Smart College CRM – Student Management System on Salesforce

Phase 3: Data Modeling & Relationships

**Goal:** To design and implement a clear and organized data model in Salesforce that accurately represents students, courses, applications, and fees, enabling efficient data management, reporting, and future automation

**1. Introduction**

In this phase, we design the data model of the Salesforce application to represent the college’s student management process. A well-structured data model ensures **scalability, data integrity, and efficient reporting**.

This phase involves working with **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 colleges, departments, or external partners.
* **Contact** → Represents students, faculty, or staff.
* **Opportunity** → Can be used for scholarship or fee-related transactions.
* **Case** → Can track student support or queries.

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

* **Student\_\_c** → Stores student details (Name, Enrollment No, Admission Date, Course, Fee Status).
* **Course\_\_c** → Represents courses offered by the college (Course Name, Duration, Credits).
* **Application\_\_c** → Tracks student applications for admission (Status, Submission Date, Assigned Officer).
* **Fee\_\_c** → Tracks fee payments (Amount, Due Date, Status, Payment Method).
* **StudentCourse\_\_c (Junction Object)** → Links Students and Courses (many-to-many relationship).

**3. Fields**

**Student\_\_c Fields:**

* Name (Text)
* Enrollment No (Auto Number)
* Admission Date (Date)
* Course (Lookup → Course\_\_c via Junction Object)
* Fee Status (Picklist → Paid, Pending, Overdue)

**Course\_\_c Fields:**

* Course Name (Text)
* Course Code (Text)
* Duration (Number – in months/years)
* Credits (Number)

**Application\_\_c Fields:**

* Student Name (Lookup → Student\_\_c)
* Application Date (Date)
* Status (Picklist → Submitted, Approved, Rejected, Waitlisted)
* Assigned Officer (Lookup → User)

**Fee\_\_c Fields:**

* Student (Master-Detail → Student\_\_c)
* Amount (Currency)
* Due Date (Date)
* Payment Status (Picklist → Paid, Pending, Overdue)
* Payment Method (Picklist → Cash, Online, Cheque)

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

* Student (Master-Detail → Student\_\_c)
* Course (Master-Detail → Course\_\_c)
* Enrollment Status (Picklist → Enrolled, Completed, Dropped)

**4. Record Types**

**Student\_\_c Record Types:**

* Regular Student
* Scholarship Student

**Application\_\_c Record Types:**

* New Admission
* Transfer Admission

**Fee\_\_c Record Types:**

* Tuition Fee
* Exam Fee
* Hostel Fee

**Course\_\_c Record Types:**

* Undergraduate
* Postgraduate

**5. Page Layouts**

**Student\_\_c Layouts:**

* Regular Layout → Shows basic info, course enrollment, fee status
* Scholarship Layout → Includes scholarship type and amount

**Application\_\_c Layouts:**

* New Admission Layout → Shows application details, assigned officer
* Transfer Layout → Shows previous college info

**Fee\_\_c Layouts:**

* Tuition Fee Layout → Shows fee breakdown, due date, payment history
* Hostel Fee Layout → Shows hostel fee details

**Course\_\_c Layouts:**

* Undergraduate Layout → Course duration, credits, syllabus
* Postgraduate Layout → Course duration, credits, electives

**6. Compact Layouts**

**Student\_\_c Compact Layout:**

* Name, Enrollment No, Admission Date, Fee Status

**Application\_\_c Compact Layout:**

* Student Name, Application Date, Status

**Fee\_\_c Compact Layout:**

* Amount, Payment Status, Due Date

**Course\_\_c Compact Layout:**

* Course Name, Duration, Credits

**7. Schema Builder**

Schema Builder will be used to:

* Visualize relationships between **Student, Course, Application, Fee, and StudentCourse**.
* Represent ERD (Entity Relationship Diagram).
* Validate fields and relationships created.

**8. Relationships**

* **Student\_\_c → Application\_\_c:** Lookup (one student can have multiple applications)
* **Student\_\_c → Fee\_\_c:** Master-Detail (one student can have multiple fees)
* **Course\_\_c → StudentCourse\_\_c:** Master-Detail
* **Student\_\_c → StudentCourse\_\_c:** Master-Detail
* **Student\_\_c → Fee\_\_c:** Lookup for payment tracking

**9. Junction Objects**

* **StudentCourse\_\_c** is the junction object between **Student** and **Course**.
* Purpose: Manage many-to-many relationships (a student can enroll in multiple courses, a course can have multiple students).

**10. External Objects (Optional)**

* **External\_Payment\_\_x** → Connects Salesforce to payment gateway (Stripe/PayPal) for tuition payments.
* **External\_Survey\_\_x** → Connects Salesforce to Google Forms/SurveyMonkey for course feedback.

**11. Documentation Deliverables**

* ERD Diagram (Student, Course, Application, Fee, StudentCourse)
* 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, scalable data structure for students, courses, applications, and fees.
* Enables automation for enrollment, fee reminders, and application approvals.
* Supports advanced reporting and dashboards (e.g., total fees collected, pending applications).
* Prepares model for integration with payment gateways and survey tools.

**Phase 3 Deliverable**

By the end of Phase 3, we have:

* Defined **custom objects, fields, and relationships** for Student Management.
* 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 (Workflows, Approvals, and Flows)**.