Healthcare Appointment & Patient Management System

Problem Statement

Hospitals often face challenges in managing patient appointments and records. Appointments are usually booked manually via phone calls or in person, leading to scheduling conflicts, missed follow-ups, and lack of visibility for both patients and doctors. Doctors cannot easily track patient history, while patients often do not receive timely reminders about visits or medication refills. Hospital administrators lack real-time insights into doctor workload and patient inflow. This leads to inefficiency, poor patient experience, and reduced operational effectiveness. A Salesforce-based Appointment & Patient Management System can automate appointment scheduling, send reminders, maintain treatment history, and provide dashboards for hospital management.

Phase 1: Problem Understanding & Industry Analysis

Understand hospital workflows, pain points, and requirements. Identify stakeholders (Doctors, Patients, Receptionists, Admin). Analyze how Salesforce CRM can digitize manual appointment booking, reduce missed follow-ups, and improve patient experience.

Phase 2: Org Setup & Configuration

Set up the Salesforce environment for the hospital system. Configure company profile, business hours, fiscal year, and hospital hierarchy. Create users (Doctors, Nurses, Receptionists), assign profiles, roles, and permission sets for data security.

Phase 3: Data Modeling & Relationships

Design the hospital database structure in Salesforce. Create custom objects (Patient, Doctor, Appointment, Treatment History). Define relationships: one patient can have many appointments, one doctor can serve many patients, and appointments connect patients with doctors.

Phase 4: Process Automation (Admin)

Automate patient workflows. Implement validation rules (no past appointment dates). Build flows to send automatic appointment confirmation/reminders via email. Assign follow-up tasks when an appointment is missed. Reduce manual intervention.

Phase 5: Apex Programming (Developer)

Develop backend logic for advanced automation. Write triggers to update patient records after appointments. Create Apex classes for bulk notifications. Use SOQL queries for doctor schedules. Implement test classes to ensure high code coverage for deployment.

Phase 6: User Interface Development

Design an intuitive hospital app interface in Salesforce. Use Lightning App Builder to create patient, doctor, and appointment tabs. Build custom record pages showing related data. Develop a Lightning Web Component (LWC) to display doctor availability in calendar format and allow patients to pick slots.

Phase 7: Integration & External Access

Enable external system integration. Connect Salesforce with SMS/email APIs for appointment reminders. Use Named Credentials for secure callouts. Provide hospital staff with mobile access to appointments and patient history using Salesforce mobile app.

Phase 8: Data Management & Deployment

Manage patient and doctor data efficiently. Use Data Import Wizard to upload initial patient records. Handle bulk uploads via Data Loader. Create duplicate rules to avoid repeated patient entries. Deploy configurations from Sandbox to Production using Change Sets or SFDX.

Phase 9: Reporting, Dashboards & Security Review

Provide hospital administrators with insights. Create reports on appointments by doctor/department, missed vs completed visits, and patient demographics. Build dashboards for performance tracking (doctor workload, hospital efficiency). Review and tighten security (Field Level Security, OWD, audit logs).

Phase 10: Final Presentation & Demo Day

Deliver a professional demo of the hospital system. Showcase end-to-end flow: booking an appointment → receiving confirmation → doctor consultation → dashboard updates. Present project architecture, challenges solved, and business benefits. Prepare handoff documentation and portfolio material.