date, Check-out Date, Total Amount.
- Guest (Standard Contact): Name, Email, Phone.
- Payment (Custom): Amount, Method, Status.
- Feedback (Custom): Guest, Rating, Comments.

Relationships:

- Room ↔ Reservation (Lookup).
- Reservation ↔ Guest (Lookup).
- Reservation ↔ Payment (Master-Detail).
- Guest ↔ Feedback (Lookup).

Record Types: Standard Booking vs Group Booking.

Page Layouts: Show related Reservations and Payments on Guest record.

Phase 4: Process Automation (Admin)

Goal: Automate booking and check-in/out processes.

- Validation Rules: Check-out date must be after check-in date.
- Approval Process: VIP or group bookings require Manager approval.
- Record-Triggered Flows: Update room status when reservation confirmed.
- Screen Flow: Receptionist booking form.
- Email Alerts: Booking confirmation, check-in reminders.
- Custom Notifications: Notify housekeeping when guest checks out.

Phase 5: Apex Programming (Developer)

Goal: Add advanced custom logic.

- Triggers: Prevent overlapping room reservations.
- Apex Class: ReservationService for reusable booking logic.
- Batch Apex: Update all rooms daily at midnight.
- Scheduled Apex: Send daily occupancy reports.
- Future Methods: Call payment gateway APIs asynchronously.
- Exception Handling: Handle errors during reservation conflicts.
- Test Classes: Ensure 75%+ coverage.

Phase 6: User Interface Development

Goal: Create user-friendly hotel booking UI.

- Lightning App: 'Hotel CRM App.'
- Tabs: Rooms, Reservations, Guests, Payments.
- LWC: Room Search by date and filter.
- Screen Flow: Self-check-in for guests.
- Utility Bar: Quick New Reservation.
- Dynamic Pages: Show related bookings on Guest record.

Phase 7: Integration & External Access

Goal: Connect with external systems.

- Payment Gateway Integration: Stripe, Razorpay.
- SMS Integration: Reminders for check-in/out.
- Google Maps API: Show hotel location.
- Salesforce Connect: External database for loyalty program.
- OAuth: Guest login via Experience Cloud portal.

Phase 8: Data Management & Deployment

Goal: Manage hotel data and ensure smooth deployment.

- Data Import Wizard: Load guest and room data.
- Data Loader: Bulk reservations import.
- Duplicate Rules: Prevent duplicate guest records.
- Backup: Weekly data export.
- Deployment: Change Sets or SFDX.
- Unmanaged vs Managed Packages: Use Managed if shared on AppExchange.

Phase 9: Reporting, Dashboards & Security Review

Goal: Provide insights and secure hotel system.

- Reports: Occupancy Rate, Revenue by Room Type, Guest Feedback.
- Dashboards: Daily Bookings, Revenue Trends.
- Dynamic Dashboards: Receptionist sees only their bookings.
- Security: FLS, IP restrictions, login hours.
- Audit Trail: Track changes to reservations.

Phase 10: Final Presentation & Demo

Goal: Deliver the project like a real system.

- Presentation: Problem → Solution → Benefits.
- Demo Walkthrough: Create booking, approve, check-in, check-out, feedback.
- Handoff: System design doc + user guide.
- Portfolio Showcase: Add to LinkedIn as a Salesforce project.