# Project Documentation

**Hospital Appointment & Health Tracker System on Salesforce**

## Phase 3: Data Modeling & Relationships

### 1. Introduction

In this phase, we design the **data model** of the Salesforce application to represent the hospital appointment and health tracking process.  
A well-structured data model ensures **scalability, patient confidentiality, and efficient reporting**.

This phase involves **Standard Objects, Custom Objects, Fields, Record Types, Page Layouts, Compact Layouts, Schema Builder, Relationships, and Junction Objects**.

### 2. Standard & Custom Objects

**Standard Objects:**

**Account** → Represents hospital departments or external partner hospitals.

**Contact** → Represents patients or doctors.

**Case** → Can track patient service requests or complaints.

**Custom Objects (specific to project):**

**Patient\_\_c** → Stores patient details (Name, Age, Gender, Contact Info, Medical History).

**Doctor\_\_c** → Represents doctors (Name, Specialization, Availability, Contact Info).

**Appointment\_\_c** → Tracks patient-doctor appointments (Date, Time, Status, Reason).

**HealthRecord\_\_c** → Stores patient medical history (Symptoms, Diagnosis, Prescription, Reports).

**Department\_\_c** → Represents hospital departments (Name, Services, Doctors linked).

**PatientDoctor\_\_c (Junction Object)** → Links Patients and Doctors for many-to-many relationships.

### 3. Fields

Each object has standard fields plus additional custom fields.

**Patient\_\_c Fields:**

Patient Name (Text)

Age (Number)

Gender (Picklist → Male, Female, Other)

Phone (Phone)

Email (Email)

Medical History (Long Text Area)

**Doctor\_\_c Fields:**

Doctor Name (Text)

Specialization (Picklist → Cardiologist, Orthopedic, Pediatrician, etc.)

Contact Info (Phone, Email)

Availability (Picklist → Available, On Leave, Busy)

Department (Lookup → Department\_\_c)

**Appointment\_\_c Fields:**

Appointment Date (Date/Time)

Status (Picklist → Scheduled, Completed, Cancelled, Rescheduled)

Reason for Visit (Text Area)

Patient (Lookup → Patient\_\_c)

Doctor (Lookup → Doctor\_\_c)

**HealthRecord\_\_c Fields:**

Symptoms (Long Text Area)

Diagnosis (Long Text Area)

Prescription (Long Text Area)

Report Upload (File)

Related Patient (Lookup → Patient\_\_c)

Related Appointment (Lookup → Appointment\_\_c)

**Department\_\_c Fields:**

Department Name (Text)

Services Offered (Long Text Area)

**PatientDoctor\_\_c (Junction Object):**

Patient (Master-Detail → Patient\_\_c)

Doctor (Master-Detail → Doctor\_\_c)

Status (Picklist → Active, Inactive)

### 4. Record Types

**Appointment\_\_c Record Types:**

OPD Appointment

Emergency Appointment

Surgery Appointment

**HealthRecord\_\_c Record Types:**

General Consultation

Lab Report

Prescription

### 5. Page Layouts

**Patient\_\_c Layouts:**

Basic Info Layout → Shows patient demographics.

Medical History Layout → Includes appointments and health records.

**Doctor\_\_c Layouts:**

Doctor Profile Layout → Shows specialization and contact details.

Availability Layout → Includes schedules and appointments.

**Appointment\_\_c Layouts:**

OPD Layout → Basic check-up details.

Surgery Layout → Pre-op & post-op details.

### 6. Compact Layouts

**Patient\_\_c Compact Layout:** Patient Name, Age, Phone, Gender.  
**Doctor\_\_c Compact Layout:** Doctor Name, Specialization, Availability.  
**Appointment\_\_c Compact Layout:** Appointment Date, Status, Doctor, Patient.  
**HealthRecord\_\_c Compact Layout:** Symptoms, Diagnosis, Prescription.

### 7. Schema Builder

Schema Builder will be used to:

Visualize relationships between Patient, Doctor, Appointment, HealthRecord, and Department.

Arrange objects to represent **ERD (Entity Relationship Diagram)**.

Validate **junction objects** and field dependencies.

### 8. Relationships

**Patient\_\_c → Appointment\_\_c** → Lookup (one patient can have many appointments).

**Doctor\_\_c → Appointment\_\_c** → Lookup (one doctor can have many appointments).

**Patient\_\_c → HealthRecord\_\_c** → Master-Detail (a patient can have many health records).

**Appointment\_\_c → HealthRecord\_\_c** → Lookup (each record tied to appointment).

**Department\_\_c → Doctor\_\_c** → Lookup (department has many doctors).

**Patient\_\_c ↔ Doctor\_\_c** → Many-to-Many (managed by junction object PatientDoctor\_\_c).

### 9. Junction Objects

**PatientDoctor\_\_c** → Junction between Patient and Doctor.

Purpose: Track ongoing doctor-patient associations beyond single appointments.

Helps in long-term treatment tracking and follow-ups.

### 10. External Objects

**External\_LabReports\_\_x** → Connects Salesforce to external lab systems for diagnostics.

**External\_WearableData\_\_x** → Connects Salesforce with health trackers (Fitbit, Apple Health) for real-time monitoring.

### 11. Documentation Deliverables

**ERD Diagram** (Patient, Doctor, Appointment, HealthRecord, Department, PatientDoctor).

**Custom Object & Field Tables** (Name, API Name, Type, Description).

**Record Type & Layout Mapping**.

**Junction Object Mapping**.

**Screenshots**: Schema Builder, Page Layouts, Compact Layouts.

### 12. Benefits of This Phase

Provides a **clear and scalable data structure** for hospital processes.

Supports **appointment scheduling, tracking, and automation**.

Enables **doctor-patient history visibility**.

Prepares the model for **automation, reporting, and integrations** in later phases.

**Phase 3 Deliverable:**

Defined objects, fields, and relationships.

Mapped record types, layouts, and compact layouts.

Designed ERD using Schema Builder.

Established junction objects for many-to-many relationships.

This forms the foundation for **Phase 4: Process Automation (Admin)**.