Phase 3 Documentation – Data Modeling & Relationships

Goal of Phase 3

To design and implement the data model for the Hospital Management System using Salesforce objects, fields, record types, page layouts, compact layouts, and relationships. This ensures that patient, visit, and treatment plan data is properly structured, accessible, and secure.

1.Standard & Custom Objects :

What was done:

- Reviewed standard objects like User for doctor/attending staff details.
- Created Custom Objects specific to the project:
    - Patient → Stores patient personal and medical details.
    - Visit → Stores patient visit details including symptoms, diagnosis, and attending doctor.
    - Treatment_Plan → Stores medical and surgical treatment plans for patients.

Output: Custom objects for Patient, Visit, and Treatment_Plan created.

2. Fields :

What was done:

- Added necessary custom fields to capture all required information.

Patient Fields:

- Patient Name, Patient ID, First Name, Last Name, Date of Birth, Gender, Age, Address, Phone, Email, Emergency Contact Name & Phone, Medical History, Primary Doctor.

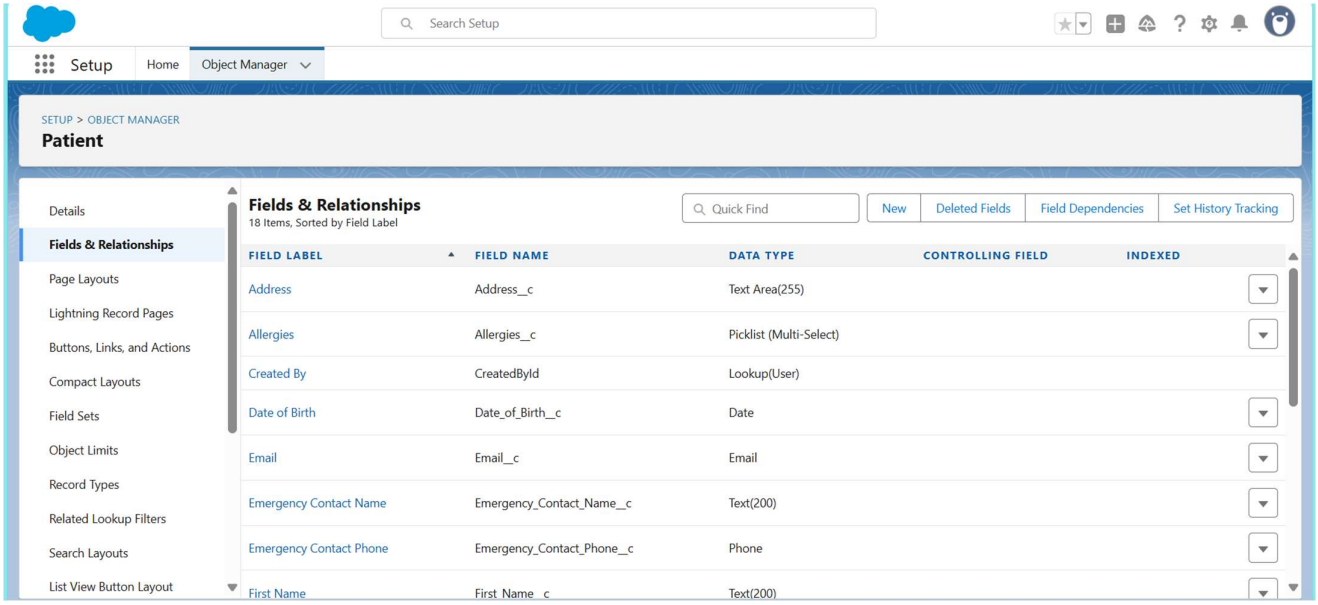
Visit Fields:

- Visit Name, Patient Lookup, Visit Date and Time, Reason for Visit, Symptoms, Diagnosis, Visit Notes, Attending Doctor.

Treatment_Plan Fields:

- Treatment Plan Name, Treatment Plan ID, Patient Lookup, Assigned Doctor, Start Date, End Date, Status, Description.

Output: Custom fields created per object.

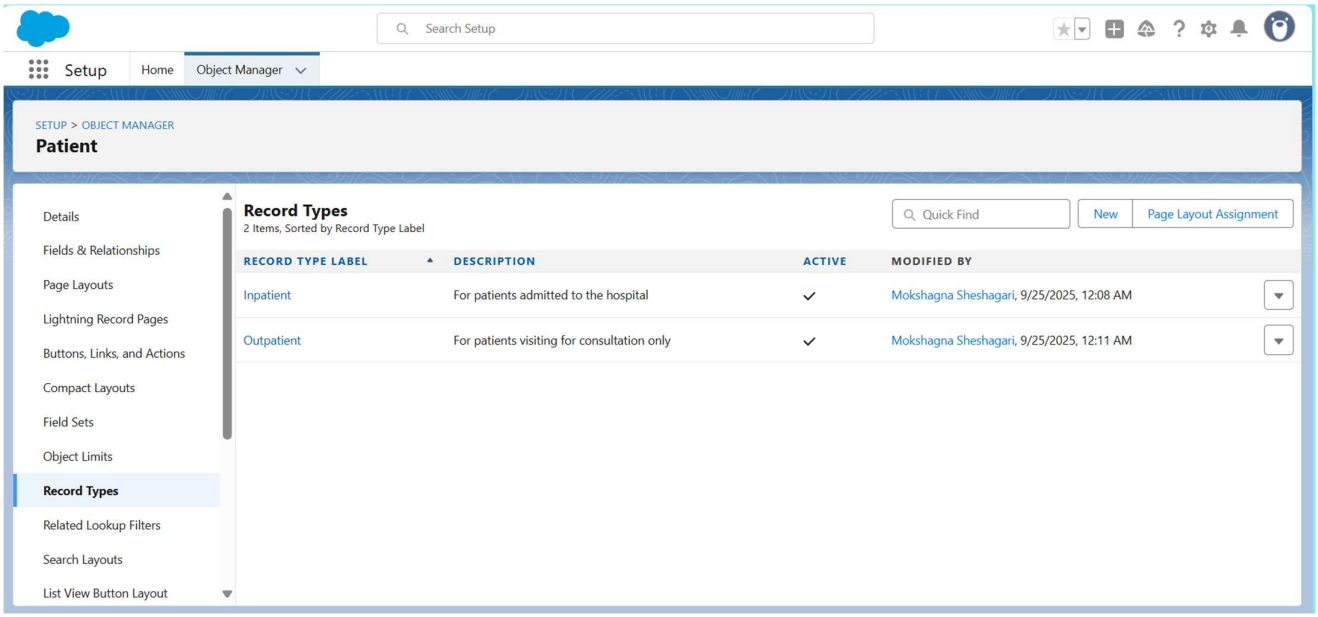


3. Record Types :

What was done:

- Applied Record Types to differentiate processes where required:
    - Patient: Inpatient & Outpatient record types to differentiate admitted patients vs. appointment visits.
    - Visit: Initial Visit & Follow-Up Visit record types to capture first-time visits vs. subsequent visits.
    - Treatment_Plan: Medical Treatment & Surgical Treatment record types to separate medical vs. surgical plans.

Output: Record Types designed to separate categories for better workflow management.



4. Page Layouts :

What was done:

- Created different page layouts to display relevant fields and control visibility:

Patient Layouts:

- Inpatient Layout: Admission Date, Room Number, Primary Doctor.
- Outpatient Layout: Appointment Date, Visit Reason.

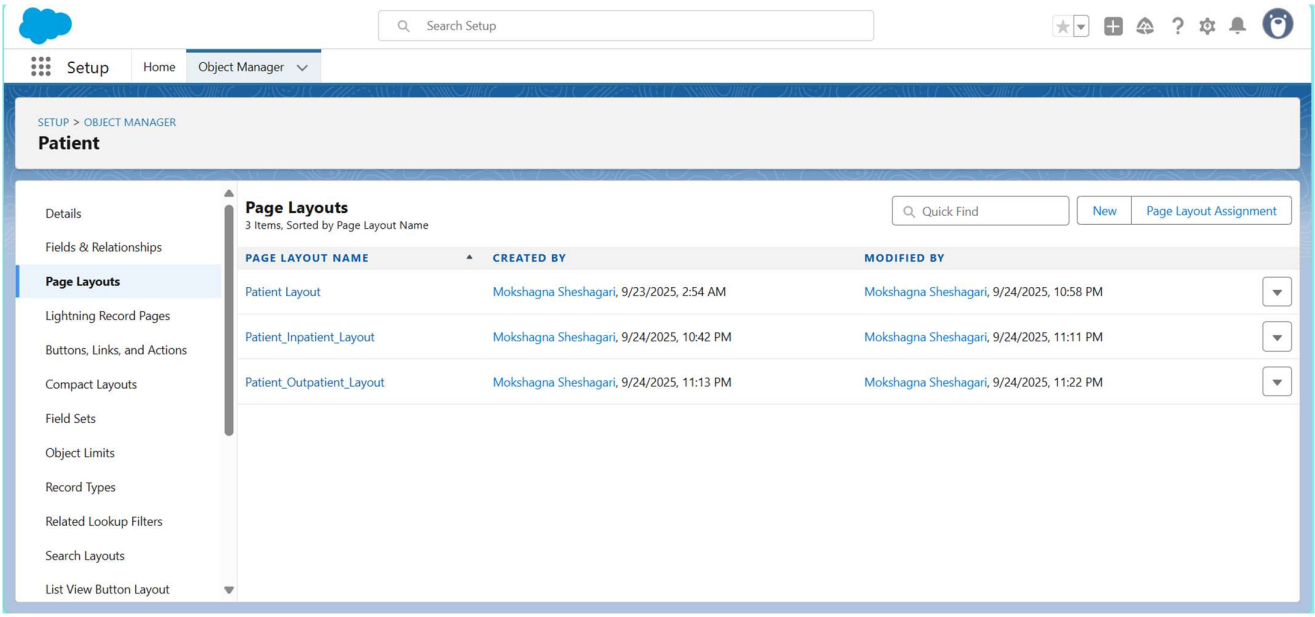
Visit Layouts:

- Initial Visit Layout: Capture full patient history including symptoms and diagnosis.
- Follow-Up Visit Layout: Focus on progress notes and test results.

Treatment Plan Layouts:

- Medical Treatment Layout: Medications, Therapy, Diet Plan.
- Surgical Treatment Layout: Procedure Details, Pre/Post-Op Notes.

Output: Separate layouts for each record type with controlled visibility and logical arrangement of fields.



5. Compact Layouts :

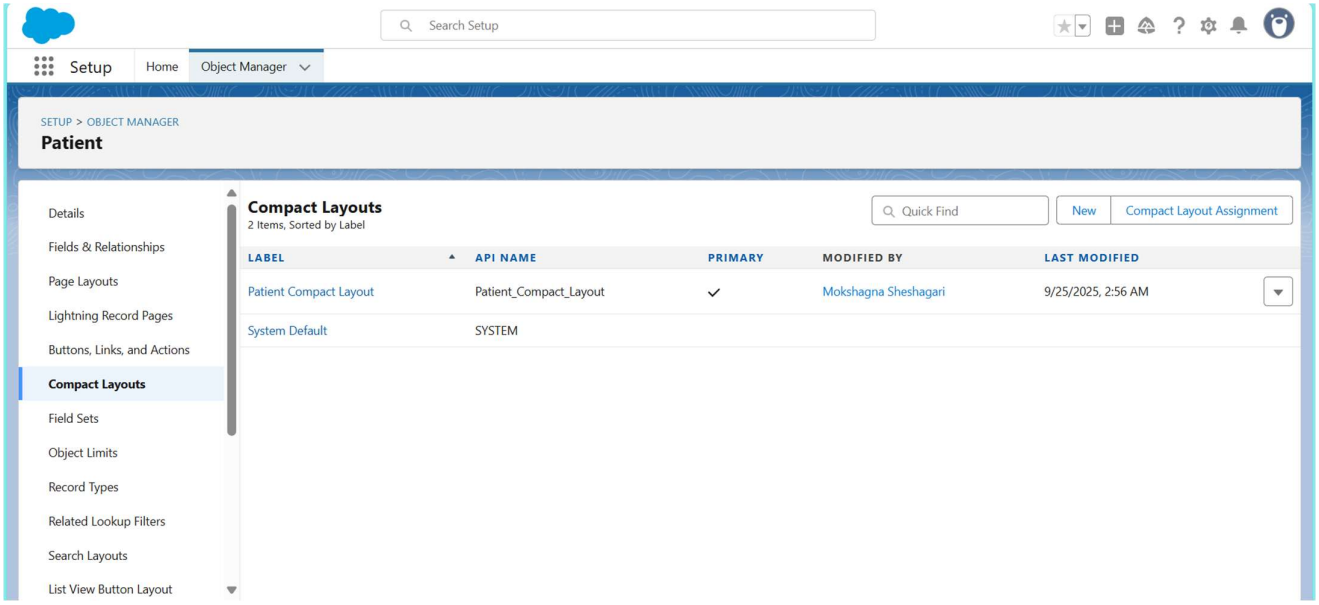
What was done:

- Created Compact Layouts for quick view in Highlights Panel and mobile interface.

Examples:

- Patient_Compact_Layout: Patient Name, Patient ID, Date of Birth, Gender, Age.
- Visit_Compact_Layout: Patient, Visit Date and Time, Reason for Visit, Attending Doctor, Diagnosis.
- Treatment_Plan_Compact_Layout: Treatment Plan Name, Patient, Assigned Doctor, Status, Start Date.

Output: Compact layouts applied to display key fields for faster access.

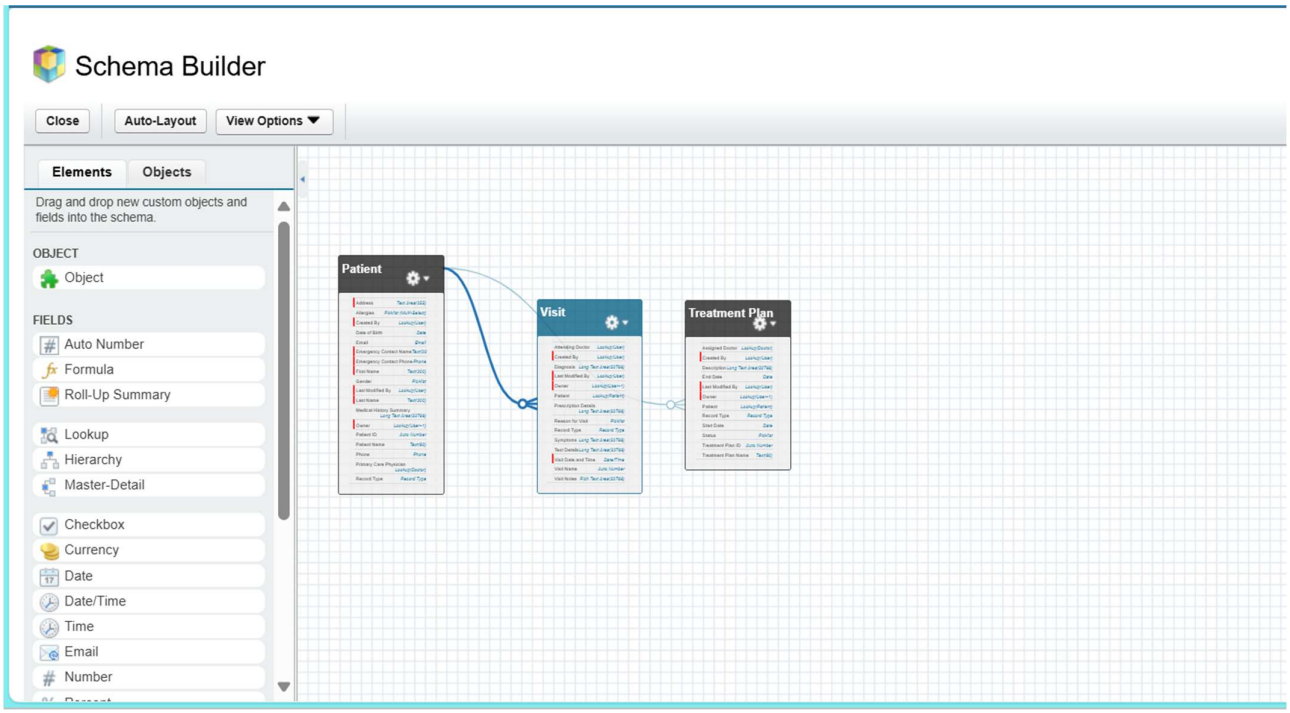


6. Schema Builder

What was done:

- Used Schema Builder to visualize objects and relationships.
- Placed Patient on the left, Visit and Treatment_Plan on the right for clarity.
- Verified relationships between objects were correctly mapped:
    - Patient ↔ Visit
    - Patient ↔ Treatment_Plan
- Ensured all fields were correctly displayed.

Output: Graphical data model available for reference and validation.



7. Relationships

What was done:

- Applied correct relationship types based on project requirements:
    1. **Lookup Relationship: Visit → Patient**
        - One patient can have many visits.
        - Visits can exist without deleting the patient.
    2. **Lookup Relationship: Treatment_Plan → Patient**
        - One patient can have multiple treatment plans.
        - Treatment plans can remain even if visits are removed.
    3. **Lookup Relationship: Visit → Attending Doctor (User)**
        - Each visit is associated with a doctor.
        - Loose link to the User object.

Output:

- Clear relationship structure defined between all objects for accurate data linkage.

Outcome of Phase 3

- Custom objects **Patient, Visit, Treatment_Plan** created with all necessary fields.
- **Record Types, Page Layouts, and Compact Layouts** set up for easy data access.
- **Lookup relationships** established between Patient, Visit, and Treatment_Plan.
- **Schema Builder** used to visualize and confirm the data model.

Result: A structured, user-friendly system for managing patients, visits, and treatment plans.