Patient Appointment & Care Management CRM

Salesforce Project Implementation Document

Industry: Healthcare

Project Type: B2C Salesforce CRM Implementation

Target Users: Patients, Doctors, Hospital Administrators, and Support Staff

Table of Contents

Problem Statement

Hospitals and clinics face challenges in managing large volumes of patient inquiries and appointments. Current processes are often manual, leading to:
- Delayed appointment confirmations.
- Difficulty for doctors to track patient history and upcoming visits.
- Missed reminders resulting in high no-show rates.
- Lack of centralized dashboards for hospital administrators.

To address this, the hospital wants to implement a Salesforce CRM that will:
• Automate appointment booking and reminders.
• Track patient medical history and visit details.
• Enable doctors to manage their schedules effectively.
• Provide dashboards for administrators to monitor utilization and patient satisfaction.

Use Cases

1. Patient Management
- Maintain records of patients (demographics, contact, medical history).
- Prevent duplicate patient records.

2. Doctor Management
- Track doctors' specialization, working hours, and appointment schedules.
- Assign patients to doctors automatically based on specialization.

3. Appointment Scheduling
- Allow receptionists or patients to book appointments.
- Prevent double bookings for the same doctor and time slot.
- Send SMS/Email confirmations and reminders.

4. Prescription Tracking
- Store prescribed medicines and dosage linked to each appointment.

5. Reporting & Dashboards
- Doctor utilization reports.
- Appointment summary (daily, weekly, monthly).
- No-show rate and patient feedback dashboards.

Phase 1: Problem Understanding & Industry Analysis

Requirement gathering from hospital staff.
Stakeholder analysis: patients, doctors, admins.
Business process mapping: appointment → doctor allocation → visit → prescription → feedback.
Industry-specific analysis: healthcare appointment challenges.
AppExchange exploration: Health Cloud components, SMS apps.

Phase 2: Org Setup & Configuration

Salesforce Enterprise Edition (or Health Cloud).
Company profile: hospital working hours, holidays, fiscal year.
User licenses: receptionists, doctors, and admins.
Profiles, roles, permission sets.
OWD: Patient data = Private.
Sharing rules for doctors to see only their appointments.
Sandbox setup for testing.

Phase 3: Data Modeling & Relationships

Custom Objects: Patient__c, Doctor__c, Appointment__c, Prescription__c.
Relationships: Appointment → Patient & Doctor (Lookup), Prescription → Appointment (Master-Detail).
Record Types: OPD vs Emergency.
Page Layouts & Compact Layouts.
Schema Builder used to visualize.

Phase 4: Process Automation (Admin)

Validation Rule: Appointment date cannot be in the past.
Flows:
- Auto-assign doctor based on specialization.
- Auto-cancel unconfirmed appointments after 24 hours.
- **Einstein Chatbot integration:** Chatbot guides patients to book, reschedule, or cancel appointments. Collected details trigger Flows that create or modify Appointment__c records in Salesforce.
Workflow: Send email confirmation.
Approval Process: Emergency appointments require admin approval.

Phase 5: Apex Programming (Developer)

Triggers: Prevent double booking, auto-create prescription.
SOQL Queries: Fetch last 3 visits.
Batch Apex: Send reminders for next-day appointments.
Future Methods: SMS confirmations via API.
Test Classes: Achieved >75% coverage.

Phase 6: User Interface Development

Lightning App for hospital CRM.
Record Pages for doctors and patients.
Tabs for Patients, Doctors, Appointments, Prescriptions.
LWC Components:
- Patient Portal: book appointment, view prescriptions.
- Doctor Console: view appointments & patient history.
- **Einstein Chatbot:** Deployed on the hospital's Salesforce Community portal, enabling conversational booking, cancellation, and rescheduling of appointments.

Phase 7: Integration & External Access

Named Credentials for SMS provider (Twilio).
Callouts for SMS/Email.
Platform Events for emergency alerts.
OAuth for secure patient login.
Einstein Chatbot Integration: Connected with SMS/Email APIs to send real-time booking confirmations. OAuth ensures secure access for patients when interacting with the chatbot on community portals.

Phase 8: Data Management & Deployment

Data Import Wizard: patient & doctor records.
Data Loader: bulk appointment data.
Duplicate Rules: prevent duplicate patient records.
Change Sets for deployment.
VS Code + SFDX for version control.

Phase 9: Reporting, Dashboards & Security Review

Reports:
- Appointments per doctor.
- Patient visits by department.

- No-show rate.
Dashboards:
- Doctor Utilization.
- Patient Satisfaction.
- Appointment Trends.
Security:
- Field-level security.
- Session settings & Audit Trail.

Phase 10: Final Presentation & Demo Day

Pitch Presentation: Problem → Solution → Demo.
Demo Walkthrough: Booking → Confirmation → Doctor Console → Dashboard.
Feedback Collection from stakeholders.
Handoff Documentation: ER diagram, flow diagrams, deployment guide.
LinkedIn Showcase with demo video.