

Hospital Management System (HMS)

Custom Objects:

1. Patients

○ Fields:

- Patient ID (Auto Number)
- First Name (Text)
- Last Name (Text)
- Date of Birth (Date)
- Gender (Picklist)
- Contact Information (Phone, Email)
- Address (Text)
- Emergency Contact (Lookup to Contact)
- Medical History (Long Text Area)
- Current Medications (Long Text Area)
- Insurance Information (Text)

2. Appointments

○ Fields:

- Appointment ID (Auto Number)
- Patient (Lookup to Patient)
- Doctor (Lookup to Doctor)
- Date and Time (Date/Time)
- Appointment Type (Picklist: Consultation, Follow-up, Procedure)
- Status (Picklist: Scheduled, Completed, Cancelled)
- Notes (Long Text Area)

3. Medical Records

○ Fields:

- Record ID (Auto Number)
- Patient (Lookup to Patient)
- Visit Date (Date)
- Diagnosis (Text)
- Treatment (Long Text Area)
- Prescription (Long Text Area)
- Doctor (Lookup to Doctor)
- Lab Results (Long Text Area)

4. Billing

○ Fields:

- Invoice ID (Auto Number)
- Patient (Lookup to Patient)
- Appointment (Lookup to Appointment)
- Billing Date (Date)
- Total Amount (Currency)
- Status (Picklist: Unpaid, Paid, Pending Insurance)
- Payment Method (Picklist: Cash, Credit Card, Insurance)

5. Inventory

○ Fields:

- Item ID (Auto Number)

- Item Name (Text)
- Category (Picklist: Medication, Equipment, Supplies)
- Quantity (Number)
- Reorder Level (Number)
- Supplier (Text)

Data Relationships

- **Lookup Relationships:**
 - **Patients to Appointments:** Each appointment is associated with a patient.
 - **Patients to Medical Records:** Each medical record is associated with a patient.
 - **Appointments to Billing:** Each billing entry is associated with an appointment.
 - **Patients to Billing:** Each billing entry is associated with a patient.
 - **Doctors to Appointments:** Each appointment is associated with a doctor.
 - **Doctors to Medical Records:** Each medical record is associated with a doctor.
- **Master-Detail Relationships:**
 - **Appointments and Medical Records:** Medical Records can be detail records under Appointments to reflect the treatment and diagnosis for each visit.
 - **Inventory and Billing:** For cases where specific inventory items are billed to patients, a master-detail relationship can be established.

Record Types

- **Patients:**
 - Outpatient
 - Inpatient
- **Appointments:**
 - Consultation
 - Follow-up
 - Procedure
- **Medical Records:**
 - General Check-up
 - Specialist Visit
 - Emergency

Validation Rules

1. **Patients:**
 - Ensure that the Date of Birth is not a future date.
 - Ensure that the emergency contact is filled if a patient is marked as critical.
2. **Appointments:**
 - Ensure that the appointment date and time are in the future.
 - Ensure that the status is set to 'Scheduled' when an appointment is created.
3. **Medical Records:**
 - Ensure that the diagnosis field is not empty when the record is saved.
4. **Billing:**
 - Ensure that the total amount is greater than zero.
 - Ensure that the payment method is selected if the status is 'Paid'.

Process Automation

1. Appointment Reminders:

- **Objective:** Send an email reminder to patients 24 hours before their appointment.
- **Rule Criteria:**
 - Object: Appointment
 - Criteria: Appointment Date = TODAY() + 1
- **Action:**
 - Email Alert: Send reminder email to the patient.

2. Billing Status Update:

- **Objective:** Automatically update the billing status to "Overdue" if the payment is not received within 30 days.
- **Rule Criteria:**
 - Object: Billing
 - Criteria: Billing Date <= TODAY() - 30 AND Status != 'Paid'
- **Action:**
 - Field Update: Update the status to "Overdue".

Process Builder

1. Automatic Follow-Up Creation:

- **Objective:** Automatically create a follow-up appointment 7 days after a patient's appointment.
- **Trigger:** Appointment record creation or update
- **Criteria:** Status = 'Completed'
- **Action:** Create a new Appointment record with Date = Appointment Date + 7 days

2. Patient Record Update:

- **Objective:** Update the patient's "Last Visit Date" field when a medical record is created.
- **Trigger:** Medical Record creation
- **Criteria:** Always true
- **Action:** Update the Patient record's "Last Visit Date" field to the Visit Date of the Medical Record

Approval Processes

1. Treatment Plan Approval:

- **Objective:** Ensure treatment plans are reviewed and approved by a senior doctor.
- **Steps:**
 - Initial Submission: Submitted by attending doctor
 - Step 1: Approval by Senior Doctor
 - Step 2: Notification sent to the attending doctor upon approval/rejection

2. Discharge Approval:

- **Objective:** Ensure discharge plans are authorized by the head of the department.
- **Steps:**
 - Initial Submission: Submitted by attending doctor
 - Step 1: Approval by Head of Department
 - Step 2: Notification sent to the attending doctor upon approval/rejection

Flows

1. Patient Intake Flow:

- **Objective:** Guide staff through the patient intake process.
- **Steps:**
 - Screen: Collect basic patient information (Name, DOB, Contact Info)
 - Screen: Collect medical history and current medications
 - Record Create: Create a new Patient record
 - Screen: Confirm creation and provide next steps
- 2. **Appointment Scheduling Flow:**
 - **Objective:** Simplify the appointment scheduling process for staff.
 - **Steps:**
 - Screen: Select patient and doctor
 - Screen: Choose appointment date and time
 - Record Create: Create a new Appointment record
 - Screen: Confirm creation and send confirmation email

Screen Flow Requirements

1. Patient Intake Flow

Objective: Guide staff through the patient intake process, ensuring all necessary information is collected accurately.

Steps:

1. **Screen 1: Basic Information**
 - Fields: First Name, Last Name, Date of Birth, Gender, Contact Information (Phone, Email), Address
2. **Screen 2: Medical History**
 - Fields: Medical History (Long Text Area), Current Medications (Long Text Area)
3. **Screen 3: Insurance Information**
 - Fields: Insurance Provider (Text), Policy Number (Text), Coverage Details (Long Text Area)
4. **Create Records: Patient**
 - Create a new Patient record using the collected information.
5. **Screen 4: Confirmation**
 - Display a confirmation message and next steps.

Configuration:

- **Flow Name:** Patient Intake Flow
- **Type:** Screen Flow
- **Elements:**
 - **Screen 1: Basic Information**
 - Field Types: Text, Date, Phone, Email, Address
 - **Screen 2: Medical History**
 - Field Types: Long Text Area
 - **Screen 3: Insurance Information**
 - Field Types: Text, Long Text Area
 - **Create Records: Patient**
 - **Screen 4: Confirmation**

2. Appointment Scheduling Flow

Objective: Simplify the appointment scheduling process for staff, ensuring all necessary details are captured.

Steps:

1. **Screen 1: Select Patient and Doctor**
 - Fields: Patient (Lookup to Patient), Doctor (Lookup to Doctor)
2. **Screen 2: Choose Appointment Date and Time**
 - Fields: Appointment Date (Date), Appointment Time (Time), Appointment Type (Picklist)
3. **Create Records: Appointment**
 - Create a new Appointment record using the collected information.
4. **Screen 3: Confirmation**
 - Display a confirmation message and option to send a confirmation email to the patient.

Configuration:

- **Flow Name:** Appointment Scheduling Flow
- **Type:** Screen Flow
- **Elements:**
 - **Screen 1: Select Patient and Doctor**
 - Field Types: Lookup (Patient, Doctor)
 - **Screen 2: Choose Appointment Date and Time**
 - Field Types: Date, Time, Picklist
 - **Create Records:** Appointment
 - **Screen 3: Confirmation**

Record-Triggered Flow Requirements

1. Send Appointment Reminder

Objective: Automatically send an email reminder to patients 24 hours before their appointment.

Trigger: Appointment record creation or update

Criteria: Appointment Date = TODAY() + 1

Actions:

1. **Action: Send Email**
 - Use a predefined email template to send a reminder to the patient's email address.

Configuration:

- **Flow Name:** Send Appointment Reminder
- **Type:** Record-Triggered Flow
- **Trigger:** When a record is created or updated
- **Criteria:** [Appointment].Appointment_Date__c = TODAY() + 1
- **Actions:**
 - **Action:** Send Email (Appointment Reminder Email Template)

2. Update Billing Status

Objective: Automatically update the billing status to "Overdue" if the payment is not received within 30 days.

Trigger: Billing record creation or update

Criteria: Billing Date <= TODAY() - 30 AND Status != 'Paid'

Actions:

1. **Action: Update Field**
 - Update the Billing Status field to "Overdue".

Configuration:

- **Flow Name:** Update Billing Status
- **Type:** Record-Triggered Flow
- **Trigger:** When a record is created or updated
- **Criteria:** [Billing].Billing_Date__c <= TODAY() - 30 AND NOT(ISPICKVAL([Billing].Status__c, 'Paid'))
- **Actions:**
 - **Action:** Update Field (Status to 'Overdue')

3. Create Follow-Up Appointment

Objective: Automatically create a follow-up appointment 7 days after a patient's appointment is marked as completed.

Trigger: Appointment record update

Criteria: Status = 'Completed'

Actions:

1. **Action: Create Record**
 - Create a new Appointment record with Date = Appointment Date + 7 days

Configuration:

- **Flow Name:** Create Follow-Up Appointment
- **Type:** Record-Triggered Flow
- **Trigger:** When a record is updated
- **Criteria:** [Appointment].Status__c == 'Completed'
- **Actions:**
 - **Action:** Create Record (New Appointment)
 - **Field Values:**
 - **Patient:** [Appointment].Patient__c
 - **Doctor:** [Appointment].Doctor__c
 - **Appointment Date:** [Appointment].Appointment_Date__c + 7
 - **Status:** 'Scheduled'

Reports and Dashboards Requirements

1. Patient Visits Report

- **Objective:** Track the number of patient visits over a specific period, segmented by various criteria (e.g., doctor, department).
- **Filters:** Date Range (e.g., current month, last quarter), Department, Doctor.
- **Groupings:** Visit Date, Doctor, Department.
- **Summary:** Count of visits, average visits per day, percentage breakdown by visit type.

2. Treatment Outcomes Report

- **Objective:** Monitor the outcomes of treatments provided to patients, including success rates and follow-up requirements.
- **Fields:** Patient Name, Treatment Provided, Outcome (Success/Failure), Follow-Up Needed (Yes/No), Date of Treatment.
- **Filters:** Date Range, Treatment Type, Outcome.
- **Groupings:** Patient Name, Treatment Type, Outcome.
- **Summary:** Success rate percentage, count of follow-up required.

3. Financial Performance Report

- **Objective:** Analyze revenue and expenses to assess the financial health of the hospital.
- **Fields:** Invoice Date, Patient Name, Billing Amount, Payment Status, Expense Category (if applicable).
- **Filters:** Date Range, Payment Status (Paid, Unpaid), Expense Category.
- **Groupings:** Invoice Date, Patient Name, Payment Status.
- **Summary:** Total revenue, average billing amount, expense breakdown if applicable.

Dashboards

1. Real-Time Patient Flow Dashboard

- **Components:**
 - **Metric:** Total number of patients currently admitted.
 - **Chart:** Daily patient admissions and discharges.
 - **Table:** List of current patients with their admission dates and status.

2. Bed Occupancy Dashboard

- **Components:**
 - **Metric:** Percentage of occupied beds versus total beds.
 - **Chart:** Daily bed occupancy trends.
 - **Table:** Breakdown of bed occupancy by ward or department.

3. Revenue Dashboard

- **Components:**
 - **Metric:** Total revenue generated year-to-date.
 - **Chart:** Revenue breakdown by source (e.g., insurance, out-of-pocket).
 - **Table:** Top paying patients or insurance providers.

Einstein Chatbots Implementation for Patient Support and Appointment Scheduling

□ Patient Support Chatbot

- **Objective:** Provide instant responses to common patient inquiries and support needs.
- **Features:**
 - **General Information:** Answer queries about hospital services, visiting hours, contact details.

- **Appointment Assistance:** Help patients schedule, reschedule, or cancel appointments.
- **Directions:** Provide directions and guidance within the hospital premises.
- **Billing and Insurance:** Assist with billing inquiries and insurance-related questions.

□ Appointment Scheduling Chatbot

- **Objective:** Assist patients in booking appointments with healthcare providers seamlessly.
- **Features:**
 - **Real-time Availability:** Check and confirm appointment availability based on patient preferences.
 - **Confirmation and Reminders:** Send appointment confirmations and reminders via email or SMS.
 - **Rescheduling and Cancellation:** Handle requests to reschedule or cancel appointments as per patient instructions.

Lightning Web Component Requirements

1. Appointment Booking Component

- **Objective:** Enable patients to book appointments with healthcare providers through an intuitive and responsive interface.
- **Features:**
 - **Calendar Integration:** Interactive calendar for selecting appointment dates and times.
 - **Real-Time Availability:** Check provider availability in real-time based on their schedule.
 - **Confirmation:** Display appointment details and send confirmation notifications via email or SMS.
- **Integration:** Connect with Salesforce CRM to create/update Appointment records and send notifications using Apex or external APIs.
- **Deployment:** Embed within Community Cloud, patient portals, or custom Lightning App pages for seamless user interaction.

2. Patient Medical Records Viewer

- **Objective:** Provide healthcare providers with a secure and comprehensive view of patient medical records for efficient patient care.
- **Features:**
 - **Record Display:** Show patient demographics, medical history, lab results, and treatment plans.
 - **Editable Fields:** Allow authorized users to update records securely within the component.
 - **Document Integration:** Display scanned documents or images related to patient records.
- **Integration:** Retrieve data from Salesforce objects (e.g., Patient records, Medical history) and ensure compliance with data privacy regulations (e.g., HIPAA).

- **Deployment:** Integrate as a Lightning App Page or embed within Salesforce Console for easy access during patient consultations.

3. Real-Time Patient Dashboard

- **Objective:** Provide hospital staff with real-time insights into patient demographics, bed occupancy, and emergency room statuses.
- **Features:**
 - **Data Visualization:** Display charts and graphs for patient demographics by age, gender, and medical condition.
 - **Bed Management:** Visual representation of bed availability and occupancy by department or ward.
 - **Emergency Room Status:** Highlight critical patient statuses and waiting times.
- **Integration:** Retrieve data from Salesforce objects and external systems via APIs for up-to-date information.
- **Deployment:** Deploy within Salesforce Lightning Experience or Community Cloud for easy access by hospital staff and administrators.

4. Healthcare Provider Directory

- **Objective:** Create a searchable directory of healthcare providers (doctors, specialists) for patient referrals and appointments.
- **Features:**
 - **Search Functionality:** Enable patients to search providers by specialty, location, and availability.
 - **Provider Profiles:** Display detailed profiles including contact information, bio, and patient reviews.
 - **Appointment Scheduling Integration:** Allow direct booking of appointments from provider profiles.
- **Integration:** Retrieve provider data from Salesforce objects or external directories via APIs for comprehensive provider information.
- **Deployment:** Integrate as a Lightning App Page or embed within Community Cloud for seamless patient engagement and provider management.

Visualforce (VF) Pages Requirements

1. Fees Receipt Generation

- **Objective:** Generate printable receipts for patient fees and payments within Salesforce.
- **Features:**
 - **Receipt Details:** Display patient name, payment date, amount paid, and payment method.
 - **Billing Information:** Include invoice numbers or reference details for tracking purposes.
 - **PDF Generation:** Provide an option to generate a downloadable PDF version of the receipt.
- **Integration:** Retrieve payment data from Salesforce objects (e.g., Payments, Invoices) using Apex controllers.

- **Deployment:** Embed the VF page within Salesforce Console or Community Cloud for access by billing administrators and patients.

2. Payment Billing Interface

- **Objective:** Provide a user-friendly interface for billing administrators to manage patient payments and invoices.
- **Features:**
 - **Payment Entry:** Allow entry of payment details including patient name, amount, date, and payment method.
 - **Invoice Management:** Display outstanding invoices, allow marking of invoices as paid, and generate payment receipts.
 - **Reporting:** Include summary reports of payments received and outstanding balances.
- **Integration:** Use Salesforce objects (e.g., Payments, Invoices) and Apex controllers to handle payment processing and invoice management.
- **Deployment:** Integrate as a VF page within Salesforce Console for billing staff to efficiently manage patient billing and payments.

3. Appointment PDF Generation

- **Objective:** Generate printable PDF documents for patient appointments with detailed information.
- **Features:**
 - **Appointment Details:** Include patient name, appointment date, time, provider details, and appointment type.
 - **Medical Instructions:** Display any specific instructions or preparations required for the appointment.
 - **PDF Formatting:** Format the PDF document with a professional layout suitable for patient records.
- **Integration:** Retrieve appointment data from Salesforce objects (e.g., Appointments, Patients) using Apex controllers.
- **Deployment:** Embed the VF page within Salesforce Console or Community Cloud for access by healthcare providers and patients.

Aura/LWC

- Create a Lightning component to display the patient details on search of name.
- Create a Lightning Web Component to fetch the list of doctors on the click of button.