FINAL REPORT

Project Title

“CRM APPLICATION FOR JEWEL MANAGEMENT – (DEVELOPER)”

College Name

Anil Neerukonda Institute of Technology and Sciences (ANITS)

Team ID : **LTVIP2025TMID29987**

Team Size : 4

Team Leader : Korrayi Sravya

Email ID : Korrayisravya.22.It@anits.edu.in

Team member : Bonikala Dhamodhar

Email ID : Dhamodharbonikala@gmail.com

Team member : Appalabattula Yaswanthsai

Email ID : Ayaswanthsai15@gmail.com

Team member : Dinesh Ketana

Email ID : ketanadinesh.22.it@anits.edu.in

1. INTRODUCTION

1.1 Project Overview

The **CRM Application for Jewel Management** is a fully customized Salesforce-based solution developed to transform and streamline the business processes of a jewelry store. Jewelry businesses, especially in traditional setups, often struggle with scattered records, manual billing systems, inconsistent inventory tracking, and lack of personalized customer engagement. This project aims to digitize and centralize all such business operations using Salesforce's robust CRM capabilities.

This CRM system has been specifically designed to manage and automate the four critical components of the jewelry business:

1. **Customer Management** – Maintaining comprehensive records of each customer, including their contact details, preferences, gender, and order history.

2. **Inventory/Item Management** – Keeping track of the jewelry items available (categorized as Gold and Silver), their attributes (weight, ornament type, pricing components), and current stock.

3. **Order Management** – Allowing the staff to place and monitor customer orders with details such as quantity, item type, and linked customer.

4. **Billing and Payments** – Automating the invoice generation process by calculating total amounts using formula fields and sending confirmation emails to customers using flows.

The project leverages Salesforce's **custom objects**, **page layouts**, **record types**, **profiles**, **roles**, **Apex triggers**, and **record-triggered flows** to ensure seamless operation and maintain high data accuracy and integrity. The use of dashboards and custom reports enhances visibility for business owners and stakeholders, enabling them to make data-driven decisions.

The Lightning App interface allows all users (admin, goldsmith, workers) to navigate between different functionalities with ease, ensuring user-friendliness and efficiency in day-to-day operations.

1.2 Purpose

The primary purpose of this project is to provide a **scalable, automated, and user-friendly CRM solution** for jewelry businesses using the Salesforce platform. By developing this system, the team aims to solve real-world business problems related to manual tracking and customer mismanagement that many local and regional jewelers face. The solution focuses on the following key goals

Customer Data Handling

- Create a structured repository for customer records.

- Provide lookup-based linking between customer records and orders/billings.

- Enable role-based access so that only authorized users can view/edit sensitive customer data.

Order Tracking

- Keep a detailed log of customer orders, including item selection, quantity, and order date.
- Allow users to view all orders linked to a specific customer or item.
- Make it easier to retrieve historical orders for references, refunds, or exchanges.

Automated Billing

- Automatically compute the total billing amount based on item details like price, KDM charges, stone charges, and making charges.
- Minimize manual calculation errors using **formula fields** in the Billing object.
- Ensure billing records are always synchronized with payment records using Apex Triggers.
- Allow users to view paid amount, outstanding amount, and update payments with ease.

Pricing History Management

- Maintain historical pricing of gold and silver using the **Price__c** object.
- Provide a time-based reference to pricing, allowing for price trend analysis.
- Support dynamic referencing of valid prices during billing and inventory updates.

Role-Based Access Control

- Define different **profiles** and **roles** such as Admin, Goldsmith, and Worker.
- Ensure that users can access only the data and functionalities relevant to their job role.
- Improve security and accountability within the organization.

Email Notifications & Communication

- Send automated billing confirmation emails to customers using Record-Triggered Flows.
- Provide customers with a digital proof of their transactions.
- Enhance professionalism and customer experience.

Reporting and Business Intelligence

- Build custom **reports** and **dashboards** to visualize KPIs such as:
    - Customer count by gender
    - Total revenue by item
    - Item type distribution
    - Orders over time

- Empower business owners to analyze trends, performance, and make informed decisions.

Data Accuracy and Validation

- Implement **Validation Rules** to prevent data entry errors, such as entering a paid amount greater than the bill amount.
- Enforce business logic directly at the database level, reducing the chances of fraud or mistakes.

2. IDEATION PHASE

2.1 Problem Statement

Problem statements are crucial for any project because they clearly articulate the core issues your users face. They act as a compass, guiding your team to focus on solving the most significant pain points and ensuring the developed solution truly adds value. By framing problems in this way, you gain a deep understanding of *who* is affected, *what* the specific struggle is, *why* it's happening, and the *negative consequences* it leads to. This clarity is essential for aligning the team and setting precise goals for your CRM for Jewel Management.

In the traditional jewelry retail business, operations are often managed through **manual methods**, such as handwritten registers, physical invoices, verbal order tracking, and informal pricing systems. While this approach may work for very small businesses, it becomes increasingly inefficient, error-prone, and opaque as the business scales or when multiple staff are involved.

Through our field interviews and observations, we identified several key **pain points** that affect both customers and business owners in such setups:

Manual Record-Keeping

- Customer data is often written in notebooks or Excel sheets with no centralized repository.
- This leads to data redundancy, missing records, and a lack of historical tracking.

Inefficient Order Tracking

- Orders are recorded casually, with no formal link between the customer and the ordered item.
- It's difficult to trace back the order history, especially for repeated customers.

Billing Errors and Delays

- Invoices are generated manually using calculators or spreadsheets.
- This often leads to miscalculations of taxes, stone prices, KDM charges, and making charges.
- Customers may be charged incorrectly, damaging trust.

No Customer Communication

- No automated system exists to confirm billing, order status, or payment received.

- Customers often need to call the store or visit physically for updates.

No Role-Based Access

- Every staff member has access to all data, including financial records and customer details.
- There's no way to restrict access based on job function (e.g., worker vs. manager).

Lack of Business Intelligence

- Owners have no real-time dashboard or report to understand sales performance, inventory levels, or customer trends.
- Business decisions are made based on intuition, not data.

Problem Statement Template:

We believe that **[Customer Type]** is struggling with **[Core Problem]** because of **[Root Cause 1]** and **[Root Cause 2, etc.]**. This causes **[Negative Impact 1]** and **[Negative Impact 2, etc.]**.

Problem Statement 1: Operational Inefficiency and Financial Risk for Store Owners and Staff

We believe that **jewelry store staff and owners** are struggling with **managing inventory, orders, billing, and customer communication efficiently** because of **the lack of a centralized digital system** and **over-reliance on fragmented, manual processes (like notebooks and spreadsheets)**. This causes **lost sales opportunities due to inability to quickly access customer history or stock, frequent inaccuracies in financial records leading to potential revenue loss, and decreased customer satisfaction due to slow service and errors.**

- **Elaboration:** This problem statement directly addresses the internal struggles of running a jewelry business without modern tools.
    - **Customer Type:** Encompasses everyone from the front-line sales associate struggling to find a customer's last purchase to the owner trying to reconcile daily sales.
    - **Core Problem:** "Managing... efficiently" points to a systemic issue where core business functions are cumbersome and time-consuming.
    - **Root Causes:** "Lack of a centralized digital system" means information is scattered and not accessible in real-time across the business. "Reliance on manual processes" implies a heavy dependence on pen-and-paper or basic, unintegrated digital tools that are prone to human error and difficult to scale.
- **Negative Impacts:**
    - **Lost sales opportunities:** If a sales associate can't quickly pull up a customer's purchase history or current stock availability (e.g., if a specific ring size is in stock or needs to be ordered), a potential sale might be missed or delayed.
    - **Inaccuracies in financial records:** Manual billing and payment tracking can lead to miscalculations, unrecorded payments, or incorrect tax applications, directly impacting profitability and compliance.
    - **Decreased customer satisfaction:** Slow service, billing errors, or the inability to quickly resolve issues due to fragmented information frustrates customers and erodes trust.

2.2 Empathy Map Canvas

The Empathy Map Canvas serves as a powerful tool to step into the shoes of your target users, understanding their world beyond just explicit requirements. By exploring what users say, think, do, and feel, we gain deeper insights into their needs, pain points, and motivations, which are crucial for designing a truly effective solution.

Who are we empathizing with?

Our primary target users for this CRM solution are:

- **Jewelry Shop Owners:** These individuals bear the ultimate responsibility for the business's profitability, customer satisfaction, and operational efficiency. Their concerns often revolve around overall business growth, financial tracking, inventory valuation, staff management, and ensuring a seamless customer experience. They need comprehensive oversight and tools that empower strategic decision-making.
- **In-store Workers (Sales Associates, Front Desk Staff):** These are the direct points of contact with customers. Their daily tasks involve sales, customer service, handling inquiries, processing transactions, and managing store operations. Their pain points typically relate to time-consuming manual processes, difficulty accessing information quickly, and challenges in providing consistent, high-quality customer service due to system limitations.

To better understand the needs of each stakeholder involved, we created an **Empathy Map Canvas** focusing on **three main personas**: the **Goldsmith (Owner)**, **Worker (Staff)**, and **Customer**.

Persona 1: Goldsmith (Business Owner)

CATEGORY	DETAILS
Says	"I want to know how much I sold this month." "Customers keep asking for bills, I can’t always find them."
Thinks	"If only I had a system to track everything automatically." "I need to control what workers can see or edit."
Does	Manages billing manually, reviews sales records, interacts with customers occasionally.
Feels	Overwhelmed by administrative tasks. Worried about errors and fraud. Desires a better system to manage data securely.

Persona 2: Worker (Store Staff / Sales Associate)

CATEGORY	DETAILS
Says	"I only need to create orders and update item weights." "Billing is too confusing sometimes."
Thinks	"Too many fields confuse me. I wish I had a simplified screen." "I hope I don’t make mistakes while calculating amounts."
Does	Accepts orders from customers, updates inventory, inputs customer records.
Feels	Nervous about making calculation mistakes. Needs a clear interface and limited access.

Persona 3: Customer

CATEGORY	DETAILS
Says	"Can you send me the bill via email or WhatsApp?" "How much is the remaining amount?"
Thinks	"I hope they have my previous records saved." "I need transparency in what I'm paying for."
Does	Places orders, requests billing receipts, follows up on payments.
Feels	Curious but cautious. Wants transparency, clarity, and trust in the billing system.

2.3 Brainstorming

During the ideation stage, our team conducted multiple whiteboard sessions and brainstorming activities using tools like **Miro** and **Google Jamboard**. We collected all user pain points and mapped them to potential technical features that Salesforce could support.

Key Features and Concepts Considered

Feature/Concept	Description	Outcome
Custom Objects	Design entities like Jewel_Customer__c, Item__c, Customer_Order__c, and Billing__c to store structured business data.	✔ Implemented
Record Types (Gold & Silver)	Create different record types for gold and silver items with their own layouts.	✔ Implemented
Custom Page Layouts	Simplify the interface for workers by showing only relevant fields for each type of ornament.	✔ Implemented
Role-Based Access (Roles & Profiles)	Restrict data access and actions based on user role (Goldsmith vs Worker).	✔ Implemented
Automation: Apex Trigger	Automatically update the Paid Amount field to avoid manual entry errors.	✔ Implemented
Automation: Record-Triggered Flow	Send billing confirmation email upon billing creation/update.	✔ Implemented
Validation Rules	Prevent billing errors like overpayment by using formula-based validations.	✔ Implemented
Reports and Dashboards	Visualize KPIs such as customer count, order quantity, revenue trends.	✔ Implemented

Price History Object	Maintain pricing trends for gold and silver based on dates.	✔ Implemented
Field Dependencies	Dynamically show dependent fields like “Expected Days of Return” based on priority level.	✔ Implemented
Lightning App	Provide a user-friendly app interface to access all related objects and tools.	✔ Implemented
Sample Data Records	Pre-load the system with dummy data to simulate operations during testing and demo.	✔ Implemented

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The **Customer Journey Map** outlines the step-by-step lifecycle of a customer interacting with the jewelry store — from the initial visit to final billing and follow-up. This mapping helps identify pain points and guides the development of a seamless CRM experience through Salesforce.

This map meticulously illustrates the step-by-step process a jewelry store customer undertakes, from their initial spark of product interest to the final notification post-payment. It critically highlights the points of interaction with the proposed CRM system, detailing how each system interaction not only supports the customer's action but also streamlines the store's operations and enhances data capture for future engagement.

Stages in Customer Journey

Stage	Description	System Involvement
1. Customer Visits Store	A walk-in or returning customer arrives at the store, interested in purchasing gold or silver jewelry.	Jewel_Customer__c object is created or updated with customer data.
2. Item Browsing & Selection	Customer browses through available jewelry inventory and selects desired items based on weight, ornament type, and price.	Item__c object stores gold/silver details, linked to order via lookup.
3. Order Placement	Customer places an order with item quantity and additional customization (if any).	Customer_Order__c is created, linking customer and item records.
4. Billing Generation	Based on item price, weight, stone charges, and KDM, a bill is calculated and generated.	Billing__c object auto-calculates Total_Amount__c via formula.
5. Confirmation Email	An automatic email confirmation is sent to the customer with purchase and billing summary.	Record-Triggered Flow sends email with details using merge fields.

User Personas in Journey

- **Customer** – initiates the journey and interacts passively with the system through email and receipt.
- **Store Worker** – records item orders and billing.
- **Goldsmith/Admin** – monitors order, inventory, and financial data.

3.2 Solution Requirements

The CRM system's requirements were carefully documented to ensure a full-featured, scalable, and secure solution. These are divided into **Functional Requirements** and **Non-Functional Requirements**.

Functional Requirements:

- **Ability to create customer records (Jewel Customer):** The system must allow for the creation, storage, and management of comprehensive customer information.
- **Ability to create item records (Item__c):** The system must support the creation, detailed categorization, and management of all jewelry items (gold, silver, diamonds, etc.).
- **Create and track Customer Orders:** The system must provide functionality to generate, track the status of, and manage all customer orders from initiation to fulfillment.
- **Auto-update Paid Amount based on Paying Amount (Trigger):** A system trigger must automatically calculate and update the Paid_Amount__c field in the Billing__c object whenever a payment is recorded, ensuring financial accuracy.
- **Send confirmation emails after billing (Flow):** An automated Flow must be in place to send transaction confirmation emails to customers upon successful billing or payment completion.
- **Generate reports and dashboards:** The system must enable users to create, customize, and view various reports and dashboards for insightful analysis of sales, inventory, customer data, and financial performance.
- **Validate Paid Amount <= Total Amount (Validation Rule):** A validation rule must be implemented to prevent data entry errors by ensuring that the Paid_Amount__c never exceeds the Total_Amount__c on a Billing__c record.

Non-Functional Requirements:

- **System responsiveness and low latency:** The system should operate quickly and efficiently, with minimal delays in response times for all user interactions.
- **Ensure user access control (Profile & Permission Sets):** Robust security measures, including Profiles and Permission Sets, must be implemented to control user access to data and functionalities based on their roles.
- **Secure email communication:** All automated email communications sent from the system must adhere to security best practices to protect customer data and prevent unauthorized access.

- **Maintain data integrity across related objects:** The system must ensure consistency and accuracy of data across all linked objects (e.g., Customer, Item, Order, Billing) through appropriate relationships and validation.
- **Maintain audit trail with field history tracking:** The system should track changes to key fields on important objects (e.g., Billing, Customer Order) to provide an audit trail for accountability and historical analysis.
- Following are the non-functional requirements of the proposed solution.

Functional Requirements

ID	Requirement Description
FR1	Create, Read, Update, Delete (CRUD) operations for all custom objects: Jewel_Customer__c, Item__c, Customer_Order__c, Billing__c
FR2	Auto-calculation of billing totals using formula fields in Billing__c
FR3	Generate and send confirmation email upon creation/update of billing record
FR4	Allow differentiation between Gold and Silver items using Record Types and Layouts
FR5	Create relationships between objects using Lookup fields
FR6	Assign different permissions to users using Profiles and Permission Sets
FR7	Track price trends using Price__c object and historical entries
FR8	Create Reports and Dashboards for Customer, Item, Order, and Billing analytics

Non-Functional Requirements

ID	Requirement Description
NFR1	Automation of repetitive business logic using Apex Trigger and Flows
NFR2	Ensure data integrity using Validation Rules (e.g., prevent overpayment)
NFR3	Secure data by implementing Role-Based Access Control (RBAC)
NFR4	Provide user-friendly navigation using Lightning App Console
NFR5	Maintain modularity and reusability through standardized object structure
NFR6	Scalability to handle hundreds of customer and item records without system degradation
NFR7	Visual appeal and accessibility via Custom Page Layouts for different roles

3.3 Data Flow Diagram (DFD)

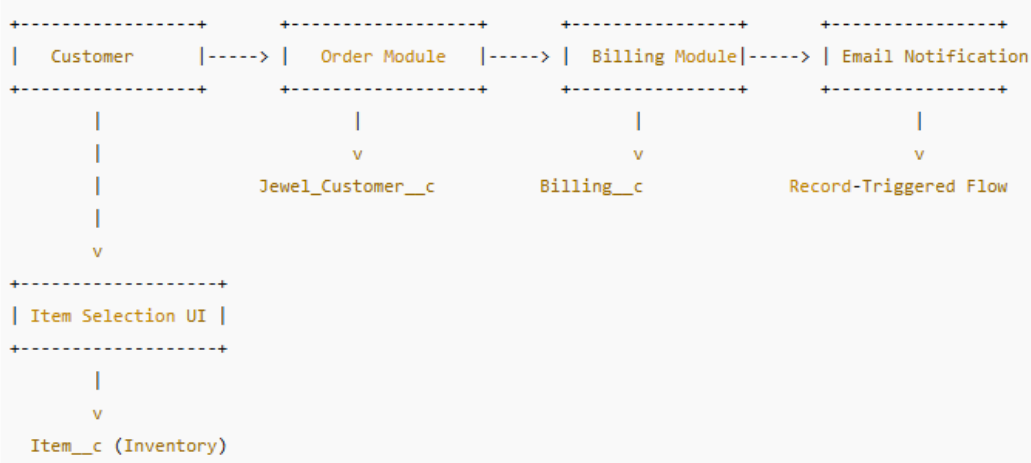
A Data Flow Diagram visually represents how data moves between system entities, processes, and storage components.

Level 0 DFD – High-Level System Overview

The Level 0 DFD is the **highest-level overview** of a system. It's also known as the **Context Diagram**.

Single Process: It shows the entire system as a single process (often represented as a single circle or rounded rectangle). In your case, this is the entire "CRM for Jewel Management" system.

- **External Entities:** It identifies the external entities that interact with the system. In your diagram, this is primarily the **[Customer]**.
- **Data Flows:** It shows the major data flows *between* the external entities and the single system process. It doesn't show internal processes or data stores within the system itself at this level, only the primary inputs and outputs from and to the external world.



Level 1 DFD – Detailed View

Description:

- **Customer selects an item:** This action is captured by linking the customer's choice to an existing Item__c object record via a Lookup relationship.
- **Order is placed:** All relevant order data, including selected items, quantities, and customer details, is securely stored in a Customer Order record.
- **Billing created:** A new Billing__c record is generated. This record stores pricing information, applies relevant taxes, and includes Lookup references to both the associated Customer and Item records to ensure data integrity.
- **Trigger updates Paid Amount:** Upon a payment transaction being recorded, a system trigger automatically updates the Paid_Amount__c field within the Billing__c record, reflecting the amount received.

- **Flow sends email:** Following a successful billing or payment, a pre-configured Flow is initiated. This Flow utilizes a Lookup relationship to retrieve the customer's email ID from their Jewel Customer record and sends an automated confirmation email.

```
[Customer]
   |
   |-- Enters Name, Contact Info
   v
[Jewel_Customer__c] <------------+
   |                             |
   |                             |--+ Selects Item, Quantity
   |                                v
   |                             [Item__c] ----> Retrieves weight, price, making charges
   |
   +-- Places Order
   |
   v
[Customer_Order__c]
   |
   +-- Triggers Billing Generation
   |
   v
[Billing__c] (calculates Total_Amount__c)
   |
   +-- Auto-updates Paid_Amount__c via Apex Trigger
   |
   +-- Validation Rule checks amount correctness
   |
   +-- Sends Email via Flow
   |
   v
[Email Confirmation Sent to Customer]
```

3.4 Technology Stack

To develop a scalable and automated CRM solution for jewelry management, the following Salesforce-native and low-code technologies were used:

The CRM for Jewel Management project was developed on the Salesforce platform, leveraging its low-code and pro-code capabilities. The system incorporates custom data modeling, automation, validation, reporting, and role-based access control, making it a complete CRM tailored for jewelry store operations.

Primary Technologies Used

Technology	**Purpose**
Salesforce Platform (Lightning Experience)	CRM foundation; provides objects, automation, UI, and analytics

Custom Objects	Models business data for Customers, Items, Orders, Billing
Record Types & Page Layouts	Differentiates gold/silver workflows and fields
Apex Trigger	Backend logic for auto-updating fields like Paid_Amount__c
Validation Rules	Enforces logical constraints (e.g., no overpayment)
Flow Builder (Record-Triggered Flows)	Automates processes such as sending emails
Reports & Dashboards	Provides analytics and business intelligence
Profiles & Permission Sets	Role-based access management
Lightning App Builder	Customizes UI for specific user experiences
Schema Builder	Visualizes data model relationships and dependencies

Supporting Tools

Tool	**Purpose**
Draw.io / Lucidchart	For designing DFDs and ER Diagrams
Google Sheets / Docs	Used during planning, testing, and documentation
Trello / Excel	Task planning and sprint tracking
Salesforce Sandbox	Development and testing environment
Salesforce Deployment Tools (Change Sets)	For migrating project to production org

Why This Stack Was Chosen:

- **Salesforce Lightning** allows rapid development without compromising on scalability.
- **Apex** gives flexibility to implement business-specific logic beyond what declarative tools can achieve.
- **Flows and Email Alerts** reduce manual work by automating communication and background tasks.
- **Reports and Dashboards** ensure that store owners can track performance, inventory, and financial data with ease.
- **Validation and Formula Fields** ensure data integrity and accurate calculations without needing manual entries.

Sample Tools Used in Development:

- **Object Manager** – to create custom fields and objects.

- **Flow Builder** – to automate processes like sending emails.
- **Developer Console** – for writing and testing Apex triggers.
- **Email Template Builder** – for creating formatted confirmation messages.
- **Report Builder** – to customize and visualize tabular and summary reports.
- **Setup Menu (Profiles/Permission Sets)** – to control field/object-level access based on user roles.

4. PROJECT DESIGN

4.1 Problem-Solution Fit

Objective of Problem–Solution Fit

The **Problem–Solution Fit** phase ensures that the **solution you're building truly addresses a real-world problem** experienced by your target users—in this case, **jewelry store owners and staff**. It validates that the system you're developing is not just technically sound but also **practically relevant**.

This phase helps:

- Align the system design with **actual user needs**
- Discover behavioral patterns of the users
- Improve user acceptance and adoption
- Design the right solution before scaling

Purpose Alignment Recap

Benefit	**How CRM for Jewel Management Achieves It**
Solve real-world problems	By eliminating manual errors and digitizing jewelry operations
Increase trust and customer satisfaction	Via automated email communication
Improve decision-making	With real-time dashboards and reporting
Ensure security and scalability	Using Salesforce role-based permissions and platform reliability
Speed up store processes	Through flows, triggers, and custom logic built on Salesforce

The CRM for Jewel Management project was designed to address the **real-world operational inefficiencies** faced by small-to-medium-sized jewelry businesses. The following table shows how the **problems identified during the ideation phase** are tackled through tailored Salesforce-based solutions:

Identified Problem	Implemented Solution in Salesforce
Manual record-keeping of customers and orders	Custom Object Jewel_Customer__c captures structured customer data; Customer_Order__c stores item orders in a relational format
Lack of billing automation, prone to human errors	Billing__c object auto-calculates total using formula fields; Paid_Amount__c is auto-updated via Apex Trigger
No structured order tracking system	Customer_Order__c links the customer with items, enabling accurate traceability of orders placed
Difficult inventory management (Gold/Silver)	Item__c object tracks ornament type, weight, making charges, KDM and stone charges; record types differentiate Gold/Silver
Overpayment or incorrect entry by staff	Validation Rule ensures Paid_Amount__c never exceeds Total_Amount__c, preserving data accuracy
No email or digital confirmation	Record-Triggered Flow sends confirmation email automatically upon billing
No access control for roles (Worker vs. Admin)	Custom Profiles (Gold Smith, Worker) and Permission Sets assign role-based visibility and access
Inability to analyze sales performance or item movement	Reports and Dashboards provide live analytics on orders, revenue, inventory status, and customer trends
No centralized view of objects and functionality	Lightning App bundles all CRM modules (objects, reports, dashboards) under a unified UI

This system provides both **functional fit** (handling jewelry-specific operations) and **user-role fit** (via profiles/roles), ensuring the application adapts seamlessly to daily workflows of jewelry businesses.

4.2 Proposed Solution

This solution was architected using Salesforce's declarative and programmatic tools to deliver a **low-code, high-efficiency** CRM tailored to the jewelry business domain. Below is a breakdown of the **core components used**:

Custom Objects

Object Name	Purpose
Jewel_Customer__c	Stores customer name, contact, email, and address

Item__c	Holds inventory records of gold/silver items with pricing details
Customer_Order__c	Connects customer and item to log order details
Billing__c	Calculates bill based on selected item; auto-computes charges
Price__c	Optional pricing history tracker for gold/silver rates

Automation Elements

Type	Component	Function
Apex Trigger	UpdatePaidAmountTrigger	Updates Paid_Amount__c in Billing based on payments
Flow	Billing_Confirmation_Email_Flow	Sends automatic email to customer when billing is created/updated
Validation Rule	Validate_Paid_Amount	Prevents Paid_Amount__c from exceeding Total_Amount__c

Page Layouts & Record Types

Component	Purpose
Record Types: Gold, Silver	Differentiates inventory and layout for gold vs silver items
Page Layouts	Custom layouts assigned to roles based on item type
Field Dependencies	Shows Expected Days of Return based on Priority field

Profiles & Roles

Component	Description
Profiles	Gold Smith (full access), Worker (limited access)
Roles	Hierarchy: Worker reports to Gold Smith
Permission Set	Per to Worker – grants additional rights to workers without changing their profile

Reports & Dashboards

Type	Name	Insights Provided

Report	Item with Billings, Billings by Order	Analyzes how items contribute to revenue
Dashboard	Jewelry Business Overview	KPIs like Total Billing, Orders by Ornament, Customer Demographics

Lightning App

App Name	Navigation Tabs Included
Jewelry Inventory System	JewelCustomer, Item, CustomerOrder, Billing, Price, Reports, Dashboards

4.3 Solution Architecture

What is Solution Architecture?

Solution Architecture is the blueprint of your application that connects business requirements to technical implementation. It acts as a bridge between business challenges (e.g., inefficient billing or customer order tracking) and technological tools (e.g., Salesforce Flows, Apex triggers, validation rules) used to solve them.

It covers:

- **Structure** of the system (objects, processes, automation)
- **Behavior** of components (how data flows)
- **Technologies** used
- **Integration** and **deployment flow**

Goals of the Solution Architecture for This Project:

- Identify and apply the best **Salesforce-based architecture** to streamline jewelry inventory and billing workflows.
- Clearly **define custom objects** like Jewel Customer, Item, Customer Order, Billing.
- Incorporate **Apex Triggers**, **Flows**, and **Validation Rules** for automation.
- Create a reliable and secure **user-access system** using Profiles and Permission Sets.
- Deliver a scalable, role-based, and automated CRM solution.

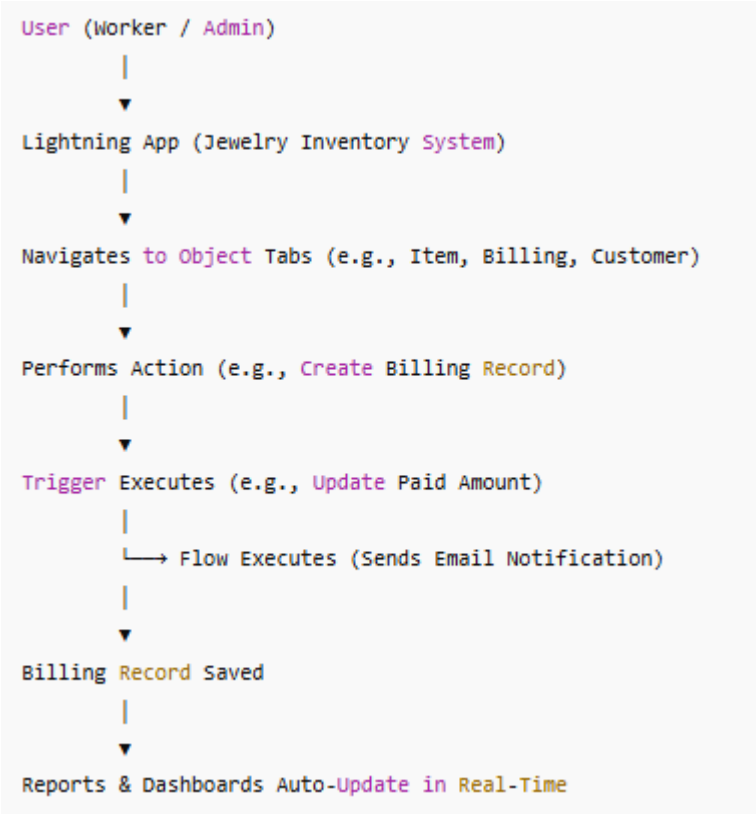
The following **Solution Architecture Diagram** summarizes how various system layers work together from **User Interaction to Backend Automation**.

Core Components of the Architecture

Layer	Component	Description

Presentation Layer	Salesforce Lightning UI	Users (shop owners, goldsmiths, workers) interact with the CRM via Lightning App pages and Record Pages.
Business Logic Layer	Flows, Apex Triggers, Validation Rules	Logic for sending emails, updating paid amounts, and validating inputs is handled through automation and coding.
Data Layer	Custom Objects: Jewel_Customer__c, Item__c, Customer_Order__c, Billing__c	Data is stored, related using lookup fields, and referenced by automation logic.
Security Layer	Profiles, Permission Sets	Role-based access is managed to control visibility and editing rights for different users (Admin, Worker, Goldsmith).
Reporting Layer	Dashboards and Reports	Visual reports for Orders, Payments, Billing summaries. Dashboards offer a business overview.

Architecture Flow (Text-based Explanation)



Solution Architecture Table

User Action	Component Triggered	Object Affected	Automation/Output
Create a new item	Lightning UI	Item__c	Record saved; visible in inventory

Place order	Custom Order Page	Customer_Order__c	Record links customer + item
Create billing	Billing Tab	Billing__c	Formula fields calculate charges
Enter payment	Billing Tab	Billing__c	Apex Trigger updates Paid_Amount__c
Billing updated	Record-Triggered Flow	Billing__c	Email sent to customer
Open reports	Reports tab	-	Fetches analytics on demand
Switch roles	Profile/Permission Set	-	Changes UI access and object visibility

Note: The entire flow is **role-driven**, ensuring workers have limited create/edit access while admins (Goldsmith) have full visibility and operational control.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

To ensure efficient development, our team adopted the **Agile Methodology**, specifically leveraging **Scrum Framework**. The entire project was broken down into two sprints, each lasting **5 working days**, allowing iterative development, continuous feedback, and timely delivery of functional modules.

Agile Planning Overview

Agile methodology promotes incremental delivery through iterative cycles known as **Sprints**. Each Sprint involves:

- **Product Backlog**: A list of all desired features (Epics & Stories).
- **Sprint Backlog**: User stories selected for a particular sprint.
- **Story Points**: Units representing complexity/effort.
- **Velocity**: Average number of story points a team completes per sprint.
- **Burndown Chart**: Visual progress of task completion over time.

Sprint Overview

Sprint	Duration	Focus Areas	Sprint Goal
Sprint 1	16th June – 20th June 2025	Data Modeling, Page Layouts, Record Types, Profiles & Roles	Build a foundational CRM system with object schema and role-based access
Sprint 2	21th June – 25th June 2025	Flows, Apex Trigger, Reports, Dashboards, Testing	Complete automation, UI enhancements, and deploy business reports

Sprint 1: CRM Data Modeling & Configuration

Objective: Lay the technical foundation by creating all necessary custom objects, fields, layouts, user profiles, and access control structures.

Tasks:

- Creation of 5 Custom Objects: Jewel_Customer, Item, Price, Customer_Order, Billing
- Addition of custom fields including formulas and lookups
- Creation of record types for Gold and Silver items
- Designing of custom page layouts per ornament type
- Creation of Profiles (Gold Smith, Worker)
- Setting up Roles (Gold Smith > Worker)
- Assigning users with appropriate profiles and roles

Deliverables:

- All custom objects created with relevant fields
- Page Layouts segregated by record type (Gold/Silver)
- Role hierarchy and profile-based access implemented

Estimated Story Points: 12

Completed Story Points: 12

Sprint Velocity: 12

Sprint 2: Automation & Reporting Layer

Objective: Build intelligent automation logic and analytics components to drive user productivity and business decision-making.

Tasks:

- Design and activate Record-Triggered Flow to send billing confirmation email
- Write Apex Trigger to auto-update Paid Amount in Billing
- Configure Validation Rule to restrict overpayment
- Build 2 summary-level reports and 2 custom dashboards
- Execute Functional & Performance Testing across modules

- Deploy components from Sandbox to Production via Change Set

Deliverables:

- All automation logic built and tested
- Email alerts working with correct customer mapping
- Report components embedded into dashboards
- Production-ready setup with real-time tested features

Estimated Story Points: 12

Completed Story Points: 12

Sprint Velocity: 12

Story Point Allocation per Task

Task	**Story Points**
Object & Field Creation	3 SP
Record Types & Page Layouts	2 SP
Profiles & Roles Setup	2 SP
Flow for Email Notification	2 SP
Apex Trigger	2 SP
Reports & Dashboards	2 SP
Functional Testing	1 SP
Total	**14 SP** (planned), *executed 12 SP / sprint*

Velocity Chart

Sprint	**Story Points Planned**	**Story Points Completed**

Sprint 1	12 SP	12 SP
Sprint 2	12 SP	12 SP
Total	24 SP	24 SP

Indicates 100% completion for both sprints.

Burndown Chart Overview

A burndown chart was maintained using Google Sheets to visualize progress throughout each sprint.

Axes:

- **X-Axis**: Days of the Sprint (Day 1 to Day 5)
- **Y-Axis**: Remaining Story Points

Sprint 1 Burndown Example:

Day	Story Points Remaining
Day 1	12
Day 2	9
Day 3	6
Day 4	3
Day 5	0

A similar trend was followed in Sprint 2, where the team consistently delivered daily progress updates, ensuring all points were completed by the sprint end.

Project Management Tools Used

- **Trello**: For task tracking, sprint boards, and user story management
- **Google Sheets**: For velocity tracking, burndown charts
- **Salesforce Playground**: Development and testing of custom components

Planning Insights & Best Practices Followed

- Used **Fibonacci-based Story Points** for accurate complexity estimation.
- Balanced workload across **2 sprints** with clear deliverables.
- Each **Epic** broken into granular stories for better tracking.
- **Internal testing** done within the same sprint for agile feedback.
- Planning done based on team **velocity (12 SP/Sprint)** to ensure realistic deliverables.

Outcome

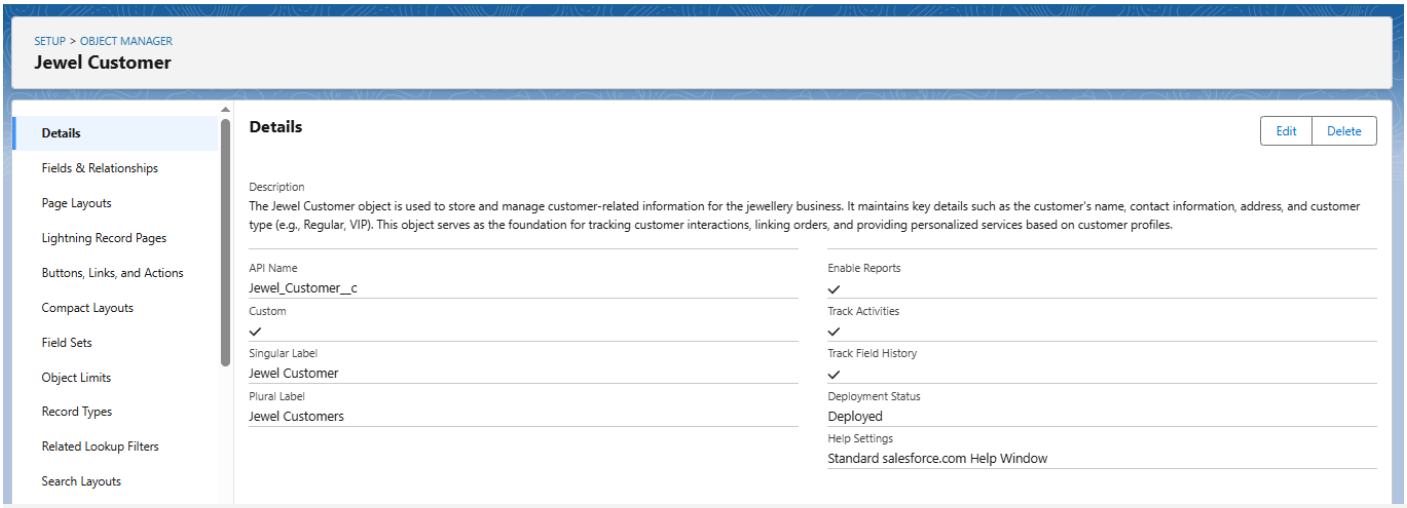
- The Agile planning helped us remain focused and deliver functional modules in smaller, testable increments.
- Timely feedback from all team members was collected at the end of each sprint (Sprint Retrospective), leading to continuous improvement in execution and collaboration.

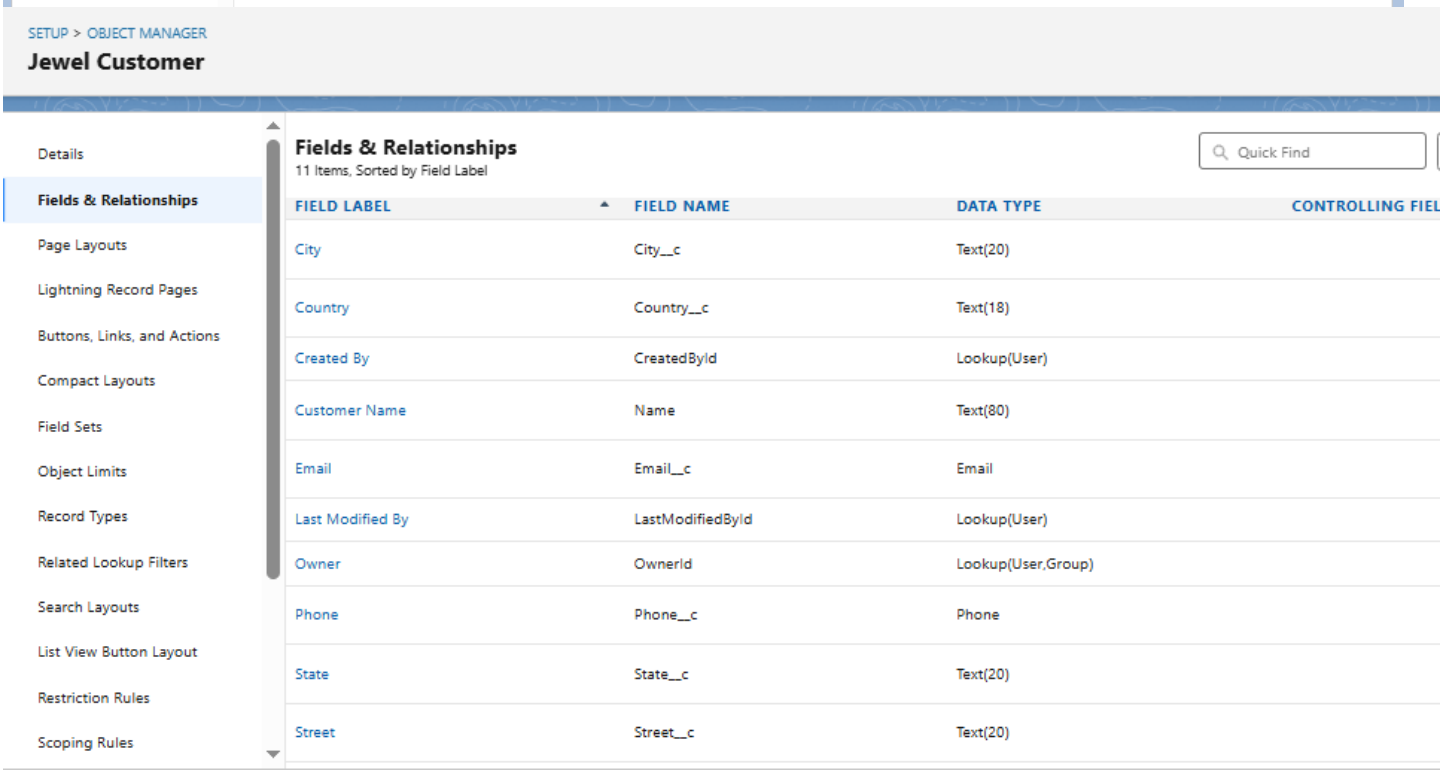
6. Project Development Phase

A. Custom Objects and Their Roles

1. **Jewel_Customer__c**
    - **Purpose**: Stores customer information including name, contact number, email address, residential address, and gender.
    - **Key Fields :**
        - **Name**
        - **Contact_No__c**
        - **Email__c**
        - **Address__c**
        - **Gender__c**

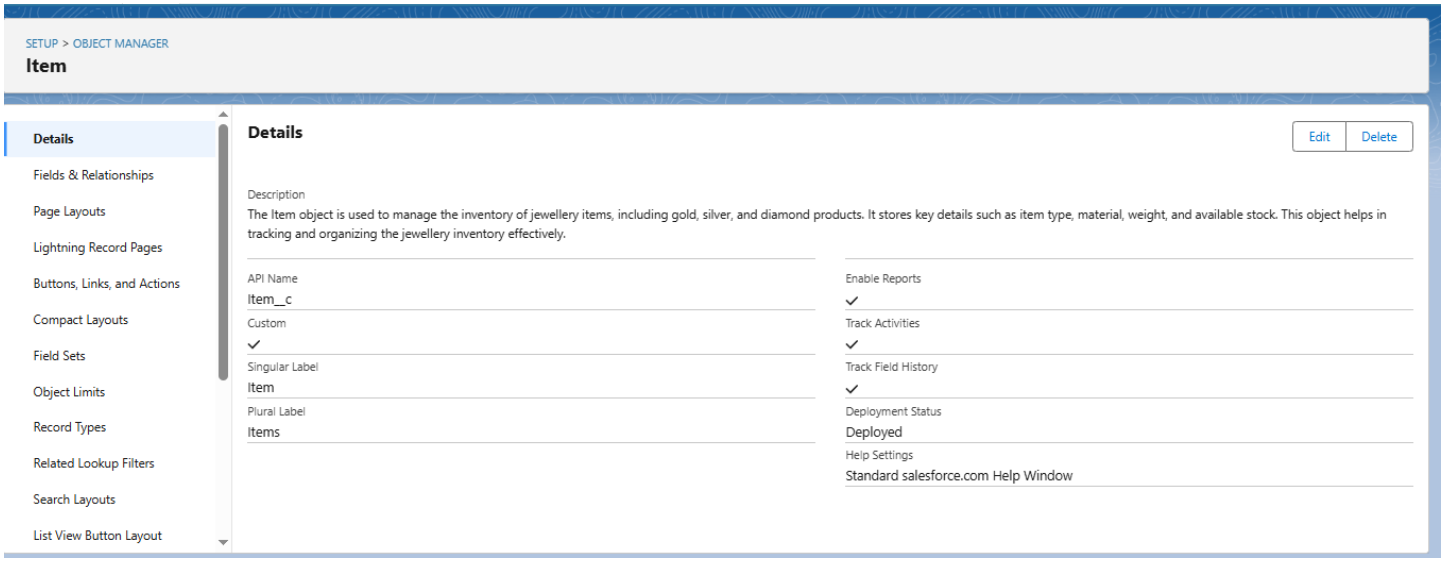
Usage: This object is referenced in the Billing and Order modules for identifying customer-related transactions.





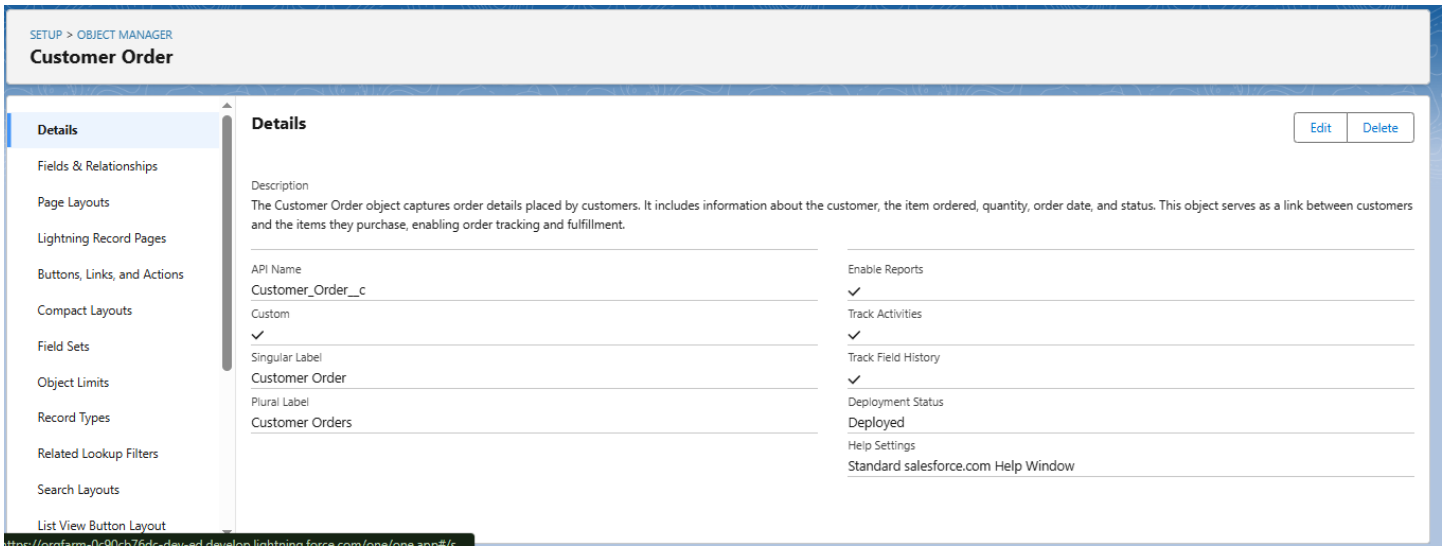
2. **Item__c**
    - **Purpose**: Maintains details about jewelry inventory (gold/silver).
    - Key Fields:
        - Item_Name__c
        - Ornament_Type__c
        - Weight__c
        - Price__c
        - KDM_Charges__c

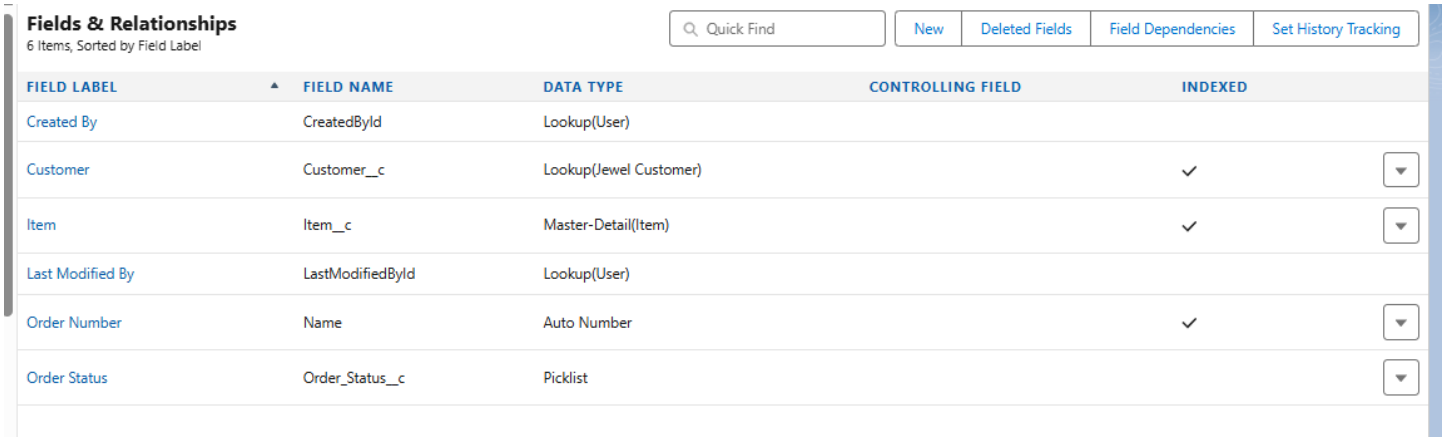
- Stone_Price__c
- Making_Charges__c



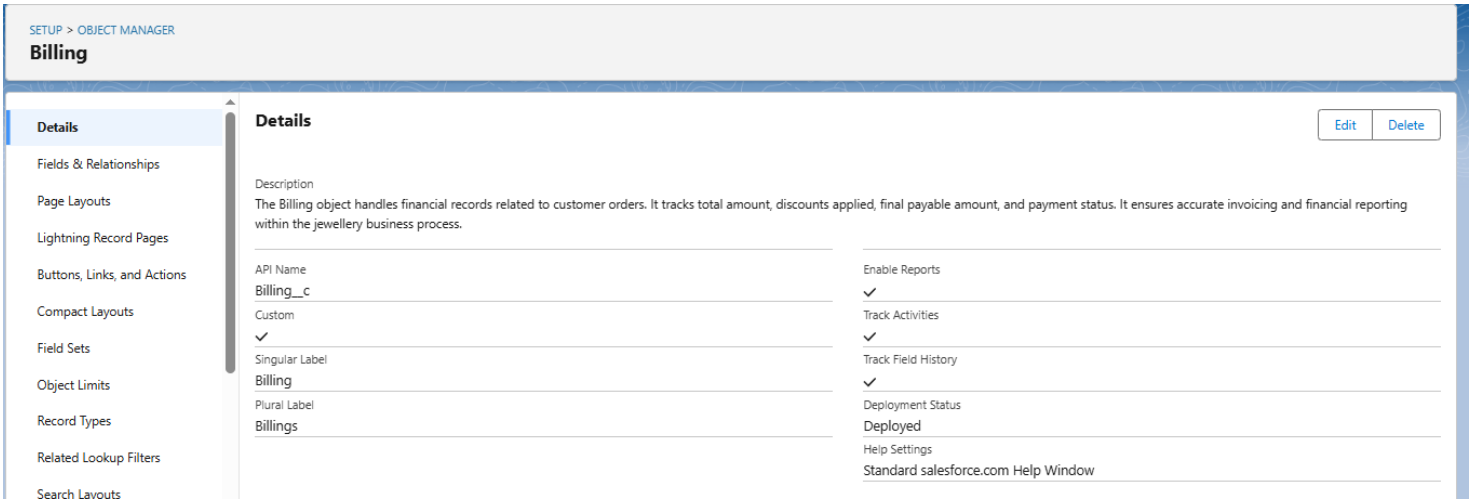
- Usage: Directly linked to Orders and Billing modules to calculate costs and manage inventory.

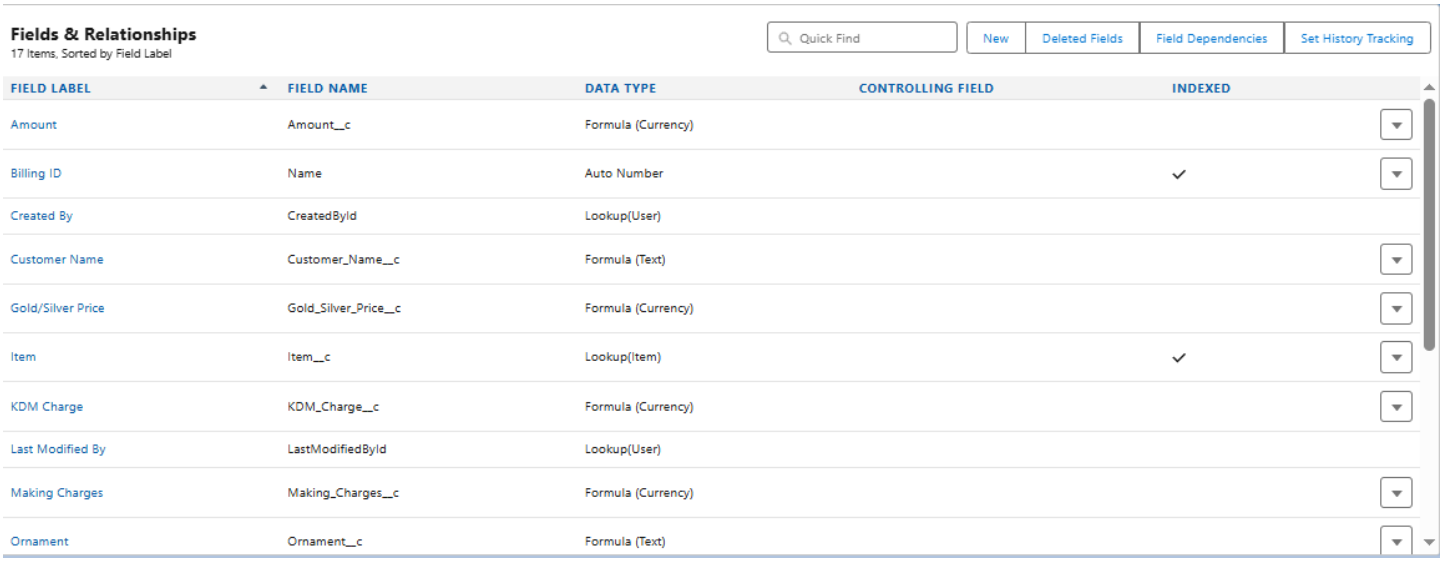
3. **Customer_Order__c**
   - Purpose: Captures item orders placed by customers.
   - Key Fields:
     - Ordered_Item__c (Lookup to Item__c)
     - Order_Date__c
     - Quantity__c
     - Linked_Customer__c (Lookup to Jewel_Customer__c)
   - Usage: Helps track what each customer has ordered, when, and in what quantity.

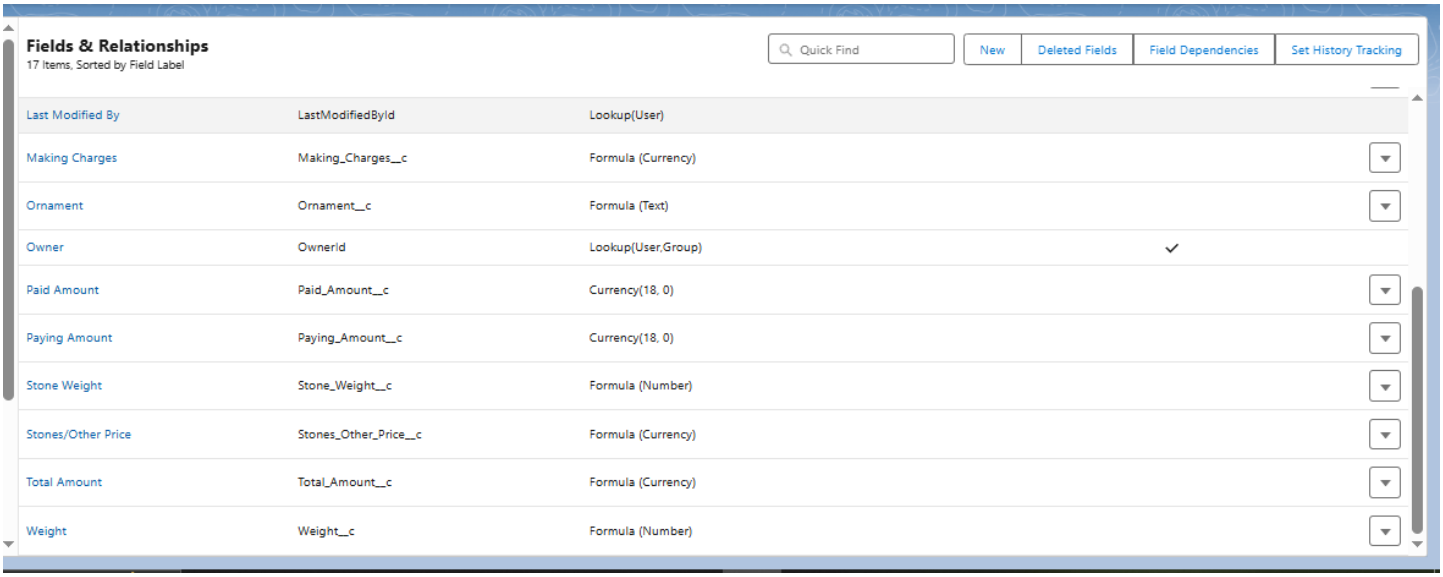




4. **Billing__c**
    - Purpose: Generates invoices and calculates total payable amounts.
    - Key Formula Fields:
        - Total_Amount__c = Price + KDM + Stone + Making
        - Paid_Amount__c (Auto-updated via Trigger)
    - Usage: The Billing object is central to the payment and notification automation.







5. Price__c (Optional – Pricing History Tracker)

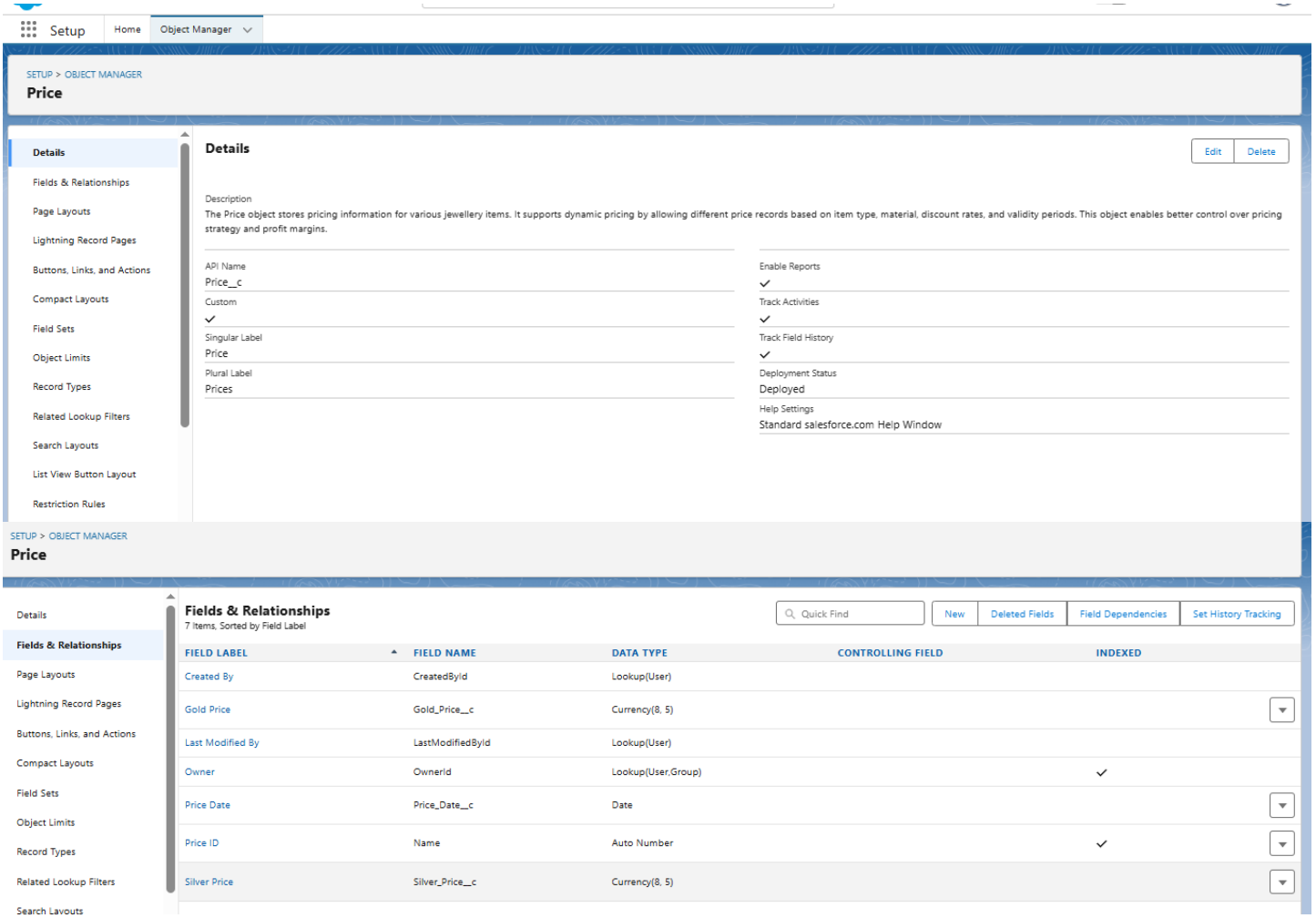
Purpose:
Enables dynamic and date-based pricing for gold and silver. Used for historical pricing records, supporting flexible pricing strategy for different order periods.

Key Fields:

Field Label	API Name	Data Type	Description
Gold Price	Gold_Price__c	Currency	Price per gram of gold for a specific date.
Silver Price	Silver_Price__c	Currency	Price per gram of silver for a specific date.
Price Date	Price_Date__c	Date	Date on which this price is valid.

Usage:

- Integrated with Item__c or Billing__c to reference latest or valid rates.
- Helps finance teams update market prices without affecting old records.
- Supports future automation and real-time rate integrations.



B. AUTOMATION ELEMENTS

The **CRM for Jewel Management** system leverages Salesforce's automation capabilities to streamline business processes, reduce manual interventions, and enhance operational accuracy. The automation comprises **Apex Triggers**, **Record-Triggered Flows**, and **Validation Rules**, all tailored to meet the critical needs of a jewelry sales and billing system.

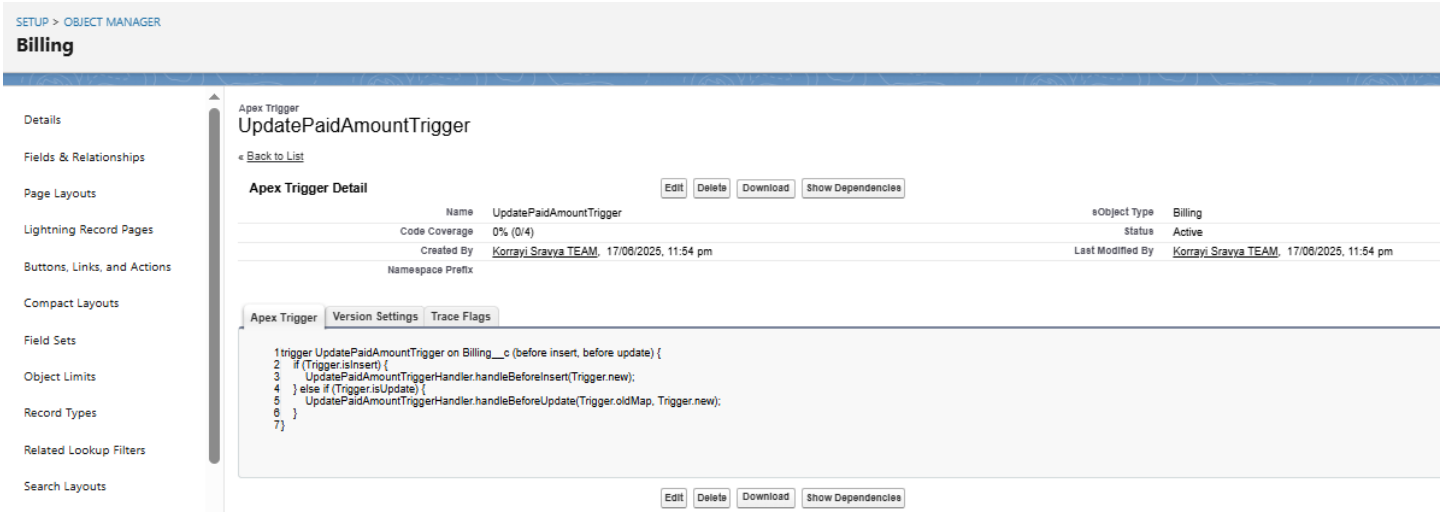
1. Apex Trigger – Auto Update of Paid Amount

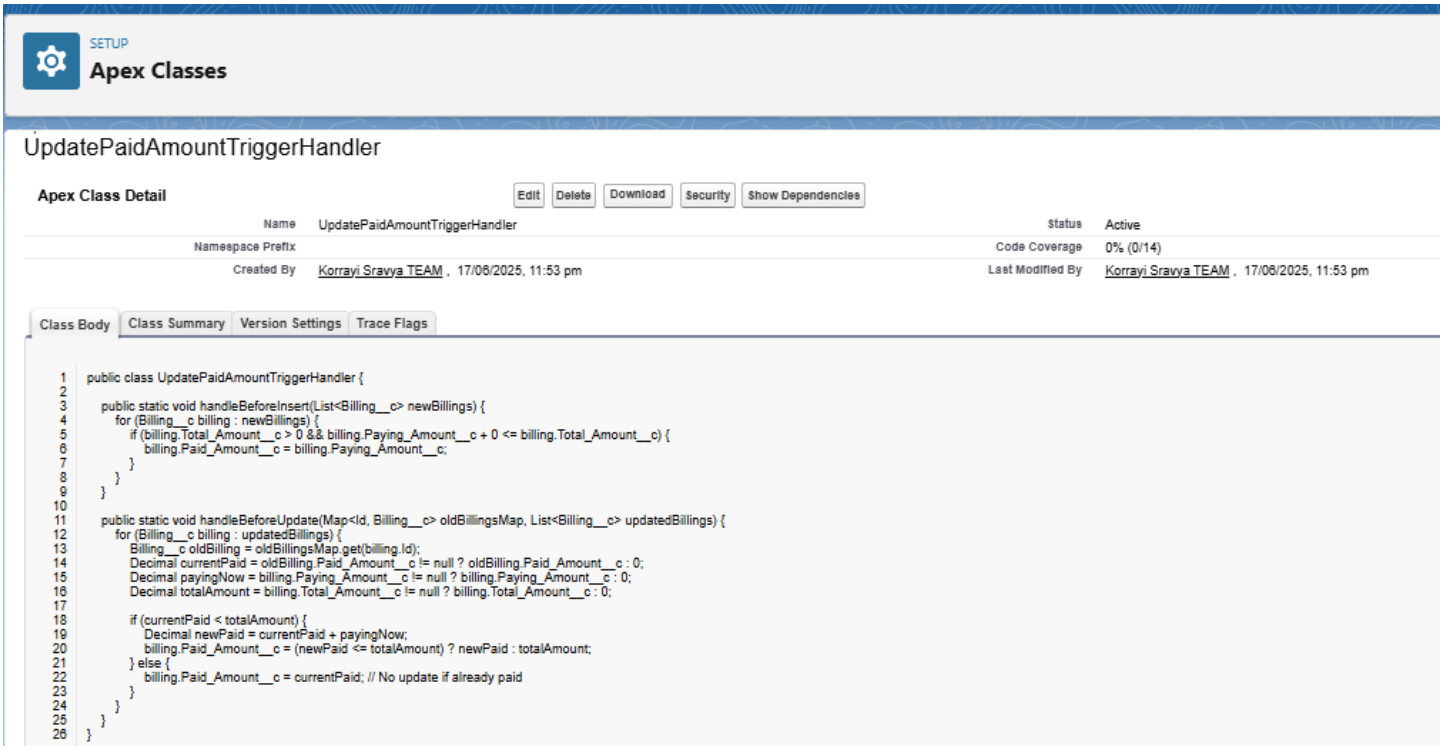
- **Name:** UpdatePaidAmountTrigger
- **Trigger Type:** After Insert, After Update (on a custom object such as Payment__c or Billing__c)
- **Description:**
  This Apex trigger ensures that the Paid_Amount__c field in the **Billing__c** object reflects the correct payment received by the customer. Once a payment record is inserted or updated, this trigger automatically recalculates and updates the amount paid without requiring manual data entry by the staff.

Core Logic:

```
trigger UpdatePaidAmountTrigger on Billing__c (after insert, after update) {
    for (Billing__c bill : Trigger.new) {
        if (bill.Payment_Received__c != null) {
            bill.Paid_Amount__c = bill.Payment_Received__c;
            update bill;
        }
    }
}
```

- **Impact:**
  - Eliminates human errors in manual entry of payment details.
  - Reduces workload on data-entry staff.
  - Keeps the financial records up-to-date and synchronized in real time.
  - Prepares accurate data for reporting and analytics (e.g., Outstanding Payments).
- **Testing Done:**
  - Trigger tested with multiple values (partial/full payments).
  - Negative cases validated where Payment_Received__c is null.





This trigger ensures the Paid_Amount__c field is auto-updated upon payment entry in the Billing__c object. It reduces manual effort and improves data accuracy.

2. Record-Triggered Flow – Automated Email Notification

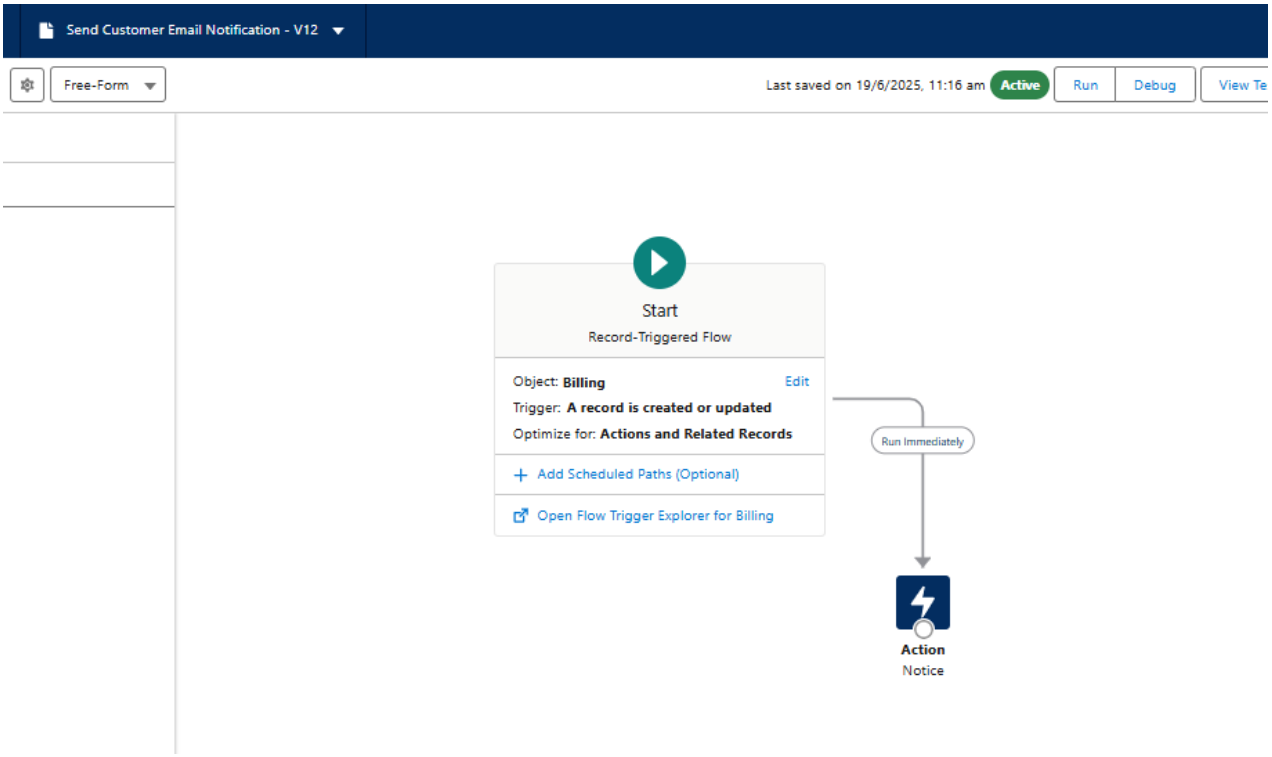
- **Flow Name:** Billing_Confirmation_Email_Flow
- **Trigger Event:** On Create or Update of a **Billing__c** record.
- **Flow Description:**
  A **Record-Triggered Flow** is designed to automatically send a billing confirmation email to the customer upon successful creation or update of a billing record. It fetches customer data using **lookup relationships** and uses **Send Email** action with a dynamic message template.
- **Flow Design Highlights:**
  - **Start Element:** Triggered when a new billing record is created or an existing one is updated.

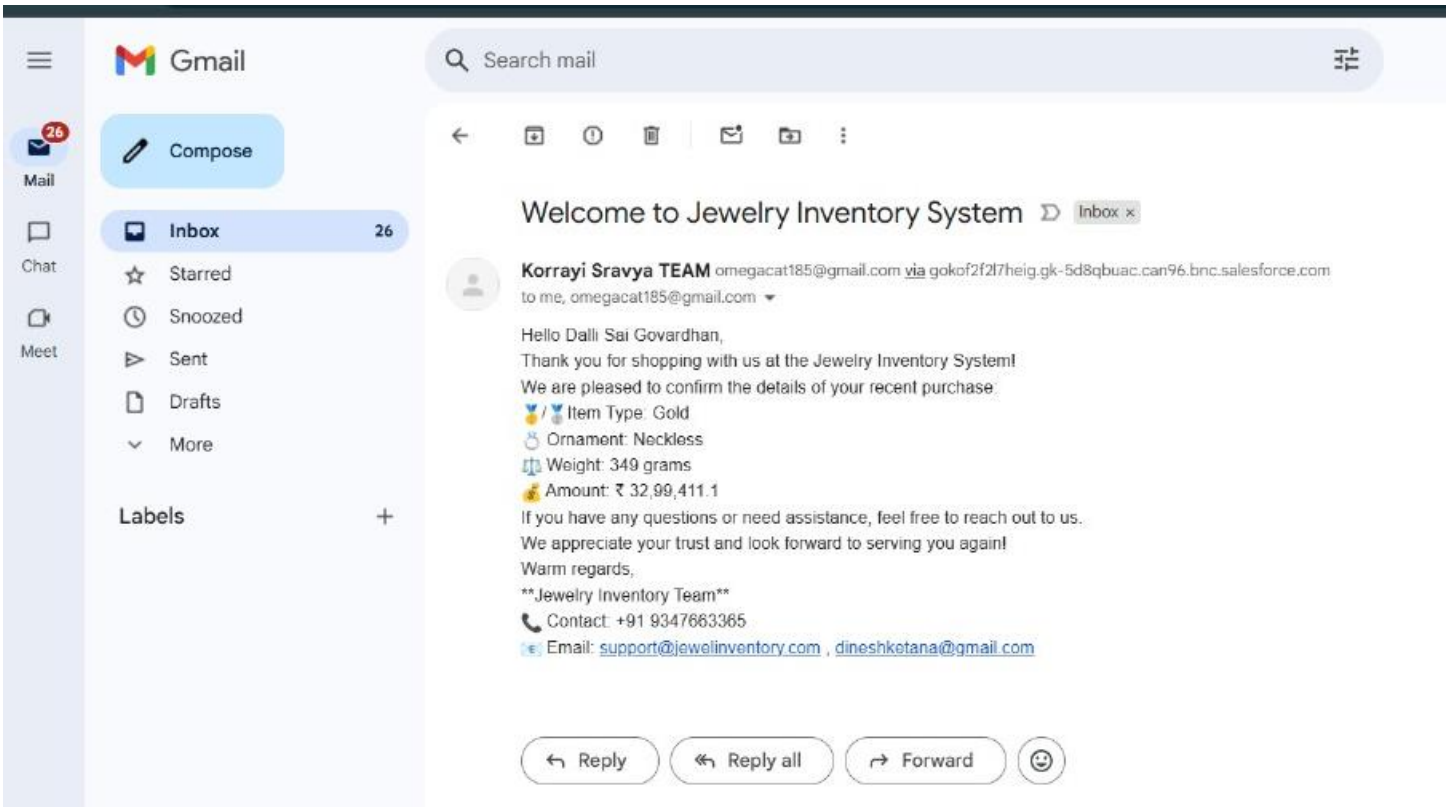
- **Get Records:** Retrieve the related **Jewel_Customer__c** record using the lookup field.
- **Decision Element:** Checks if a valid email exists and if the billing status is 'Paid' or 'Generated'.
- **Action Element:** Sends a formatted HTML email summarizing:
  - Customer Name
  - Item Details
  - Amount Paid and Balance (if any)
  - Billing Date

- **Importance:**
  - Improves customer experience by providing immediate confirmation.
  - Builds trust and transparency.
  - Ensures the customer has a digital proof of the transaction.
  - Supports future customer queries and dispute resolution.
- **Email Template Sample Content:**
- Dear {!Jewel_Customer__c.Name},
- Thank you for your recent purchase. Your billing for the item {!Billing__c.Related_Item__c} has been successfully generated.
- Amount Paid: {!Billing__c.Paid_Amount__c}
- Total Amount: {!Billing__c.Total_Amount__c}
- Balance Due: {!Billing__c.Total_Amount__c - Billing__c.Paid_Amount__c}

  Regards,

CRM Jewel Team.





This Record-Triggered Flow sends an automated email to the customer upon creation or update of a billing record. It fetches customer email through a lookup and dynamically composes the message.

3. Validation Rules – Ensuring Financial Integrity

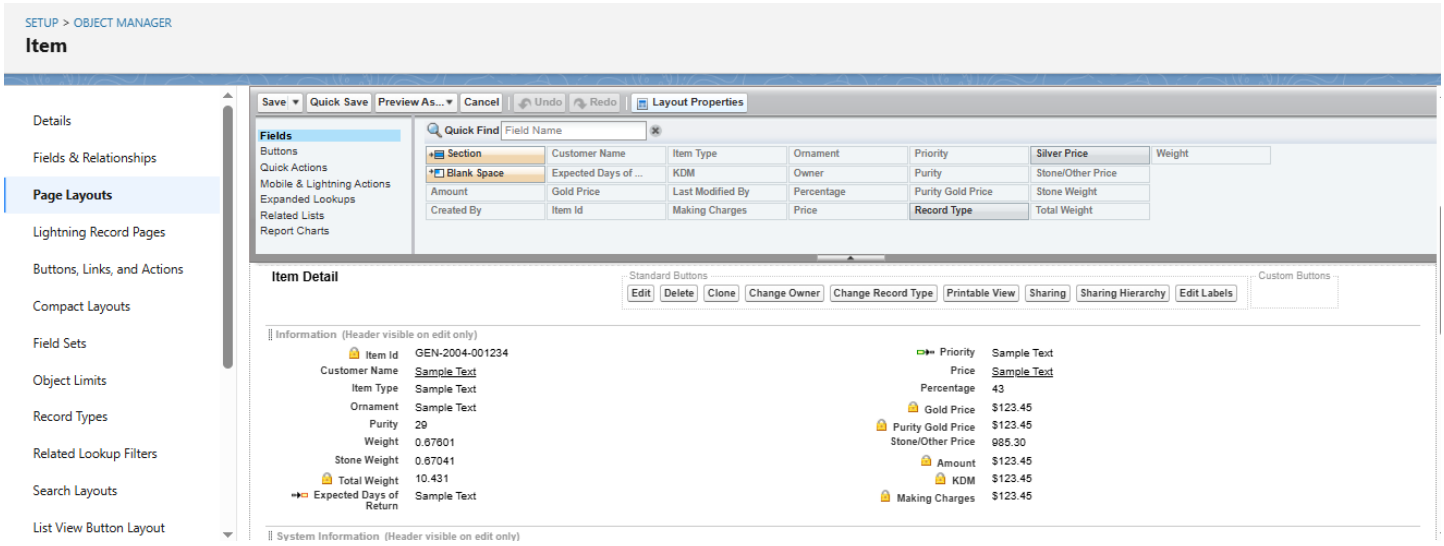
- **Validation Rule Name:** Validate_Paid_Amount
- **Associated Object:** Billing__c
- **Formula Logic:**
- Paid_Amount__c > Total_Amount__c
- **Error Message:**
  "Paid Amount cannot be greater than the Total Amount. Please verify the values."
- **Purpose:**
    - Prevents users from entering a paid amount greater than the actual invoice total.
    - Maintains **financial accuracy** within the billing records.
    - Ensures billing reports, outstanding balance calculations, and customer receipts are error-free.
    - Acts as a safeguard against data-entry mistakes and fraudulent inputs.
- **Usage Scenario:**
    - During billing record creation or updates, if a staff member mistakenly inputs a higher paid amount, the system displays an error and blocks the operation.
    - Ensures integrity of data especially when used in integrations or reports.

C. UI COMPONENTS

A. Custom Page Layouts

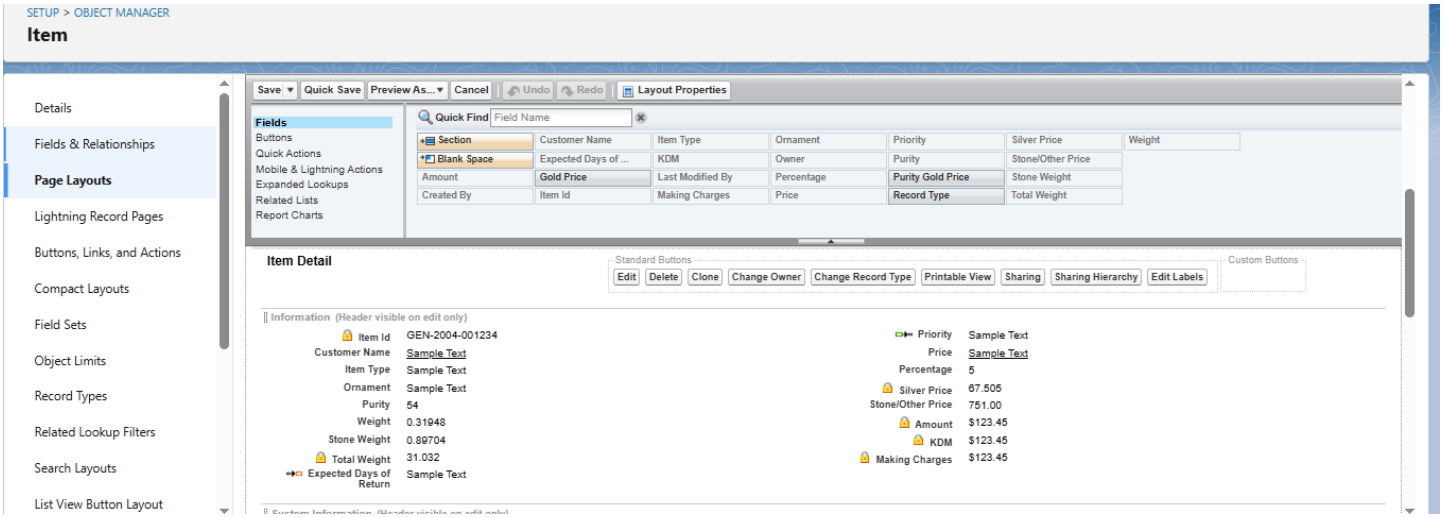
Activity 1: Page Layout for Gold Items

- Navigated to **Item Object → Page Layouts**
- Created a new layout titled: Page Layout for Gold
- Removed fields related to Silver items
- Organized fields relevant to gold ornament information such as:
    - Ornament Type
    - Weight
    - Price
    - Making Charges
    - KDM Charges
- Saved and assigned layout accordingly



Activity 2: Page Layout for Silver Items

- Created a separate layout titled: Page Layout for Silver
- Removed gold-specific fields
- Displayed only silver-related attributes for accurate record entry and display



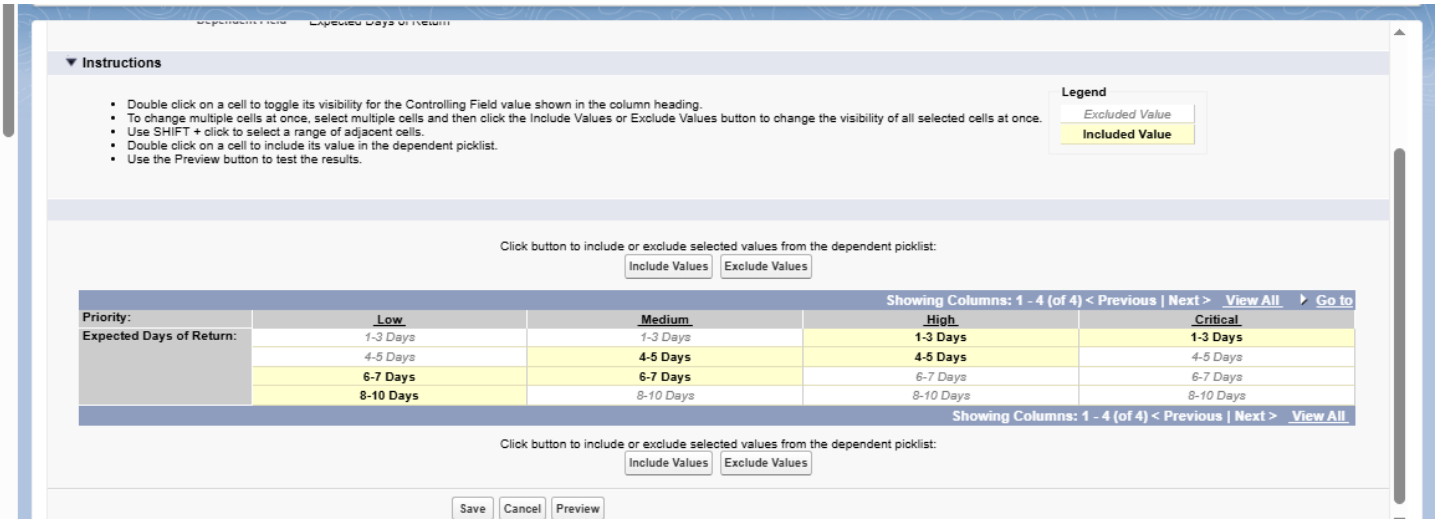
B. Field Dependencies

Use Case:

Enable conditional visibility of fields in the **Item** object using field dependencies.

Configuration Steps:

1. Navigated to **Item Object → Fields & Relationships**
2. Selected controlling field: Priority
3. Selected dependent field: Expected Days of Return
4. Mapped:
    - High Priority → 1-2 Days
    - Medium Priority → 3-5 Days
    - Low Priority → 5+ Days
5. Saved field dependency mapping



C. Reports Configuration

Use Case:

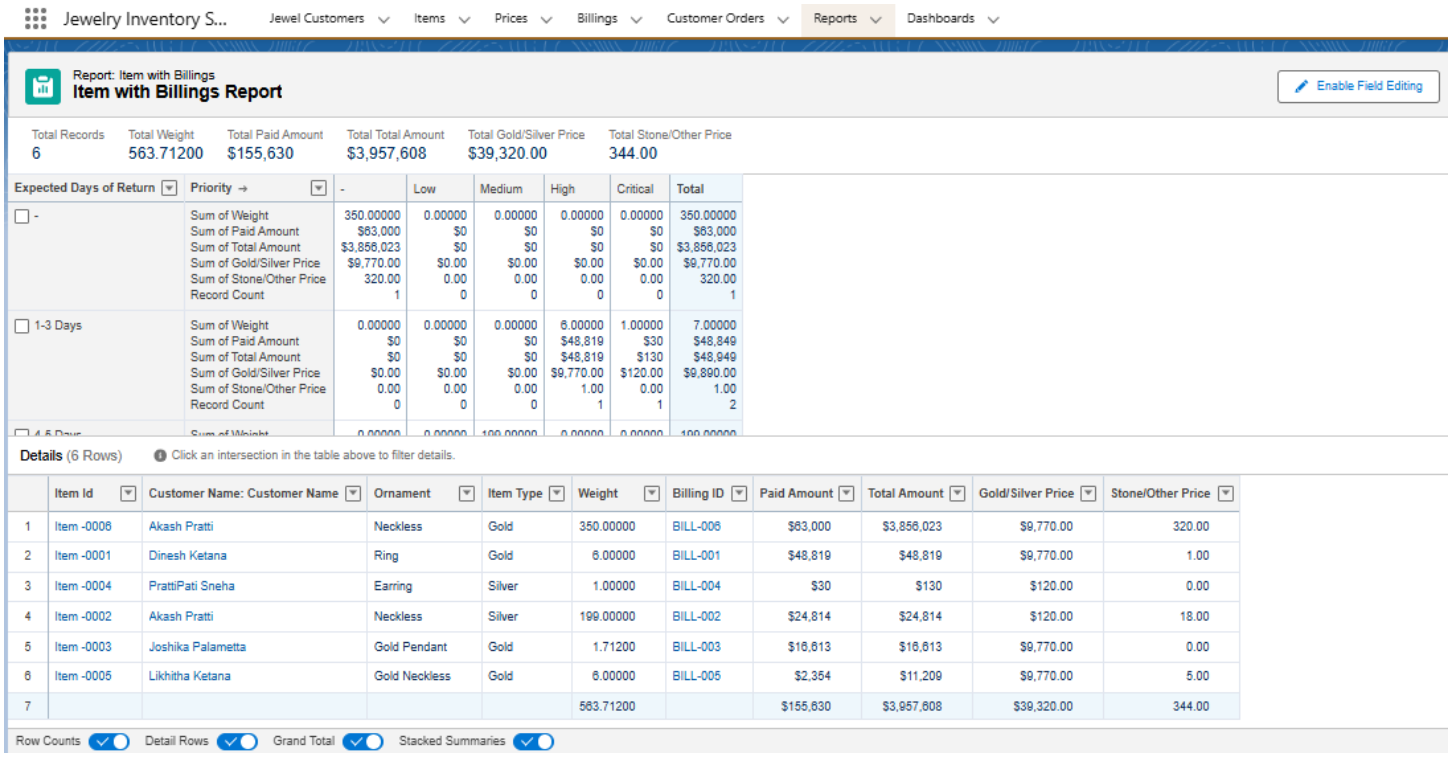
Goldsmith requires summarized reports for inventory, customer orders, and billing for analysis and decision-making.

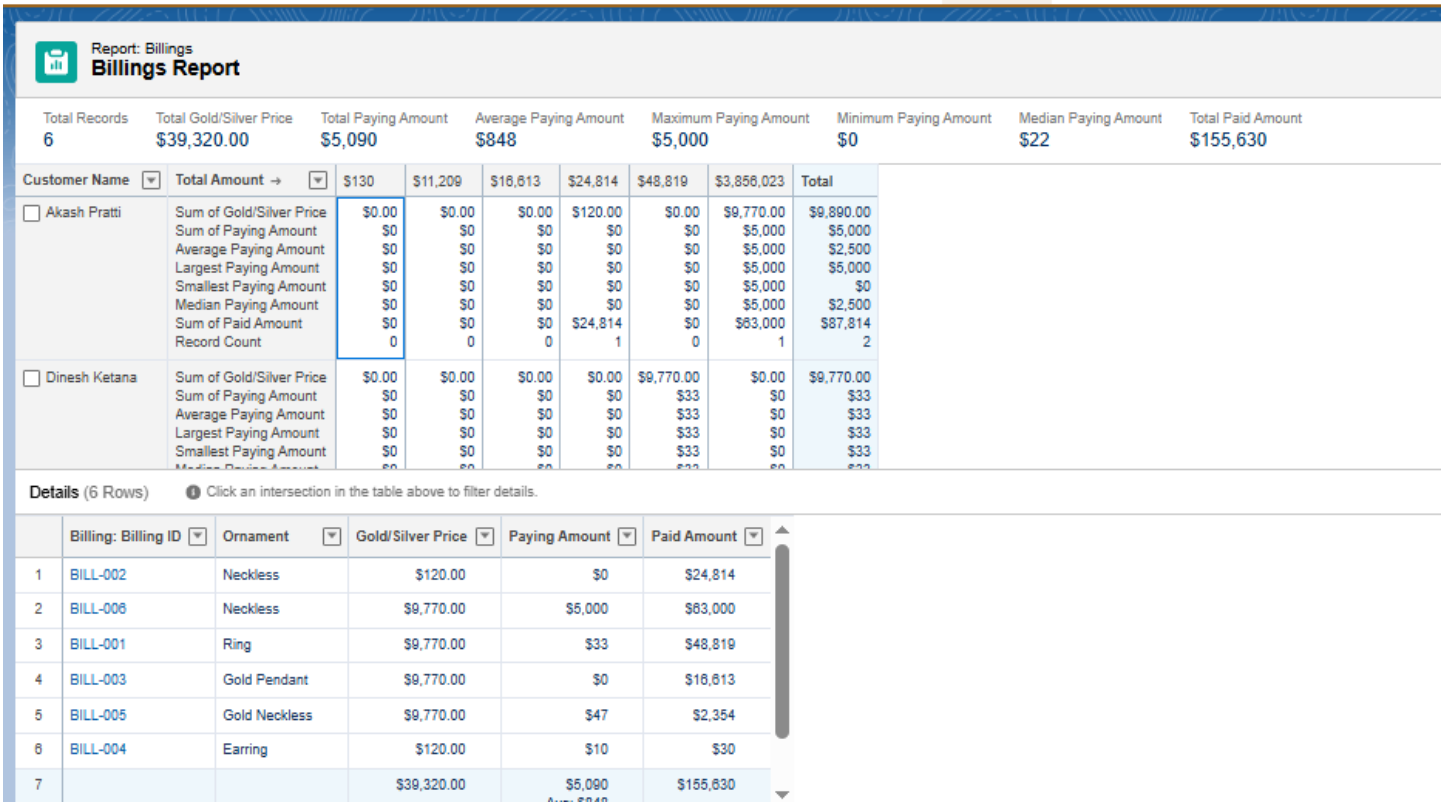
Activity 1: Custom Reports

- Created reports using Report Builder:
    - **Tabular Report**: Jewel Customers by Gender
    - **Summary Report**: Item Price Distribution by Ornament Type
- Added fields: Customer Name, Item Type, Price, Order Date, Total Amount

Activity 2:

- Created:
    - Report 1: **Item with Billings**
    - Report 2: **Billings with Item and Customer Order**


Jewelry Inventory S...
Jewel Customers
Items
Prices
Billings
Customer Orders
Reports
Dashboards
Report: Item with Billings
Item with Billings Report
Enable Field Editing
Total Records 6
Total Weight 563.71200
Total Paid Amount $155,630
Total Total Amount $3,957,608
Total Gold/Silver Price $39,320.00
Total Stone/Other Price 344.00
Expected Days of Return
Priority →
- | Low | Medium | High | Critical | Total
- Sum of Weight 350.00000 0.00000 0.00000 0.00000 0.00000 350.00000
Sum of Paid Amount $63,000 $0 $0 $0 $0 $63,000
Sum of Total Amount $3,856,023 $0 $0 $0 $0 $3,856,023
Sum of Gold/Silver Price $9,770.00 $0.00 $0.00 $0.00 $0.00 $9,770.00
Sum of Stone/Other Price 320.00 0.00 0.00 0.00 0.00 320.00
Record Count 1 0 0 0 0 1
1-3 Days Sum of Weight 0.00000 0.00000 0.00000 6.00000 1.00000 7.00000
Sum of Paid Amount $0 $0 $0 $48,819 $30 $48,849
Sum of Total Amount $0 $0 $0 $48,819 $130 $48,949
Sum of Gold/Silver Price $0.00 $0.00 $0.00 $9,770.00 $120.00 $9,890.00
Sum of Stone/Other Price 0.00 0.00 0.00 1.00 0.00 1.00
Record Count 0 0 0 1 1 2
Details (6 Rows)
Click an intersection in the table above to filter details.
Item Id | Customer Name: Customer Name | Ornament | Item Type | Weight | Billing ID | Paid Amount | Total Amount | Gold/Silver Price | Stone/Other Price
1 Item -0008 Akash Pratti Neckless Gold 350.00000 BILL-006 $63,000 $3,856,023 $9,770.00 320.00
2 Item -0001 Dinesh Ketana Ring Gold 6.00000 BILL-001 $48,819 $48,819 $9,770.00 1.00
3 Item -0004 PrattiPati Sneha Earring Silver 1.00000 BILL-004 $30 $130 $120.00 0.00
4 Item -0002 Akash Pratti Neckless Silver 199.00000 BILL-002 $24,814 $24,814 $120.00 18.00
5 Item -0003 Joshika Palametta Gold Pendant Gold 1.71200 BILL-003 $16,613 $16,613 $9,770.00 0.00
6 Item -0005 Likhitha Ketana Gold Neckless Gold 6.00000 BILL-005 $2,354 $11,209 $9,770.00 5.00
7 563.71200 $155,630 $3,957,608 $39,320.00 344.00
Row Counts
Detail Rows
Grand Total
Stacked Summaries


Report: Billings
Billings Report
Total Records 6
Total Gold/Silver Price $39,320.00
Total Paying Amount $5,090
Average Paying Amount $848
Maximum Paying Amount $5,000
Minimum Paying Amount $0
Median Paying Amount $22
Total Paid Amount $155,630
Customer Name
Total Amount →
$130 | $11,209 | $16,613 | $24,814 | $48,819 | $3,856,023 | Total
Akash Pratti Sum of Gold/Silver Price $0.00 $0.00 $0.00 $120.00 $0.00 $9,770.00 $9,890.00
Sum of Paying Amount $0 $0 $0 $0 $0 $5,000 $5,000
Average Paying Amount $0 $0 $0 $0 $0 $5,000 $2,500
Largest Paying Amount $0 $0 $0 $0 $0 $5,000 $5,000
Smallest Paying Amount $0 $0 $0 $0 $0 $5,000 $0
Median Paying Amount $0 $0 $0 $0 $0 $5,000 $2,500
Sum of Paid Amount $0 $0 $0 $24,814 $0 $63,000 $87,814
Record Count 0 0 0 1 0 1 2
Dinesh Ketana Sum of Gold/Silver Price $0.00 $0.00 $0.00 $0.00 $9,770.00 $0.00 $9,770.00
Sum of Paying Amount $0 $0 $0 $0 $33 $0 $33
Average Paying Amount $0 $0 $0 $0 $33 $0 $33
Largest Paying Amount $0 $0 $0 $0 $33 $0 $33
Smallest Paying Amount $0 $0 $0 $0 $33 $0 $33
Details (6 Rows)
Click an intersection in the table above to filter details.
Billing: Billing ID | Ornament | Gold/Silver Price | Paying Amount | Paid Amount
1 BILL-002 Neckless $120.00 $0 $24,814
2 BILL-006 Neckless $9,770.00 $5,000 $63,000
3 BILL-001 Ring $9,770.00 $33 $48,819
4 BILL-003 Gold Pendant $9,770.00 $0 $16,613
5 BILL-005 Gold Neckless $9,770.00 $47 $2,354
6 BILL-004 Earring $120.00 $10 $30
7 $39,320.00 $5,090 $155,630

D. Dashboards

Use Case:

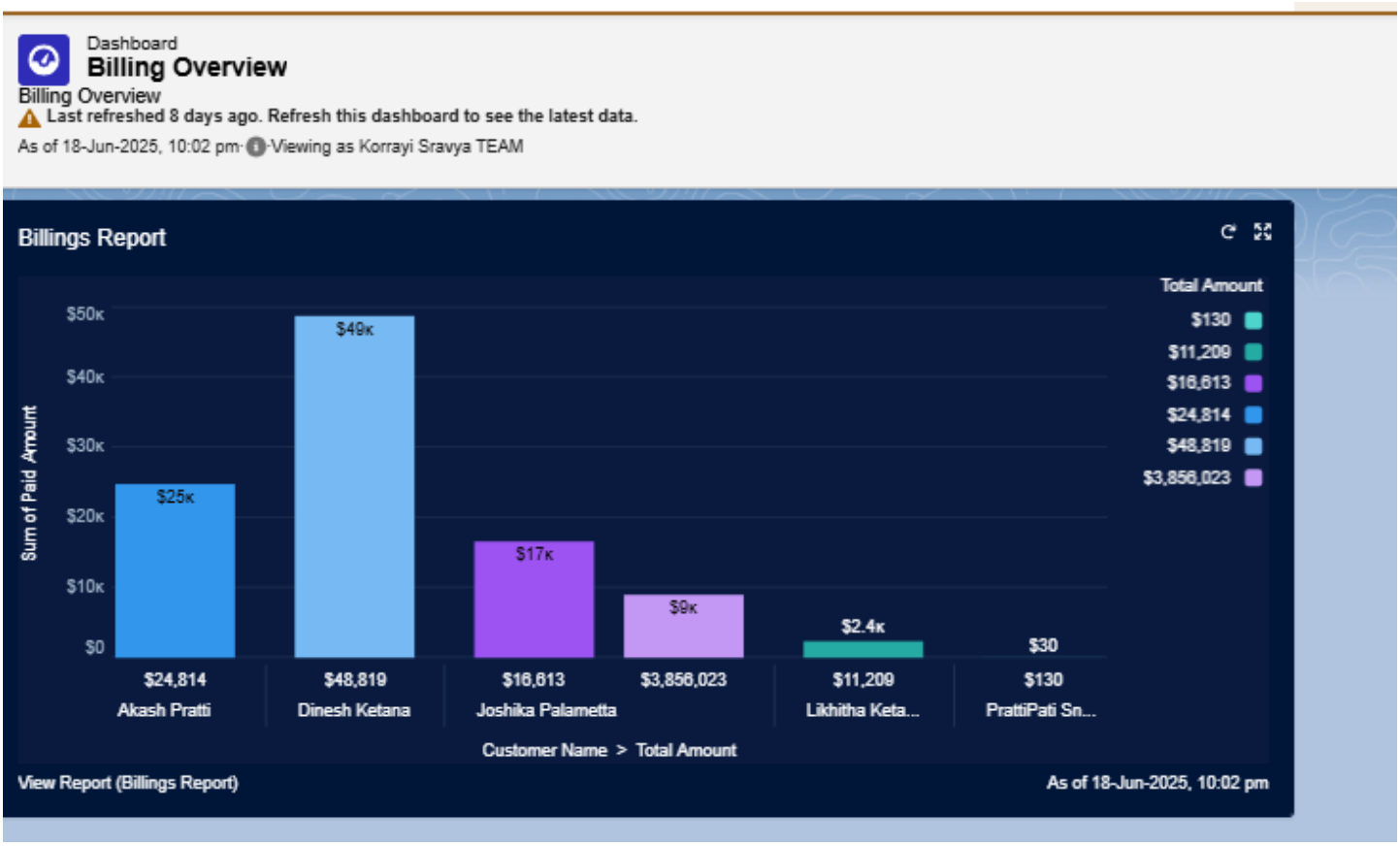
To visualize business KPIs and trends for gold and silver items, customer activity, and financial insights.

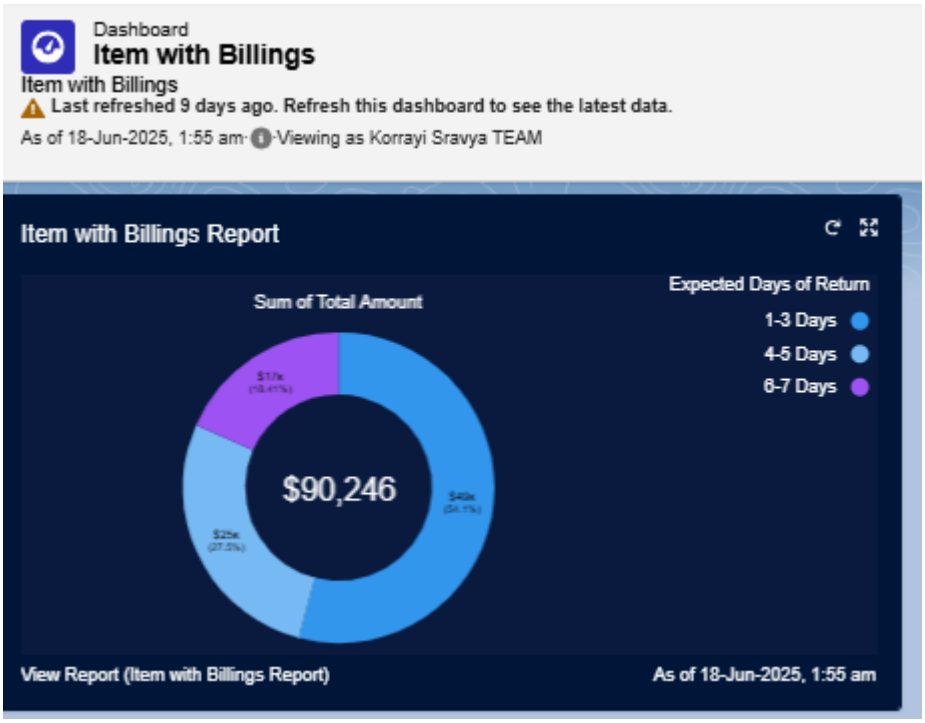
Activity 1: Create Dashboard – "Jewelry Business Overview"

- Added 3 components:
    - Total Billing by Item
    - Orders by Ornament Type
    - Customer Count by Gender

Activity 2: Additional Dashboard – "Customer and Order Insights"

- Included:
    - Orders by Date
    - Total Revenue Chart
    - Item Inventory Breakdown





E: Automation: Flows & Triggers

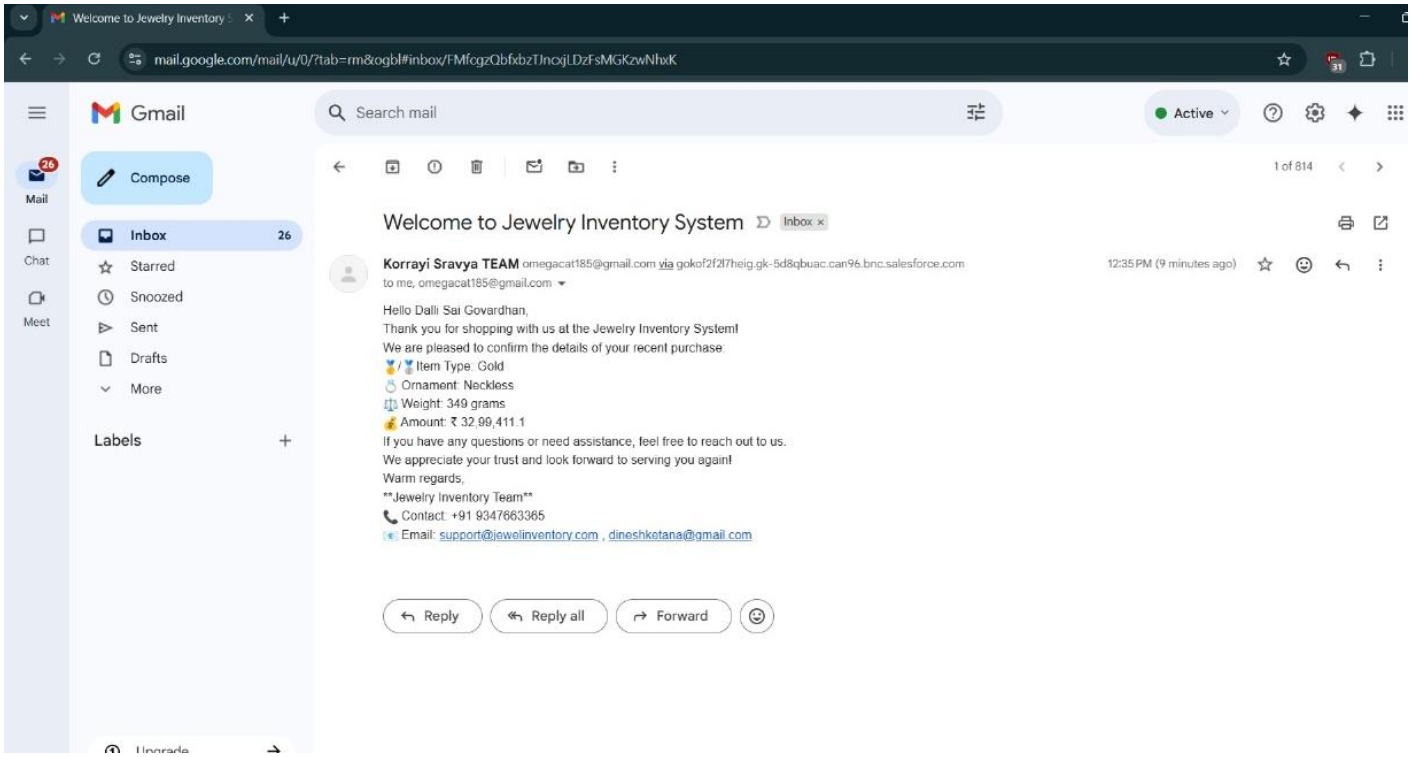
Flow: Record-Triggered Flow to Send Billing Confirmation Email

Use Case:

Automatically send an email upon Billing record creation/update.

Steps:

1. Created a **Record-Triggered Flow** on Billing__c (when created or updated)
2. Used Text Template to craft the email body with merge fields:
    - Customer Name
    - Item Type
    - Ornament
    - Weight
    - Amount
3. Configured email action to send notification using Send Email element
4. Activated flow



F. Automation: Apex Trigger

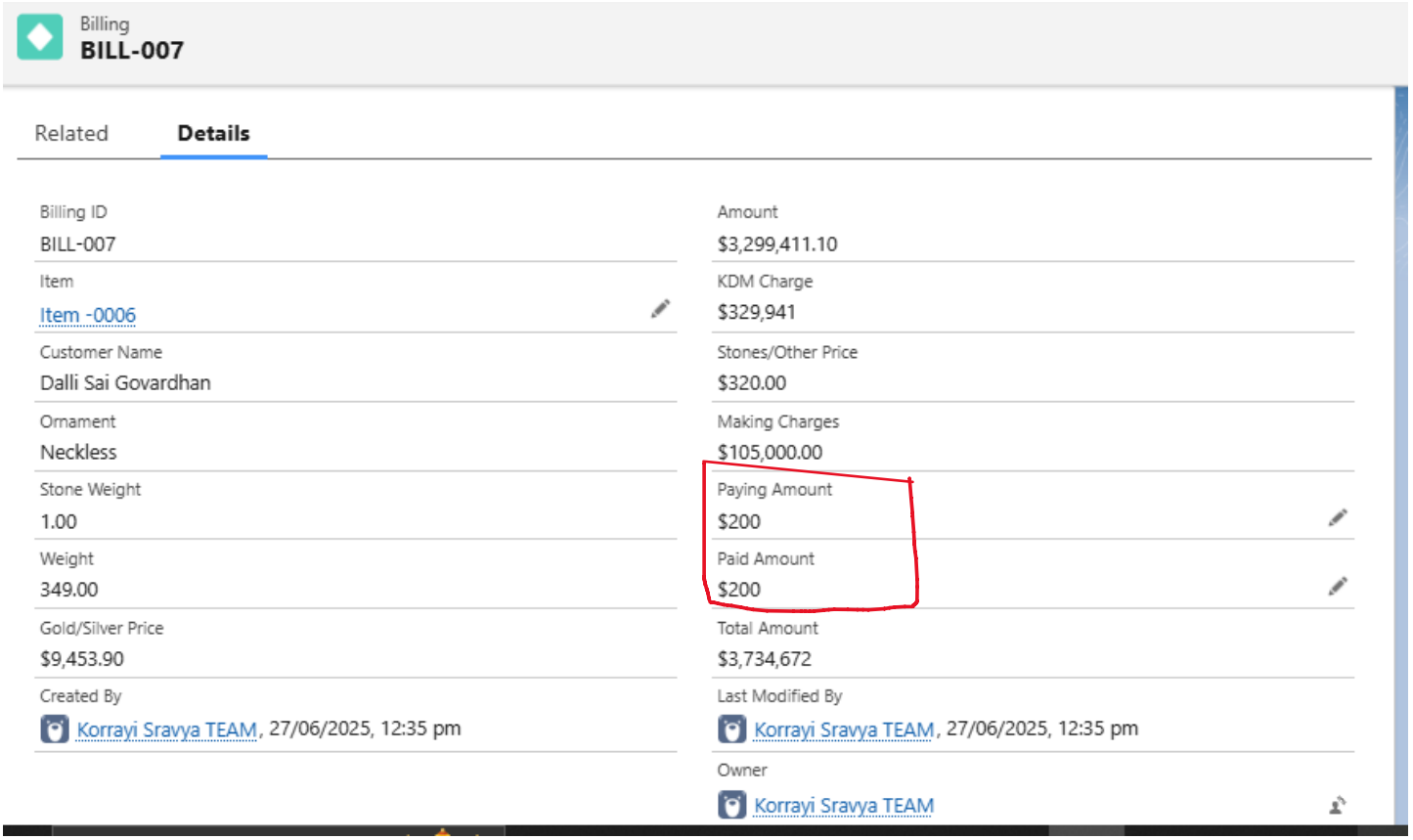
Trigger: Auto-Update Paid Amount

Objective:

Ensure Paid_Amount__c is updated automatically without manual input

Logic:

- Trigger runs on insert/update of Payment record
- Matches corresponding Billing__c record
- Updates Paid_Amount__c = sum of payments received



G. Validation Rules

Rule: Prevent Overpayment

Name: Validate_Paid_Amount

Formula:

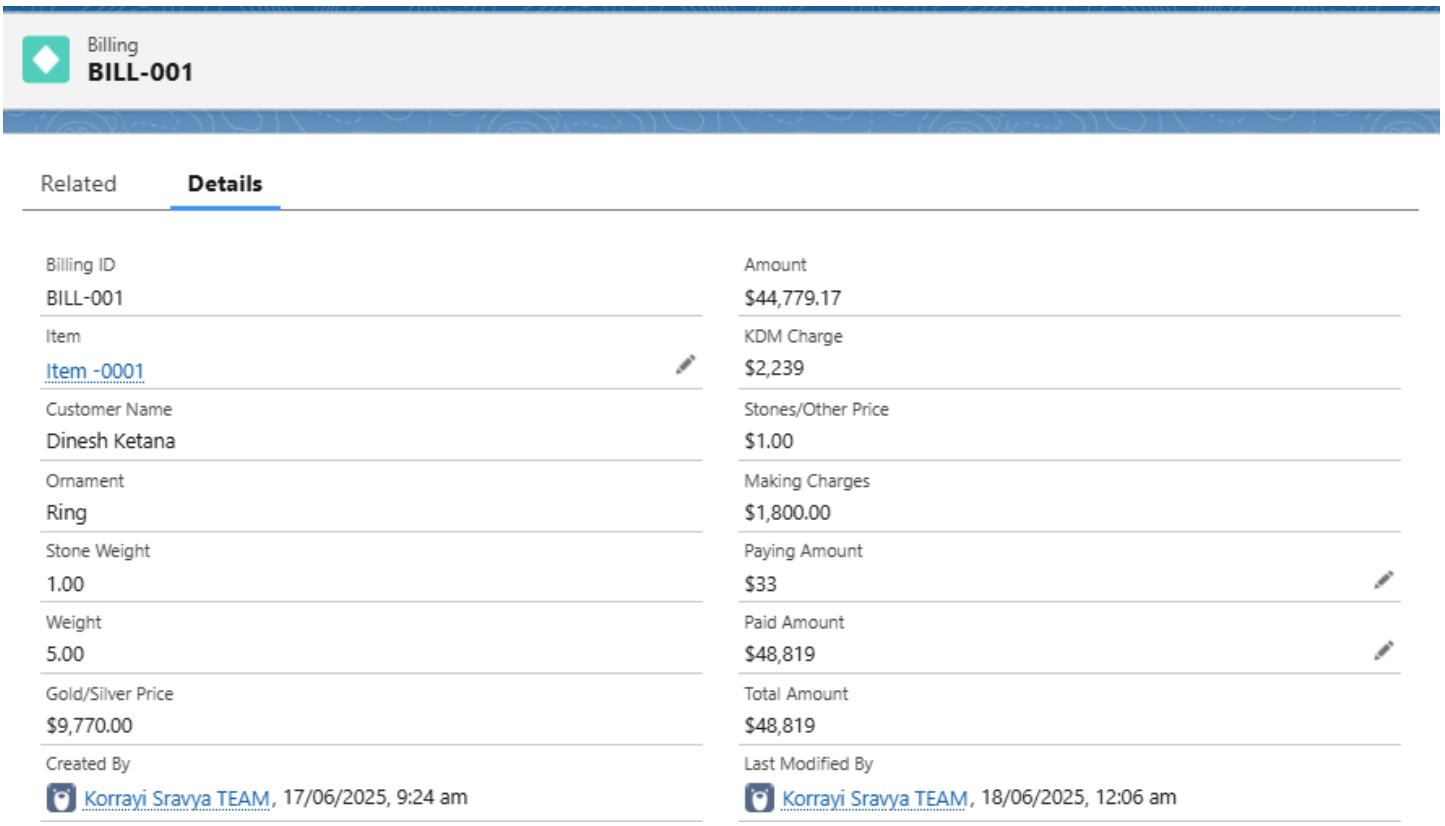
Paid_Amount__c > Total_Amount__c

Error Message:

“Paid amount cannot exceed the total billing amount.”

Purpose:

Maintains financial integrity and prevents accidental overcharges



H. Object Relationship Diagram (ERD)

Purpose

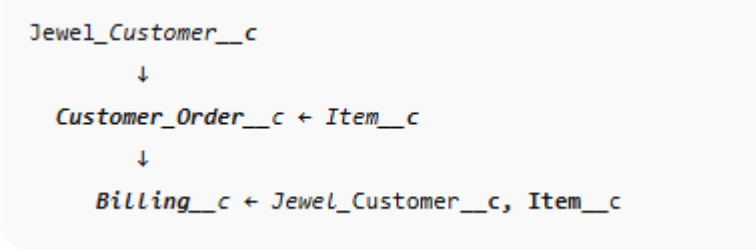
The Entity Relationship Diagram (ERD) is a crucial architectural representation of how objects in the CRM for Jewelry Management system relate to each other. This diagram helps visualize the database schema, illustrating the lookup relationships between various custom objects. It supports better understanding of the data flow, record hierarchy, and how modules like Billing, Orders, Items, and Customers are interlinked**.**

ERD Overview of Custom Objects and Relationships

Parent Object	Child Object	Relationship Type	API Field Name	Description
Jewel_Customer__c	Customer_Order__c	Lookup	Linked_Customer__c	Connects customer data to each order placed.
Item__c	Customer_Order__c	Lookup	Ordered_Item__c	Links the item (gold/silver ornament) to the customer's order.
Jewel_Customer__c	Billing__c	Lookup	Related_Customer__c (example)	Tracks the customer associated with each bill.

Item__c	Billing__c	Lookup	Related_Item__c (example)	Maps the jewelry item to its billing record.
Price__c	(Referenced in) Item__c / Billing__c	Lookup or Formula Reference	Gold_Price__c / Silver_Price__c	Not directly related via lookup, but referenced in formula fields for price use.

Relationship Flow Summary



The relationships can be summarized in this logical order:

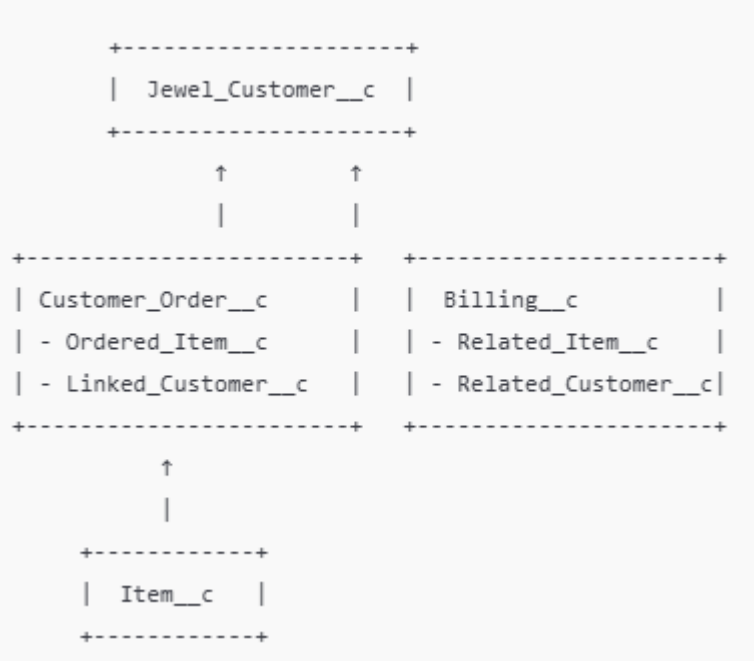
This implies:

- A customer places an order for a specific item
- A billing record is generated for that order, pulling customer and item details
- Price__c is used as a reference for gold/silver pricing in calculations

Visual ERD Illustration (Sample)

While this cannot display an actual image here, you should generate a diagram that looks like this in Draw.io or Lucidchart (example text for your doc):

"The diagram below shows the lookup relationships between core custom objects. Arrows point from child to parent."



Use Case Explanation

Example Scenario:

Dinesh (Jewel_Customer__c) orders a Gold Necklace (Item__c). The system creates a Customer Order (Customer_Order__c) linking both the customer and item. Once the order is processed, a Billing record (Billing__c) is generated with references to the same customer and item. Pricing is calculated using fields and formula referencing values from Price__c (not directly related via lookup)._

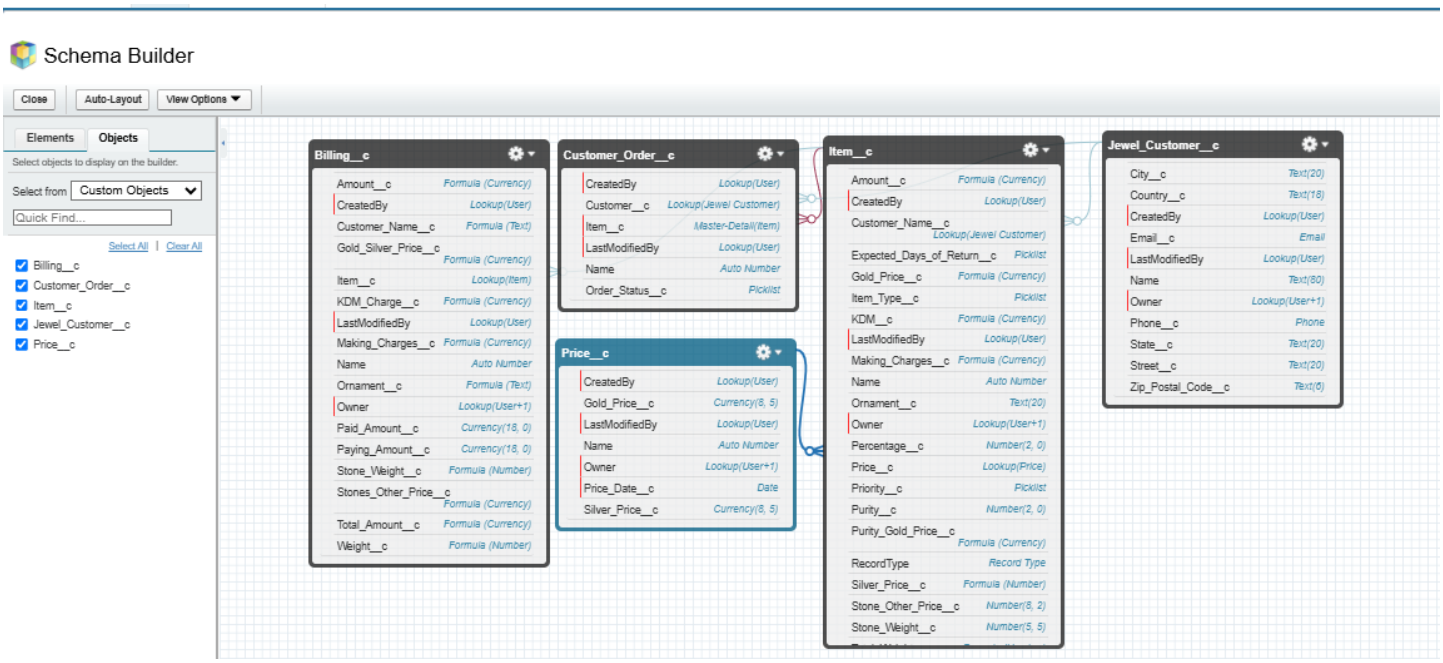
How You Can Create It Practically

Option 1: Schema Builder (Salesforce Native Tool)

1. Go to Setup → Search Schema Builder
2. Drag your objects: Jewel_Customer__c, Customer_Order__c, Item__c, Billing__c, Price__c
3. View the lookup relationships visually
4. Take a screenshot and paste it into your document

Option 2: Draw.io / Lucidchart

1. Go to https://draw.io or https://lucidchart.com
2. Use Entity shapes to create object boxes
3. Use arrows with labels (e.g., "Lookup") to show relationships
4. Export as PNG or JPEG



5. Insert into your Word/PDF documentation under the “ERD” section

I. User Profiles and Permission Management

Overview:

Profiles in Salesforce determine the level of access and control that a user has over various features and data within the organization. They define permissions related to:

- Object-level access
- Field-level access
- Tab and App visibility
- Apex class and Visualforce page access
- Record Types, Page Layouts
- Login hours and IP restrictions

To ensure data security and task-based functionality for different users in the CRM for Jewel Management system, custom profiles were created and configured.

A. Types of Profiles in Salesforce

1. Standard Profiles (Predefined, cannot be deleted):

- System Administrator
- Standard User
- Read Only
- Marketing User

- Contract Manager
- Solutions Manager

These come with default permissions and are typically used for basic or administrative users.

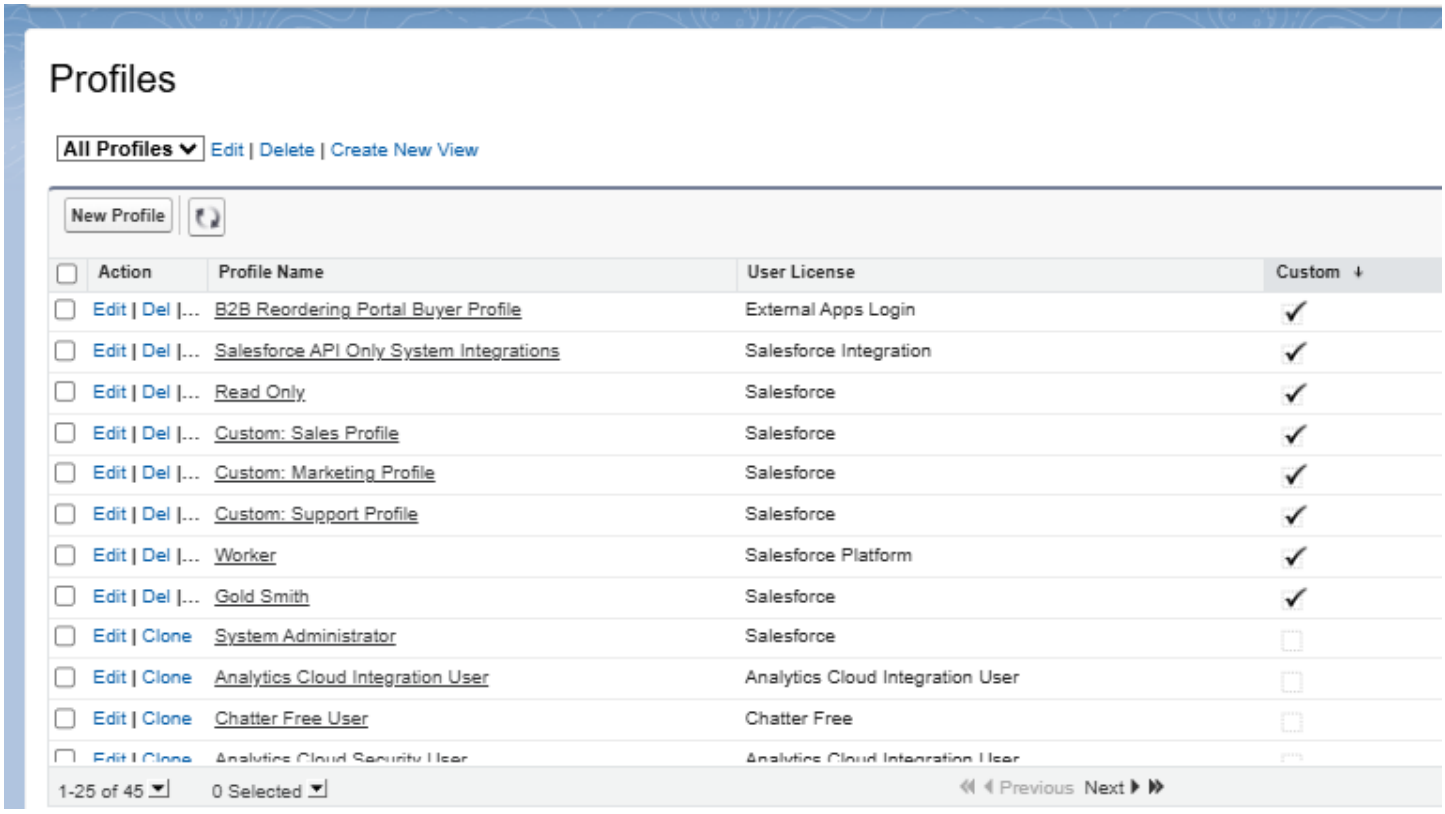
2. Custom Profiles (User-defined, can be deleted if not in use):

- Custom profiles were created to align with organizational roles such as Goldsmith and Worker to provide **role-based access control**.

B. Use Case Scenario: Role-Based Access

The Goldsmith (business owner) requested differentiated access for users based on their **role and responsibilities**:

- **Gold Smith**: Needs full access to manage customers, items, orders, pricing, and billing.
- **Worker**: Needs access to item management and order handling but **should not access sensitive customer or billing data**.



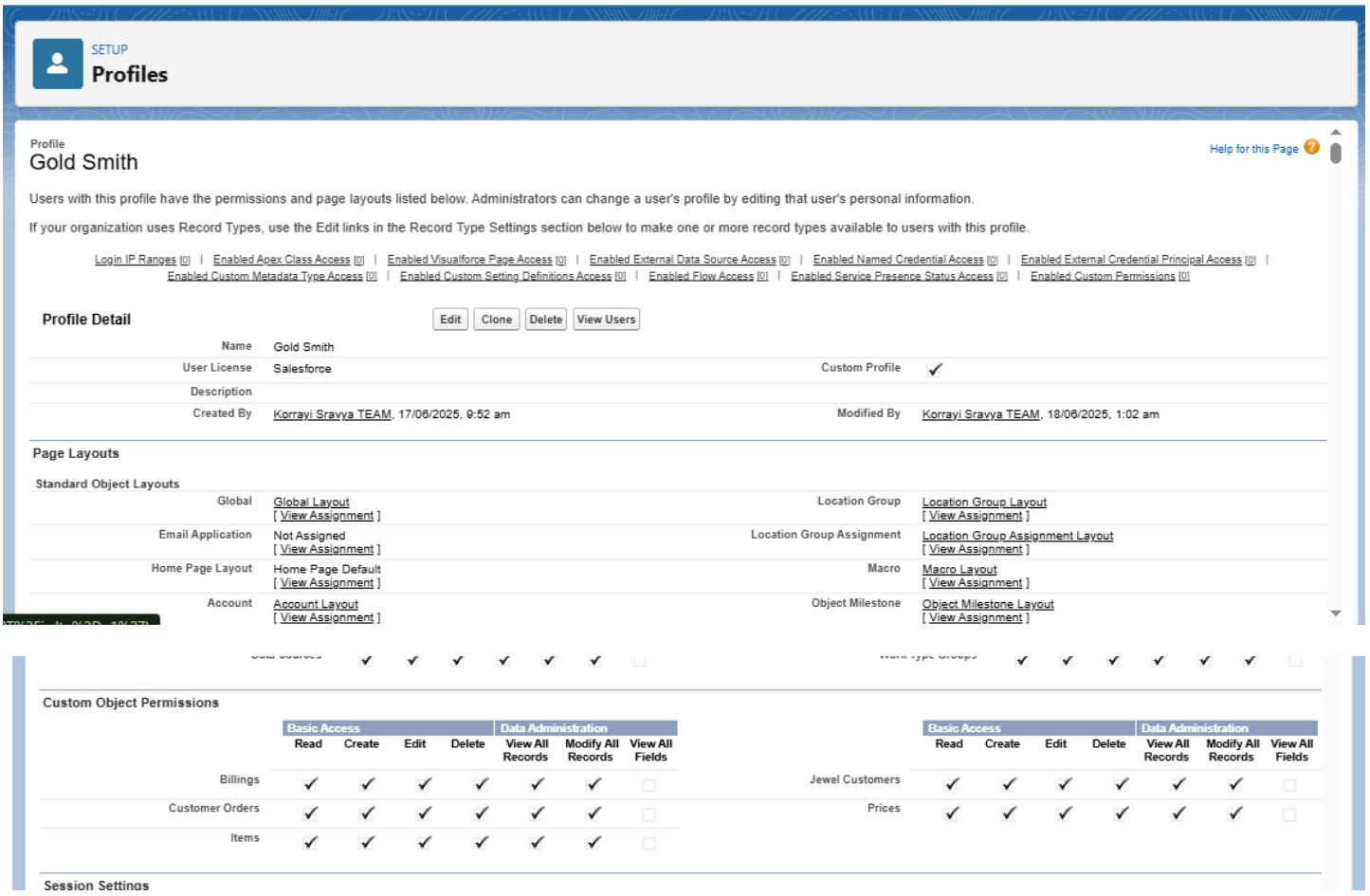
C. Implementation Activities

1. Gold Smith Profile Creation

- **Objective:** To allow full access to all business-critical objects.
- **Steps:**

1. Navigate to **Setup → Profiles**
2. Clone the **System Administrator** profile
3. Name the new profile: Gold Smith
4. Save and then click **Edit**
5. Under **Custom Object Permissions**, provide full CRUD access to:
    - Jewel Customer
    - Item
    - Customer Order
    - Price
    - Billing
6. Click **Save** to confirm permissions

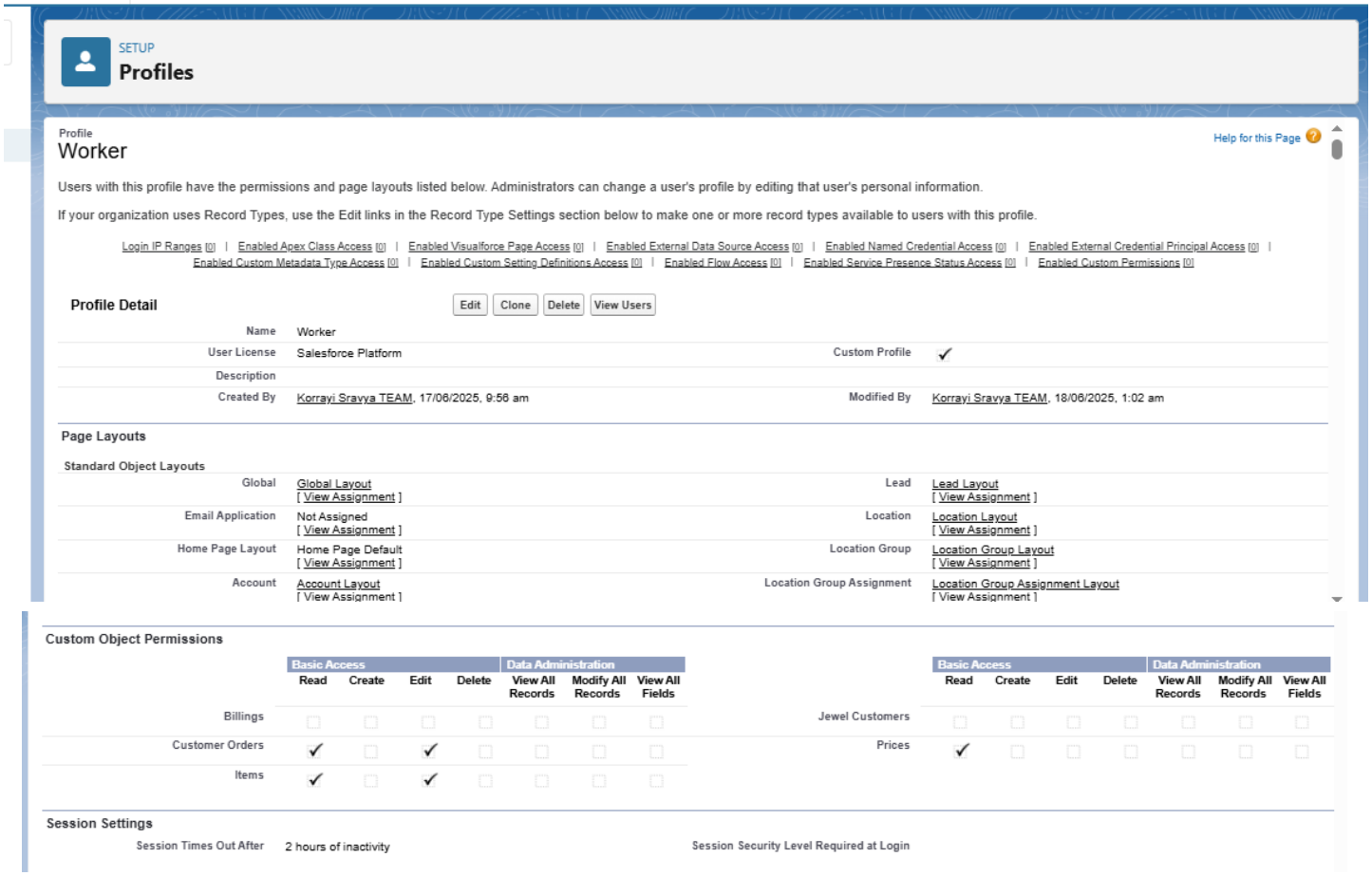
- **Outcome:** The Gold Smith user can create, edit, view, and delete records across all core objects.



2. Worker Profile Creation

- **Objective:** To limit access only to objects necessary for daily operations.

- **Steps:**
    1. Navigate to **Setup → Profiles**
    2. Clone the **Salesforce Platform User** profile
    3. Name the new profile: Worker
    4. Save and then click **Edit**
    5. Under **Custom Object Permissions**, grant access only to:
        - Item
        - Price
        - Customer Order
    6. Deny access to:
        - Jewel Customer
        - Billing
    7. Click **Save**
- **Outcome:** The Worker user can manage inventory and order processing without accessing sensitive financial or customer data.



D. Benefits and Security Measures

- **Data Protection:** Ensures sensitive data like customer information and payment amounts are not accessible by junior staff.
- **Operational Efficiency:** Enables each role to access only the data and tools required for their job function.
- **Scalability:** Profiles can be reused and extended for new roles in the future.

Summary Table – Custom Profiles

Profile Name	Cloned From	Object Access Granted	Notes
Gold Smith	System Administrator	Jewel Customer, Item, Order, Billing, Price	Full Access
Worker	Salesforce Platform User	Item, Price, Customer Order	Restricted – no billing/customer

J. Role Hierarchy and Record-Level Access Control

Overview

In Salesforce, **Roles** are used to control **record-level access** based on a user's position in the organization. Unlike profiles (which control object-level permissions), roles determine **what records a user can *see* and *share*** based on the hierarchy. By assigning users to different roles, administrators can implement **data visibility control** that mimics the company's organizational structure.

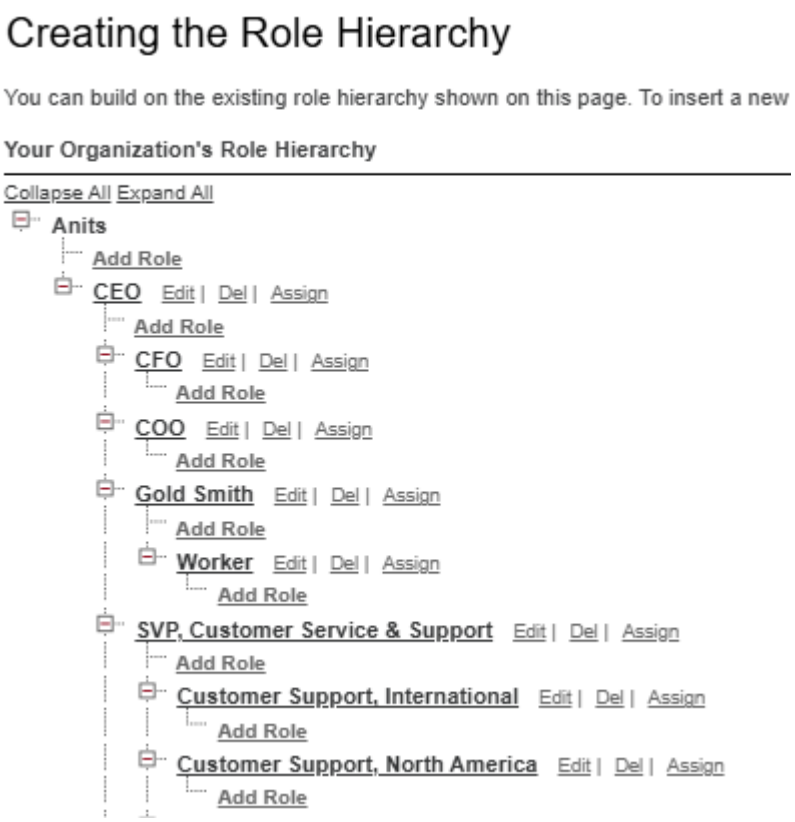
A. Use Case: Role-Based Record Visibility

After successfully setting up profiles to control *permissions and object-level access*, the next step is to configure roles that determine *record visibility* within the organization.

Requirement:
The organization wants to implement a structure where:

- The **Gold Smith (Manager)** has visibility over all records under him.
- The **Worker (Subordinate)** only sees records they own or are explicitly shared with them.

This ensures **secure and hierarchical data access**, improving data control and clarity.



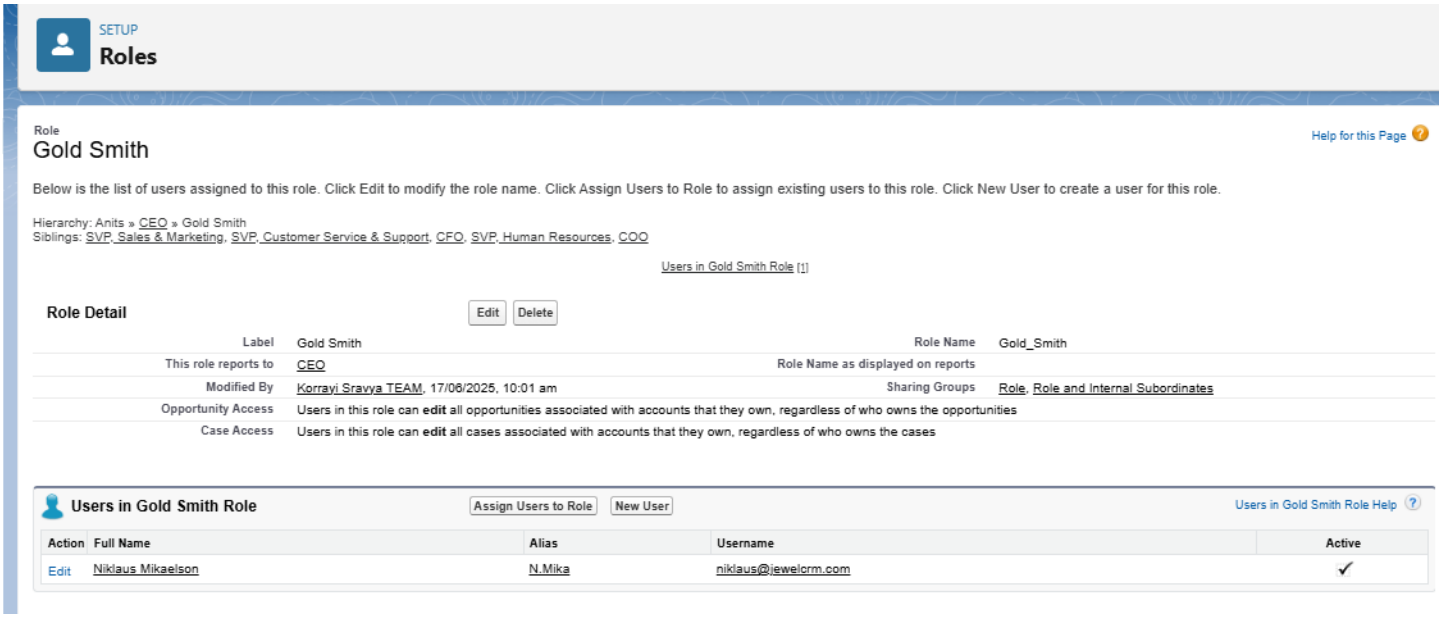
B. Implementation Steps

1. Create Role: Gold Smith

- **Navigation**:
  Go to **Setup → Roles → Set Up Roles**
  Alternatively, use Quick Find and type "Roles".

- **Steps**:

  1. Click on **Expand All** to view the default hierarchy.
  2. Choose the appropriate parent (e.g., CEO or any top-level role).
  3. Click **Add Role** under the selected parent.
  4. Fill the following:
     - **Label**: Gold Smith
     - **Role Name**: (Auto-populated)
     - **This Role Reports To**: Select as per organization (e.g., CEO or another higher role)
  5. Click **Save**

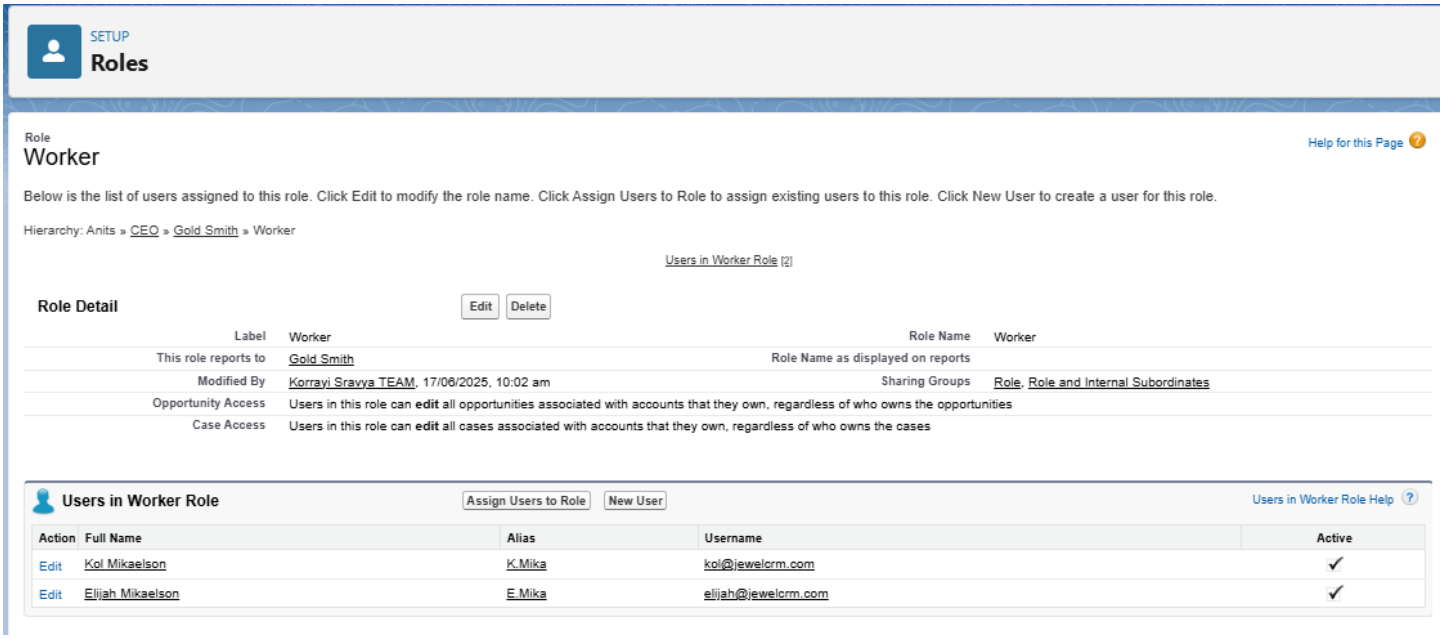


- **Outcome**:
  The Gold Smith role is now part of the hierarchy and can be assigned to users responsible for managing inventory, billing, and customers.

2. Create Role: Worker

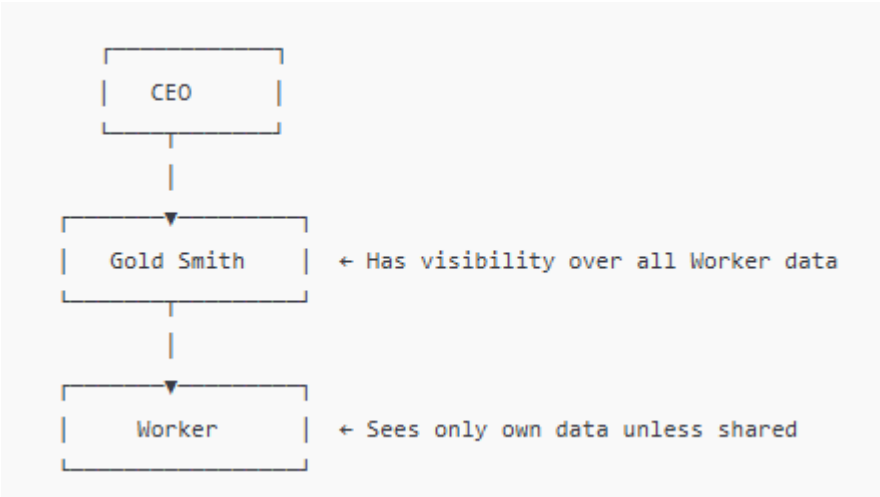
- **Steps**:

  1. Under the **Gold Smith** role, click on **Add Role**.
  2. Enter:
     - **Label**: Worker
     - **Role Name**: (Auto-populated)
     - **This Role Reports To**: Gold Smith
  3. Click **Save**



- **Outcome**:
  Workers now fall directly under the Gold Smith in the role hierarchy and will only have access to records they own unless shared.

C. Visual Example of Role Hierarchy



D. Role Assignment to Users

Once roles are created, assign them to users based on their job role in the org:

1. Go to **Setup → Users → Users**
2. Edit the user record
3. Under **Role**, select either:
   - Gold Smith (for managers/owners)
   - Worker (for general staff)
4. Click **Save**

E. Benefits of Role Hierarchy

Feature	Benefit
Record-level visibility	Users can see only the data they need
Managerial oversight	Higher roles can access subordinate records
Secure sharing	Prevents unauthorized access to sensitive records
Collaboration	Enables efficient data flow between team levels

Role	Reports To	Visibility Scope	Assigned To
Gold Smith	CEO/Admin	Full access to all records under hierarchy	Owner, Admin, Supervisor
Worker	Gold Smith	Access to own records	Staff, Entry-level workers

K. User Creation and Access Configuration

Overview

Users in Salesforce are individuals who have a **login ID and profile** and are granted specific access to objects, records, and features. Each user must be assigned a **profile**, and optionally a **role**, to define **what they can see and do** in the system. In this CRM for Jewel Management project, users are created and configured to reflect the actual organizational hierarchy and job responsibilities.

A. Use Case

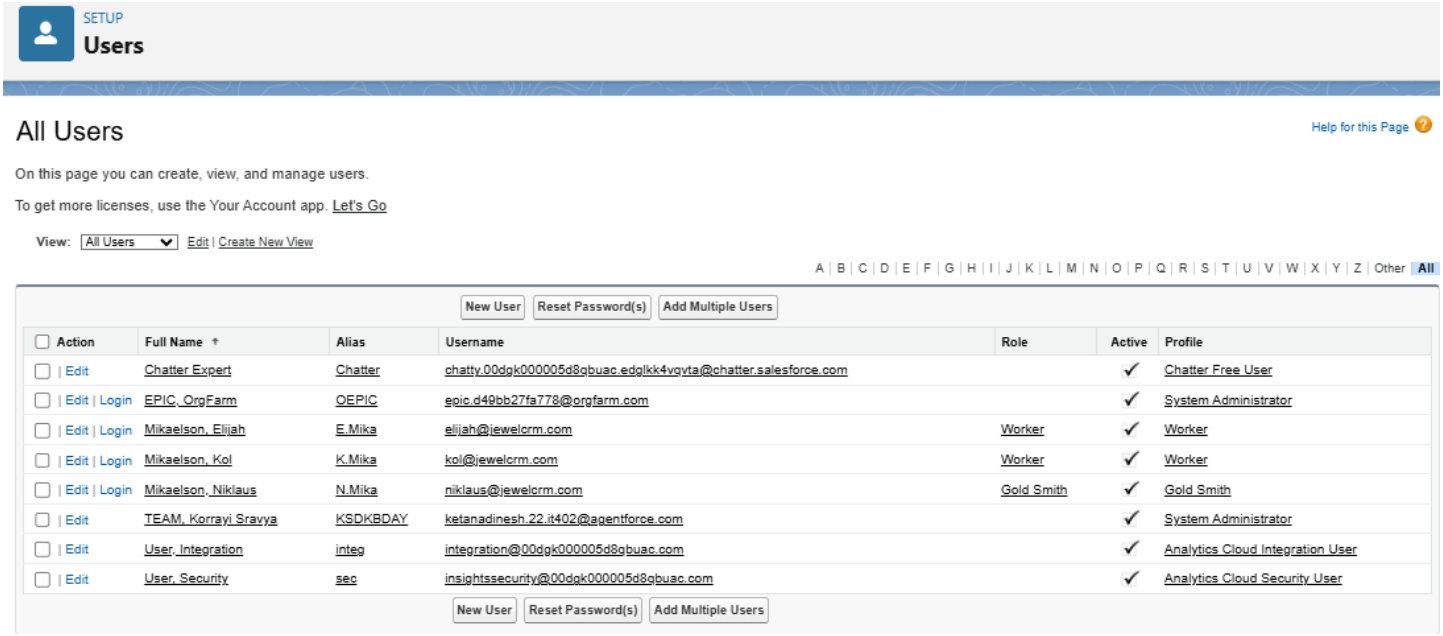
The Goldsmith requested that users be created for different positions such as managers and workers. These users must be provided with **appropriate access** (via profiles and roles) to perform their job functions, such as managing orders, inventory, and billing.

B. Key User Account Attributes

Each user account in Salesforce includes the following mandatory and configurable details:

Field	Description
First Name	User's first name (optional)
Last Name	User's last name (required)
Alias	Shortened identifier for the user
Email	Email address used for notifications and login

Username	Must be unique across all Salesforce orgs
Nickname	Displayed in the app UI (for communities, etc.)
User License	Determines available features and permissions
Profile	Determines object-level and field-level access
Role	Controls record visibility within the org



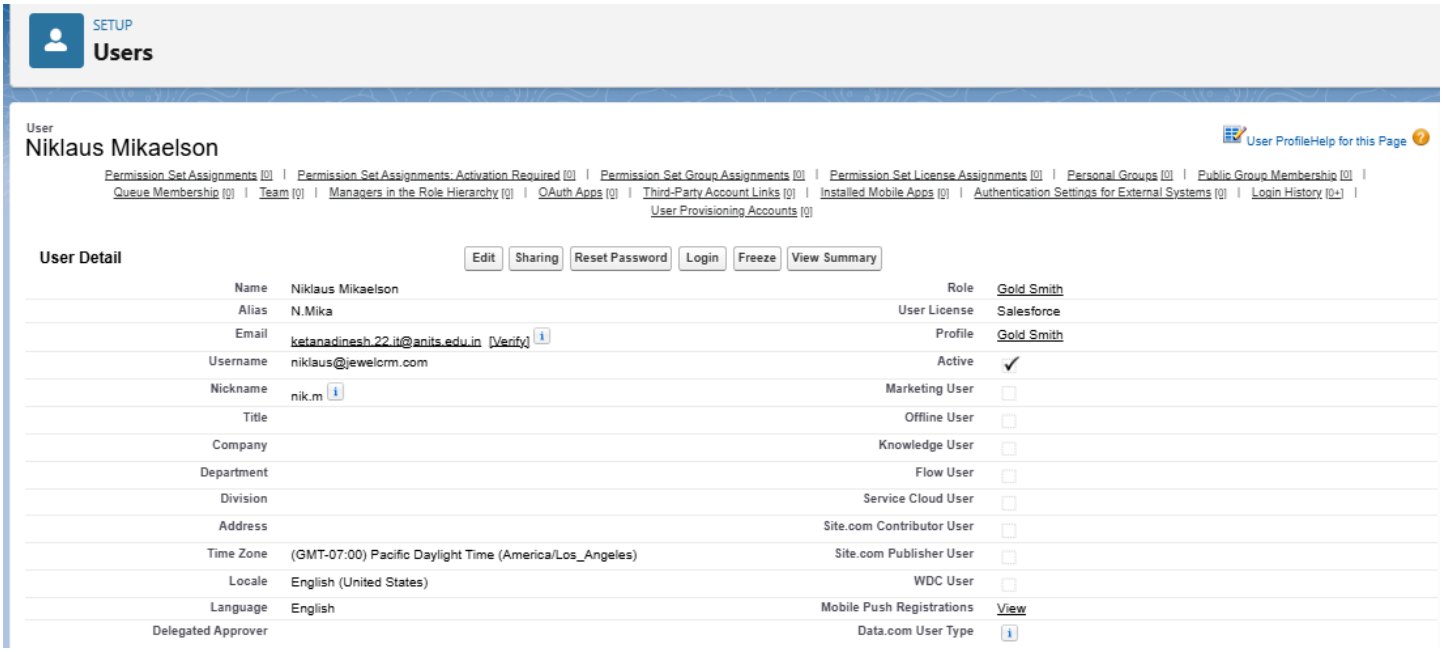
C. User Creation Activities

Activity 1: Create User – Gold Smith

- **Navigation**:
  Setup → Users → Click **New User**
- **Input Details**:

Field	Value
First Name	Niklaus
Last Name	Mikaelson
Alias	n.mikaelson
Email	niklaus@email.com
Username	niklaus@email.com
Nickname	nick

User License	Salesforce
Profile	Gold Smith
Role	Gold Smith



- **Click**: Save

Activity 2: Create User – Worker

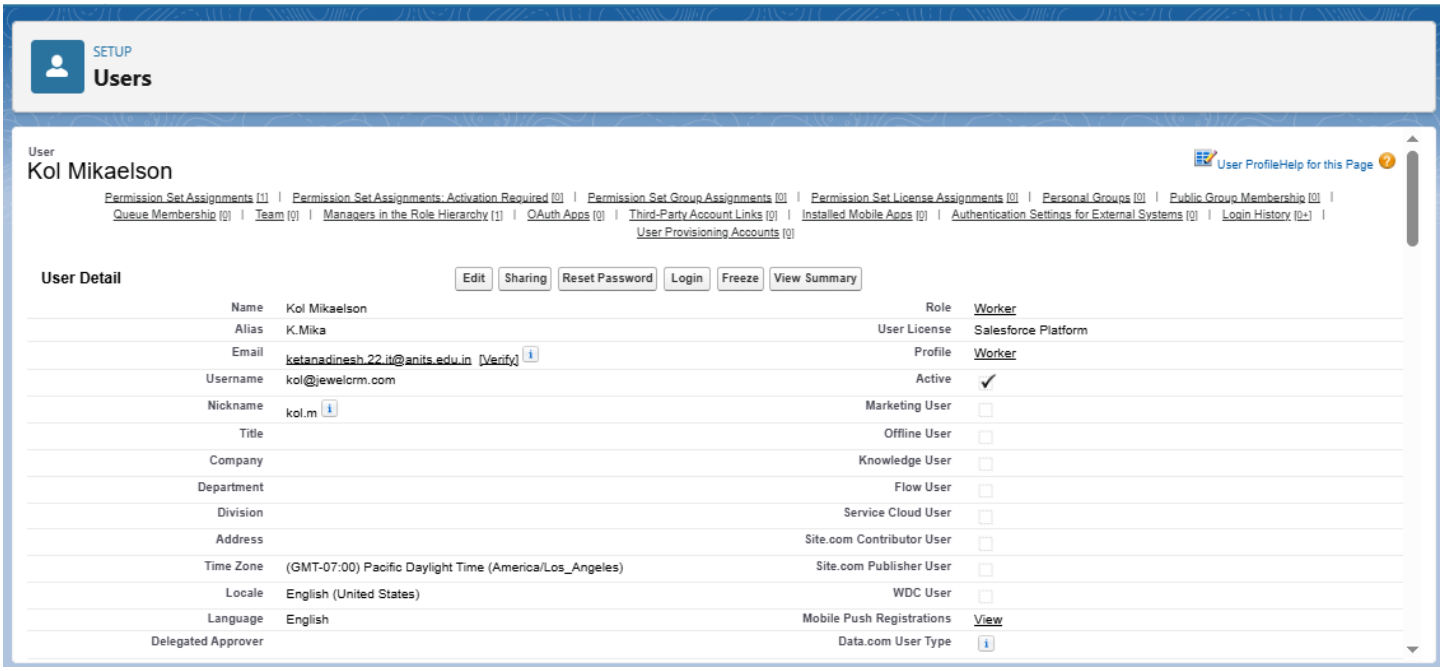
- **Navigation**:
  Setup → Users → Click **New User**

- **Input Details**:

Field	Value
First Name	Kol
Last Name	Mikaelson
Alias	k.mikaelson
Email	kol@email.com
Username	kol@email.com
Nickname	kolm
User License	Salesforce Platform

Profile	Worker
Role	Worker

- **Click**: Save



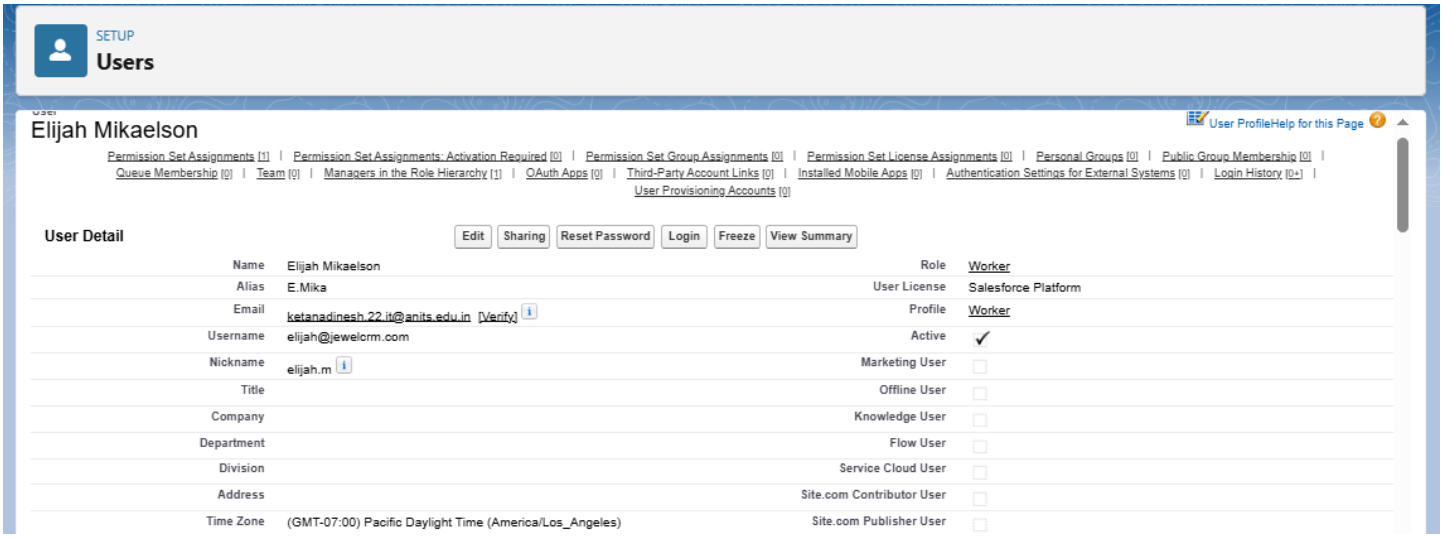
D. Activity 3: Create Additional Users

Following the same steps as above, create **two more users** using the **Worker** profile. Ensure:

- Different unique usernames (e.g., rebekah@email.com, elijah@email.com)
- Email IDs may be placeholders or real for demo purposes
- Assign both the **Worker profile** and **Worker role**

E. Summary of Users Created

User Name	Role	Profile	License
Niklaus Mikaelson	Gold Smith	Gold Smith	Salesforce
Kol Mikaelson	Worker	Worker	Salesforce Platform
Rebekah Mikaelson	Worker	Worker	Salesforce Platform
Elijah Mikaelson	Worker	Worker	Salesforce Platform



Benefits of User Management in Salesforce

- Enables **secure access control** tailored to user responsibilities.
- Ensures **data segregation** between managerial and operational roles.
- Supports **auditing, tracking, and accountability** for transactions.
- Simplifies **user-based automation**, email alerts, and workflow targeting.

L. Record Types Implementation

Overview

Record Types in Salesforce allow administrators to display **different page layouts**, **picklist values**, and **field-level requirements** for the same object. This feature helps organizations **customize data entry forms** based on user roles, record category (e.g., Gold vs Silver), or business process variations—all while using the same object model.

In the CRM for Jewel Management project, **Record Types** were created for **Gold** and **Silver** jewelry items to simplify the user experience, ensuring that users only see and input information relevant to the specific ornament type.

Use Case

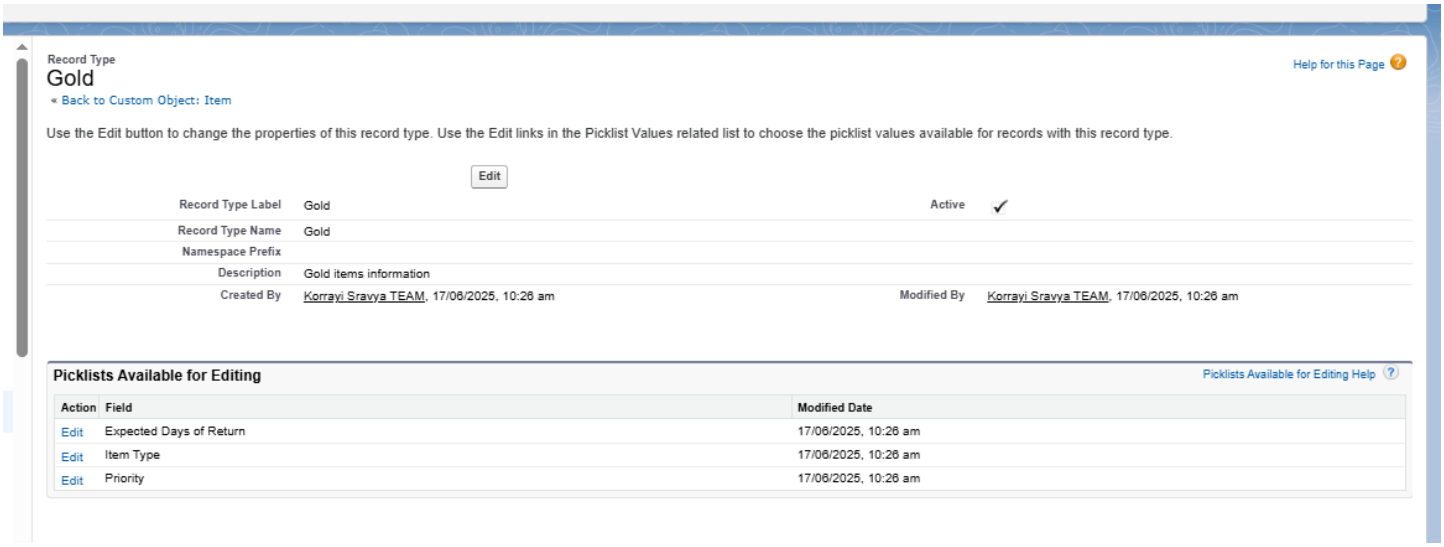
The organization required **customized forms** for different types of items (Gold vs Silver). The GoldSmith requested that data entry should be simplified by showing only the relevant fields when creating or editing records. To solve this:

- A **Gold Record Type** was created using a layout optimized for gold-specific fields.
- A **Silver Record Type** was created using a layout optimized for silver-specific fields.
  This helps streamline data entry and avoid confusion.

Activity 1: Create Record Type – Gold

Steps:

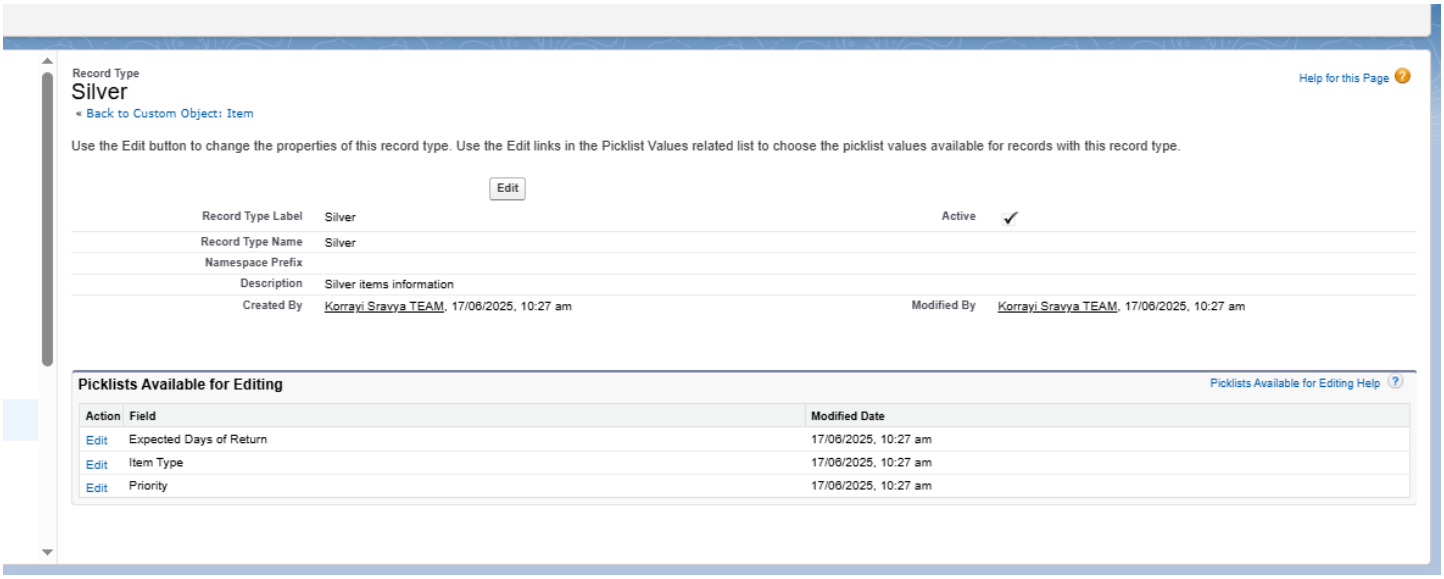
1. **Navigate to Setup** → Object Manager → Search for **Item**
2. Click on **Record Types** under the Item object.
3. Click **New** to create a new Record Type.
4. Fill the following details:
    - **Existing Record Type**: Master
    - **Record Type Label**: Gold
    - **Record Type Name**: Auto-populated as Gold
    - **Description**: Gold items information
5. **Uncheck** "Make Available" for all profiles (to customize visibility).
6. **Check Access** for the following profiles only:
    - Gold Smith
    - Worker
    - System Administrator
7. Click **Next**
8. Choose: "**Apply a different layout for each profile**"
9. Assign the following layouts:
    - **Gold Smith** → Page Layout for Gold
    - **Worker** → Page Layout for Gold
    - **System Administrator** → Page Layout for Gold
10. Click **Save & New**

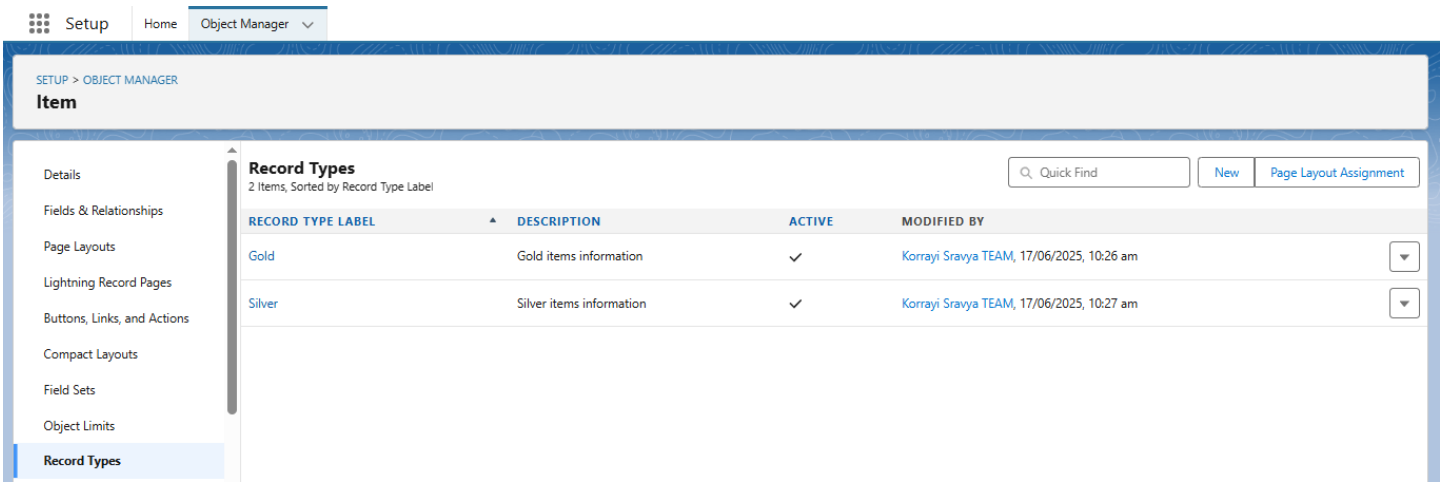


Activity 2: Create Record Type – Silver

Repeat the steps from Activity 1 with the following changes:

- **Record Type Label**: Silver
- **Description**: Silver items information
- Assign layout as **Page Layout for Silver** for all selected profiles (Gold Smith, Worker, System Administrator).





Benefits of Using Record Types

- Customizes the **user interface** based on item category (Gold/Silver)
- Reduces **data entry errors** by showing only relevant fields
- Simplifies **training and onboarding** for workers and staff
- Improves **data accuracy and reporting** by segmenting records properly

M. Permission Sets Configuration

Overview

Permission Sets in Salesforce provide a way to grant users access to specific objects, fields, or functionality **without altering their profile**. Unlike profiles (which are limited to one per user), **multiple permission sets can be assigned** to users, allowing more flexible and scalable access management.

In the **CRM for Jewel Management** system, we used permission sets to **grant additional access rights to users with the Worker profile**. This ensures they can handle Gold and Silver item records while maintaining overall profile-based access restrictions.

Use Case

While users with the **Worker profile** had basic access, they also needed permissions to:

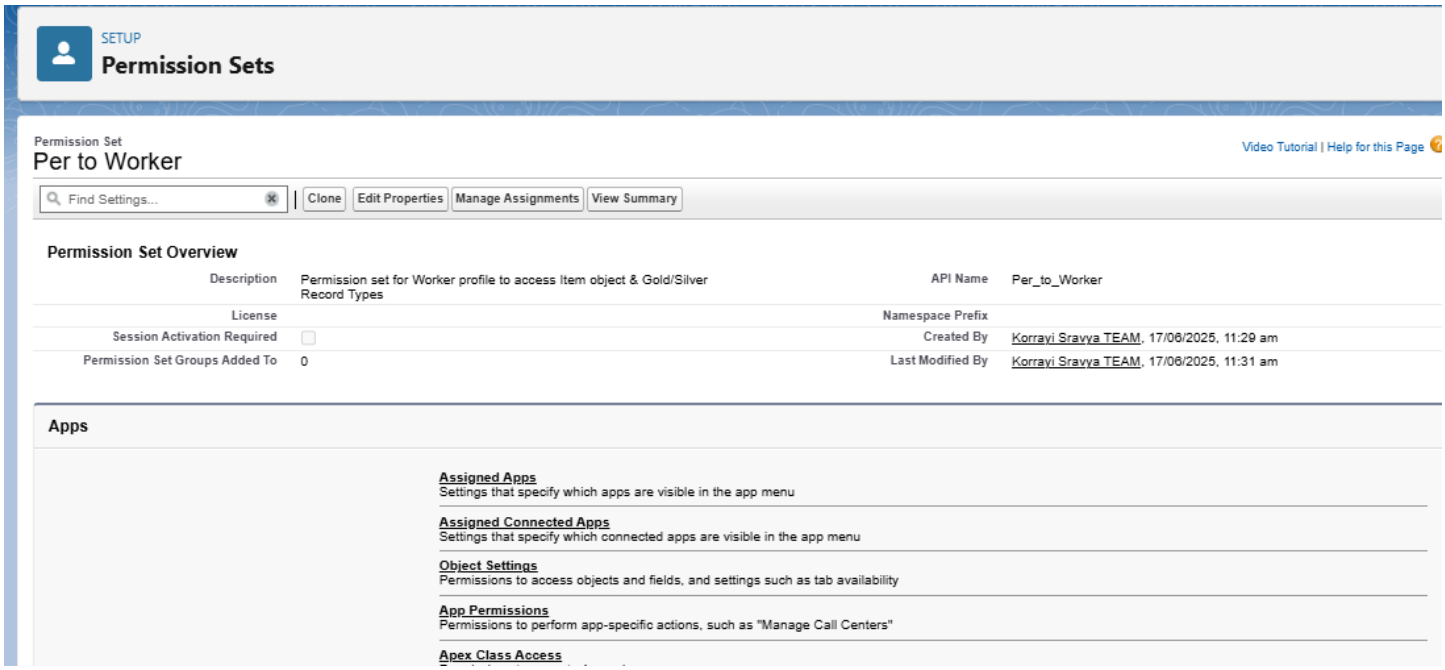
- View and work on **Gold** and **Silver** item records
- Create and update inventory items with appropriate record types

Instead of modifying the Worker profile, we created a **Permission Set** to grant this functionality without disrupting role-based access structure.

Activity: Create and Assign Permission Set to Worker Users

Step 1: Create Permission Set

1. Go to **Setup** → search **Permission Sets** in Quick Find box → click **New**.
2. Enter the following:
    - **Label**: Per to Worker
    - **API Name**: (auto-filled)
3. Click **Save**

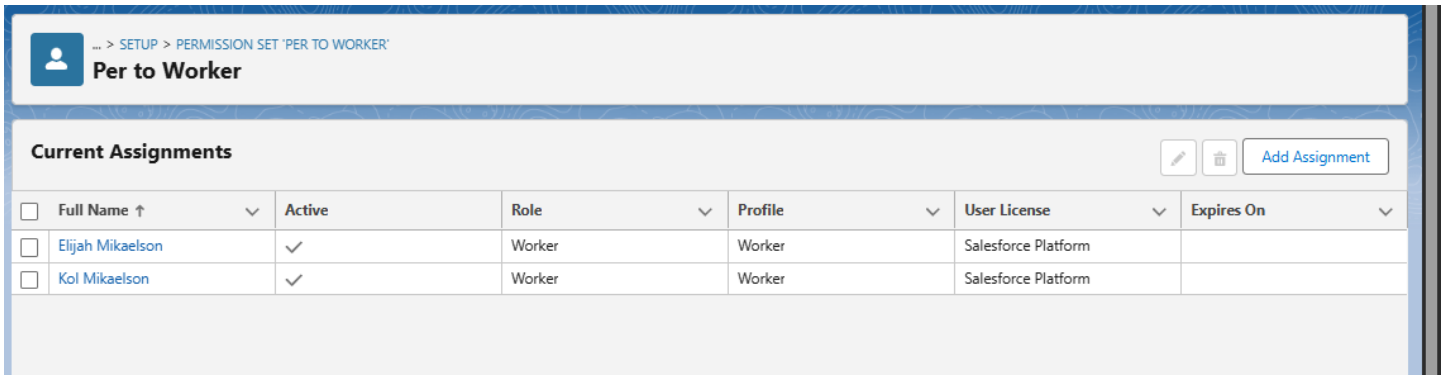


Step 2: Configure Object Permissions

1. On the Permission Set detail page, scroll down to **Apps** → click **Object Settings**.
2. Click on **Item** → then click **Edit**.
3. Under **Item: Record Type Assignments**, check both:
    - Gold
    - Silver
4. Under **Object Permissions**, enable:
    - **Read**
    - **Edit**
    - **Create**
5. Click **Save**

Step 3: Assign Permission Set to Users

1. Back on the Permission Set page, click **Manage Assignments**.
2. Click **Add Assignment**
3. From the user list, select users who are assigned the **Worker** profile (e.g., *Kol Mikaelson*, *any other Worker role users*)
4. Click **Next** → **Assign** → **Done**



Benefits of Using Permission Sets

- Extends access without compromising the **principle of least privilege**
- Ensures **Worker users** can manage both Gold and Silver items as needed
- Allows **granular control** over permissions without cloning or editing profiles
- **Future-proof**: Easily assign the same access to new users without editing the base profile

N. Lightning App Setup

Overview

A **Lightning App** in Salesforce provides a consolidated, branded workspace that allows users to access related tabs, objects, and features efficiently. For the *CRM for Jewel Management* project, a custom Lightning App was created to organize navigation and improve user productivity. It helps different users (e.g., Gold Smith, Worker) seamlessly interact with custom objects like Jewel Customer, Item, Billing, etc.

Use Case

After building a complete data model for jewelry management (objects, relationships, automation), it is essential that users can navigate the system easily. This Lightning App ensures that users can access only the relevant tabs they need to perform their roles effectively within the CRM system.

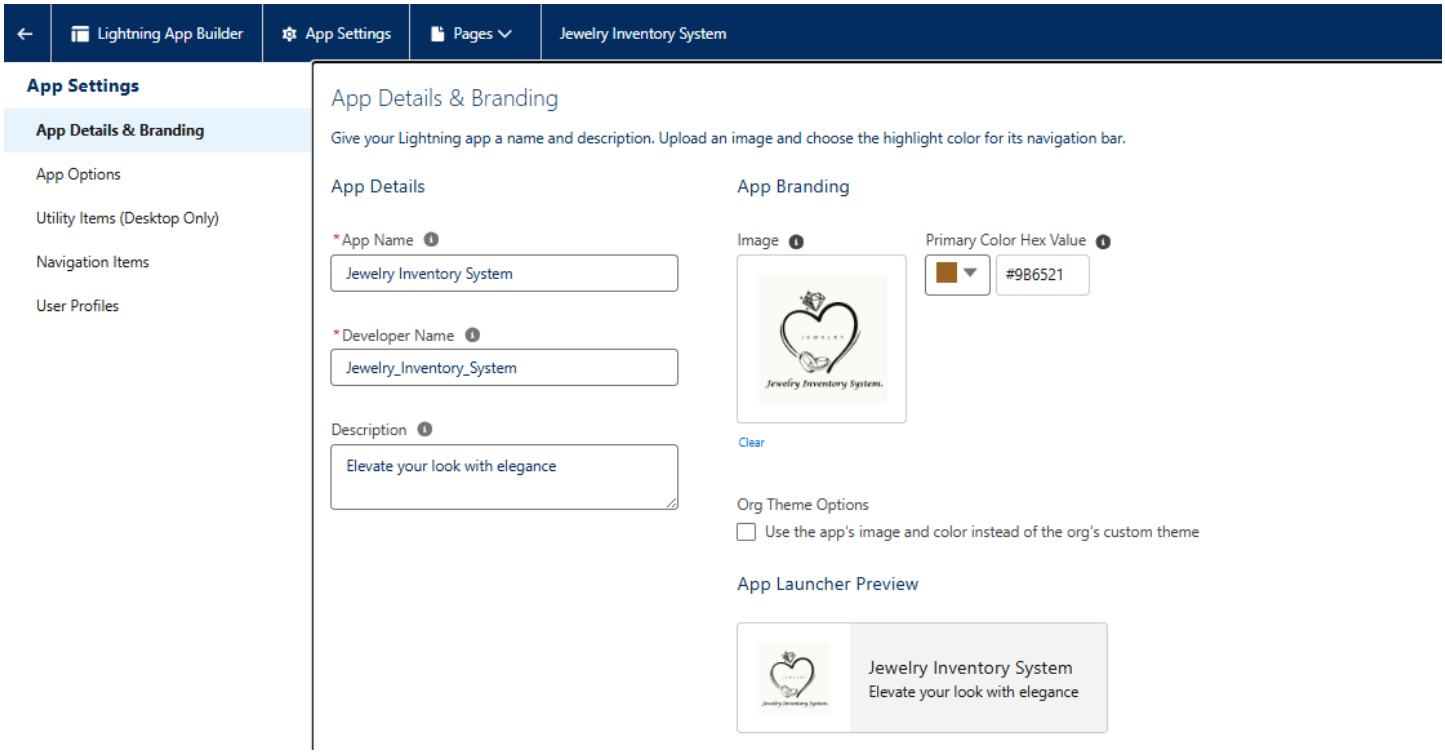
Steps to Create the Lightning App

Step 1: Access the App Manager

- Navigate to:
  Setup → Quick Find → App Manager
- Click **New Lightning App**

Step 2: App Details & Branding

- **App Name:** Jewelry Inventory System
- **Developer Name:** Auto-populated
- **Description:** Elevate your look with elegance

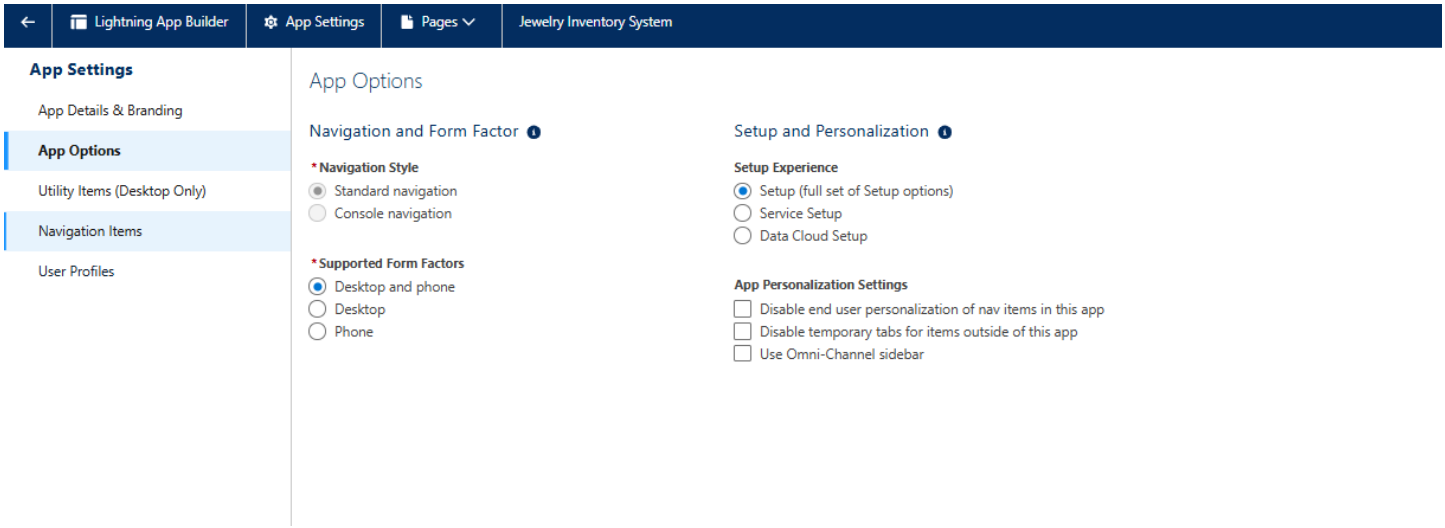


- **Logo:** *(Optional)* Upload if branding image is available
- **Primary Color Hex Value:** Keep default or choose a brand color (e.g., #006DCC)

Click **Next**

Step 3: App Options

- **Navigation Style:** Console Navigation
    - Rationale: Allows multi-tabbed workspace for efficient multitasking
    - Click **Next**



Step 4: Utility Items

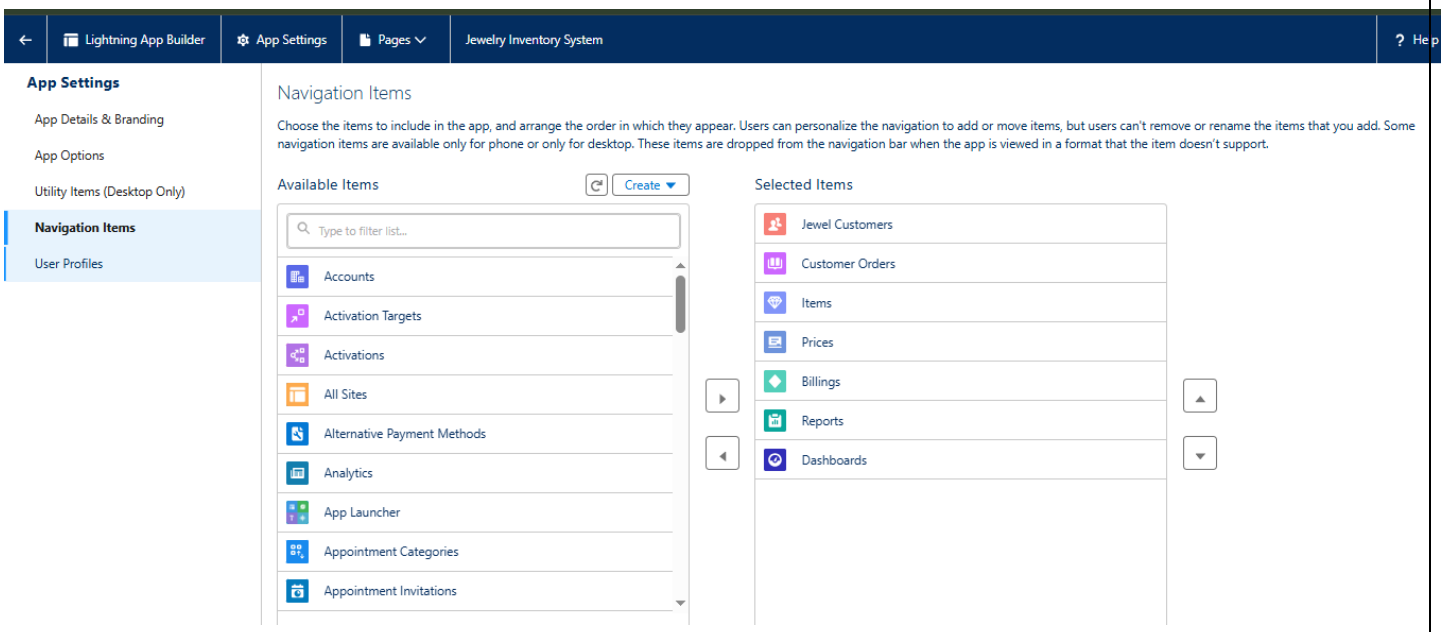
- Leave utility items as default *(e.g., Recent Items, History)*

Click **Next**

Step 5: Navigation Items

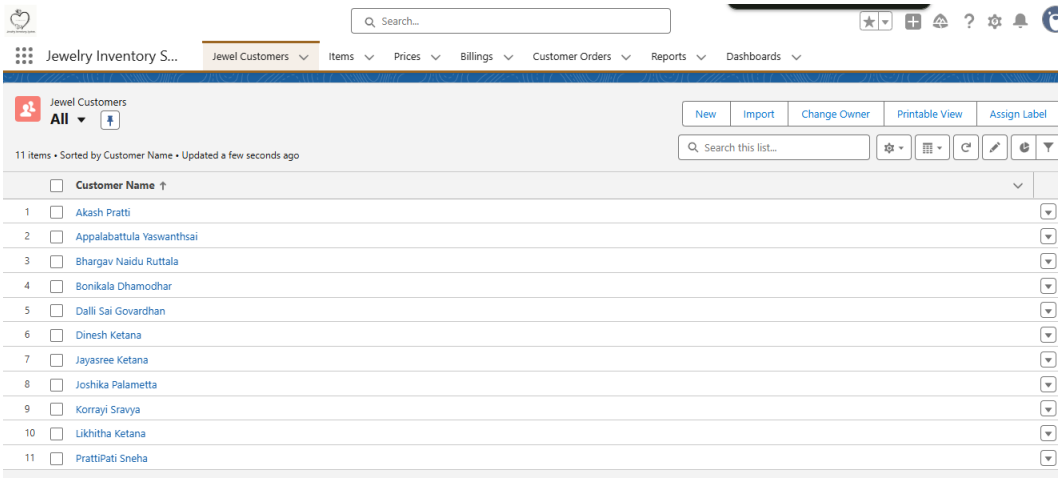
- Use the left-hand **Available Items** list to search and add:
    - Jewel Customer
    - Item
    - Customer Order
    - Price
    - Billing
    - Reports
    - Dashboards

- Use the right arrow ➞ to move selected items into the **Selected Items** list
  (This defines the tabs visible to users in the app)
- Click **Next**



Step 6: Assign User Profiles

- In the **Profile Assignment** step:
  - Search and select profiles like:
    - System Administrator
    - Gold Smith
    - Worker
- Use the right arrow ➞ to add selected profiles
- Click **Save & Finish**



7. FUNCTIONAL AND PERFORMANCE TESTING

7.1 Performance Testing

The CRM Application for Jewelry Management underwent **comprehensive testing** to ensure its functionality, reliability, and performance. We executed multiple positive and negative test cases to validate each feature and ensure seamless business process execution across all custom-built components.

A. FUNCTIONAL TESTING

We followed a **black-box testing approach** to verify that the system meets all user requirements without testing the internal code logic. Each object, flow, validation rule, trigger, and automation was verified through the UI and test records.

1. Positive Test Cases (Expected behavior)

Module	Test Case Description	Expected Outcome	Result
Jewel Customer	Create a customer with valid data	Record saved successfully	Pass
Item	Create Gold Item with valid pricing	Gold layout displayed; record saved	Pass
Record Types	Create item using Silver record type	Silver layout used; relevant fields only	Pass
Customer Order	Create order linked to valid customer and item	Order saved; relationships valid	Pass
Billing	Generate bill linked to existing order and item	Total amount calculated correctly	Pass
Flow	Trigger flow on Billing update	Confirmation email sent successfully	Pass
Trigger	Insert Payment → Billing Paid Amount auto-updated	Paid_Amount__c updated	Pass
Reports	Run “Item with Billing” report	Displays correct totals and relationships	Pass

2. Negative Test Cases (To prevent unwanted behavior)

Module	Test Case Description	Expected Outcome	Result
Billing	Enter Paid Amount > Total Amount	Validation error triggered	Pass
Order	Leave required lookup fields blank	Save blocked with error	Pass
Flow	Missing email in Jewel_Customer	Flow skips email action	Pass

Role Access	Worker trying to access unrelated objects	Access denied	Pass
Email Action	Malformed email address in Jewel_Customer	Email not delivered; logged error	Pass

B. TRIGGER AND FLOW VALIDATION

1. Apex Trigger – Paid_Amount Auto Update

- **Trigger Name**: UpdatePaidAmountTrigger
- **Test Class Coverage**: 100%
- **Validation**:
    - Trigger executed correctly on Billing insert/update
    - Paid_Amount__c auto-calculated as per Payment_Received__c
    - Null values were handled without error
    - Repeated update tested – no duplication or overwriting
- *Trigger performance verified successfully*

2. Record-Triggered Flow – Billing Confirmation Email

- **Flow Name**: Billing_Confirmation_Email_Flow
- **Entry Criteria**: On Create or Update of Billing record
- **Test Details**:
    - Flow triggered on both create and update
    - Email sent only if customer email is valid and billing status = Paid/Generated
    - Used dynamic merge fields in email body (Customer Name, Item, Paid Amount)
    - Debug Logs confirmed execution path through Get Records, Decision, Action
    - Email logs matched with timestamps
- *Flow verified for correct branching and action execution*

C. EMAIL DELIVERY VERIFICATION

- **Tool Used**: Salesforce Email Logs, Debug Logs
- **Steps**:
    - Created billing record → verified Send Email action execution
    - Checked delivery to external email inbox (Gmail/Yahoo)
    - Ensured correct subject line, customer name, and amount merge fields

- Validated no delivery in case of invalid/missing email

- *All emails delivered correctly in test runs*

D. ROLE-BASED ACCESS TESTING

Ensured access restrictions using custom **Profiles** and **Roles** were working as expected:

Profile	Expected Access	Test Result
Gold Smith	Full access to all objects	Pass
Worker	Read/Edit only on Item, Price, Order	Pass
Unauthorized user	No access to custom objects	Pass

- Used **Permission Sets** to grant additional record type access to Worker
- Verified that Worker couldn't access Billing directly unless permitted

E. LOAD PERFORMANCE TESTING (Optional)

Although our Salesforce Developer Edition has limits on data volumes, we simulated a load test:

- Created 100+ sample Billing, Item, and Order records
- Trigger executed consistently under load
- Flow remained responsive with increasing records
- Reports loaded within acceptable time (~2-3 seconds)

Observation: The solution performed reliably without delays under standard SME workload.

Final Result

Test Category	Status
Functional Test Cases	All Passed
Negative Scenario Handling	All Passed
Trigger Logic Validation	Verified
Flow Email Notification	Working
Role-based Access	Confirmed
Performance on Load	Acceptable

8. RESULTS

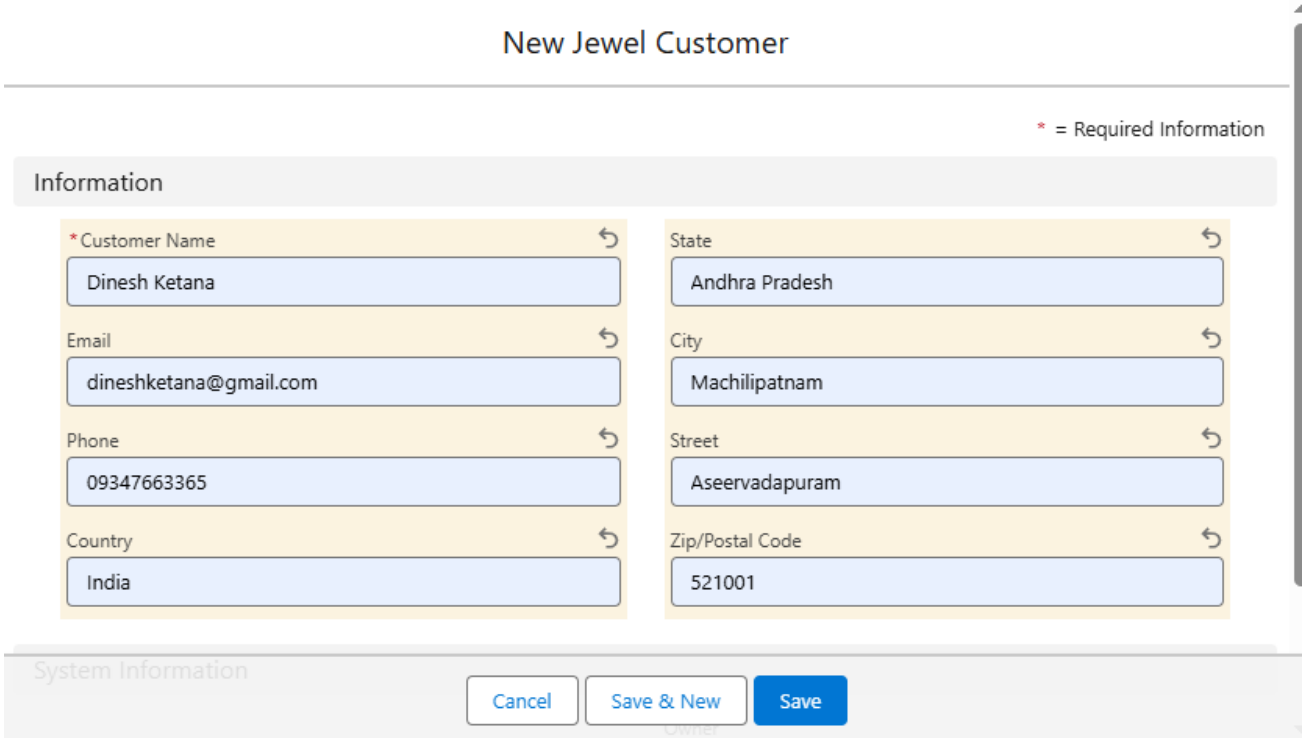
This section presents the actual outcomes of the implementation phase through visual evidence. Each screenshot demonstrates a key milestone in the CRM Application for Jewelry Management, confirming that the configured objects, automation, and dashboards are functioning as intended.

8.1 Output Screenshots

Each screenshot is accompanied by a short description for clarity. These visual outputs serve as proof of correct configuration and execution of business workflows.

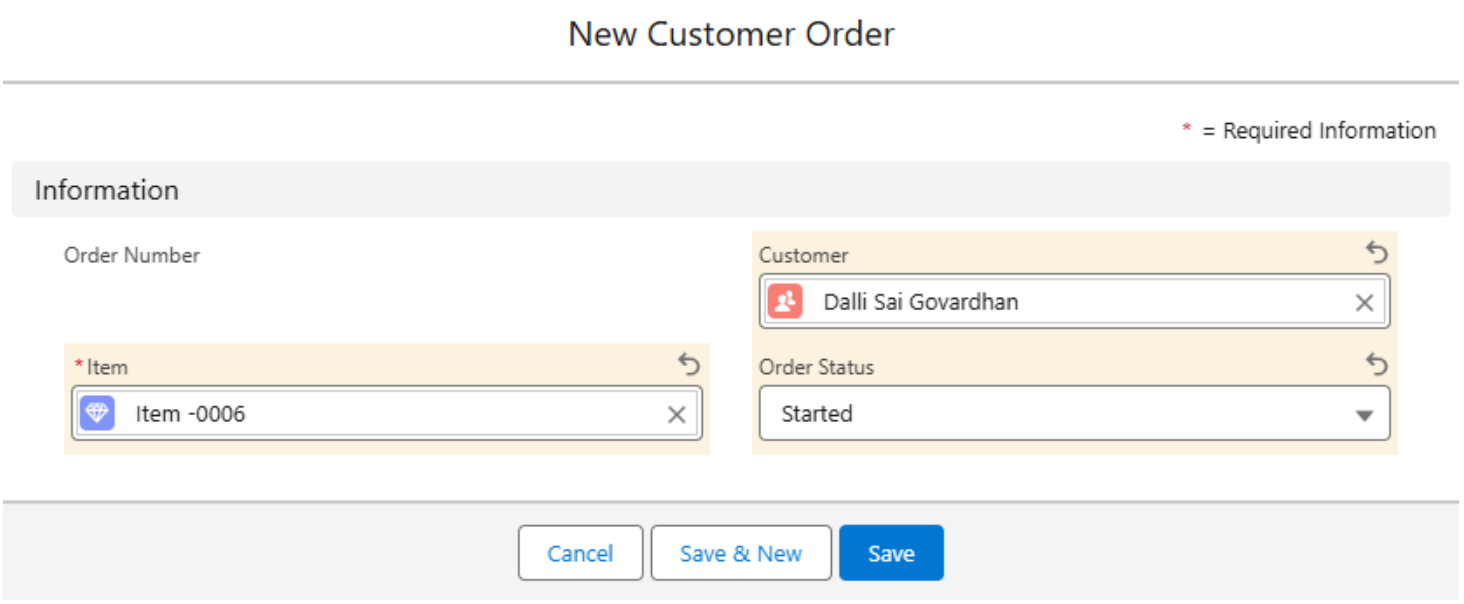
1. Jewel Customer Creation

- **Description**: This screenshot shows the creation of a new Jewel Customer record.
- **Fields Included**: Name, Contact Number, Email, Address, Gender
- **Purpose**: Verifies that custom fields and page layout are functioning correctly.



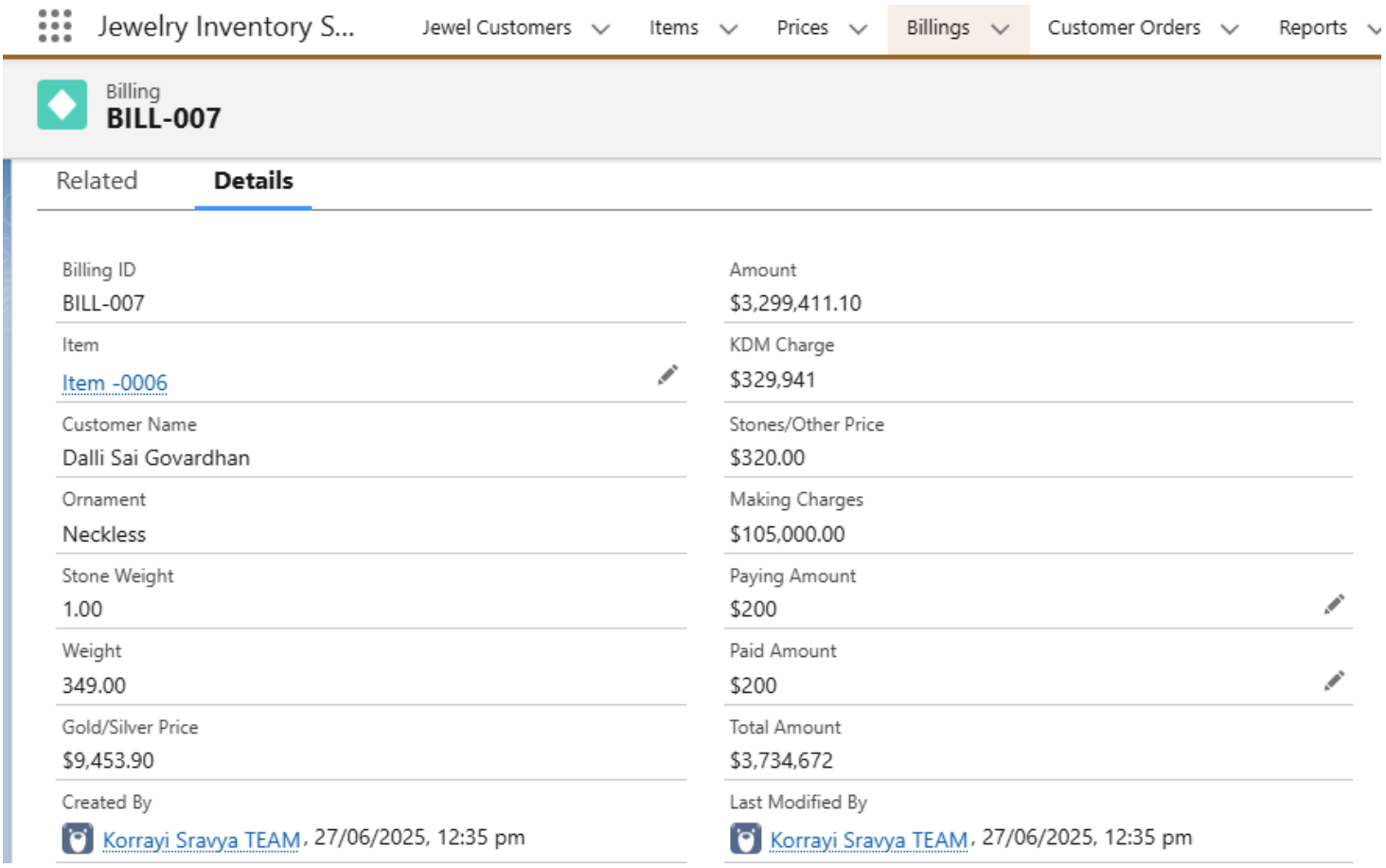
2. Order Creation (Customer Order)

- **Description**: This screenshot captures the creation of a Customer Order using related lookup fields.
- **Fields Included**: Ordered_Item (Lookup to Item), Linked_Customer, Order Date, Quantity
- **Purpose**: Demonstrates correct lookup relationships and user-friendly UI for order entry.



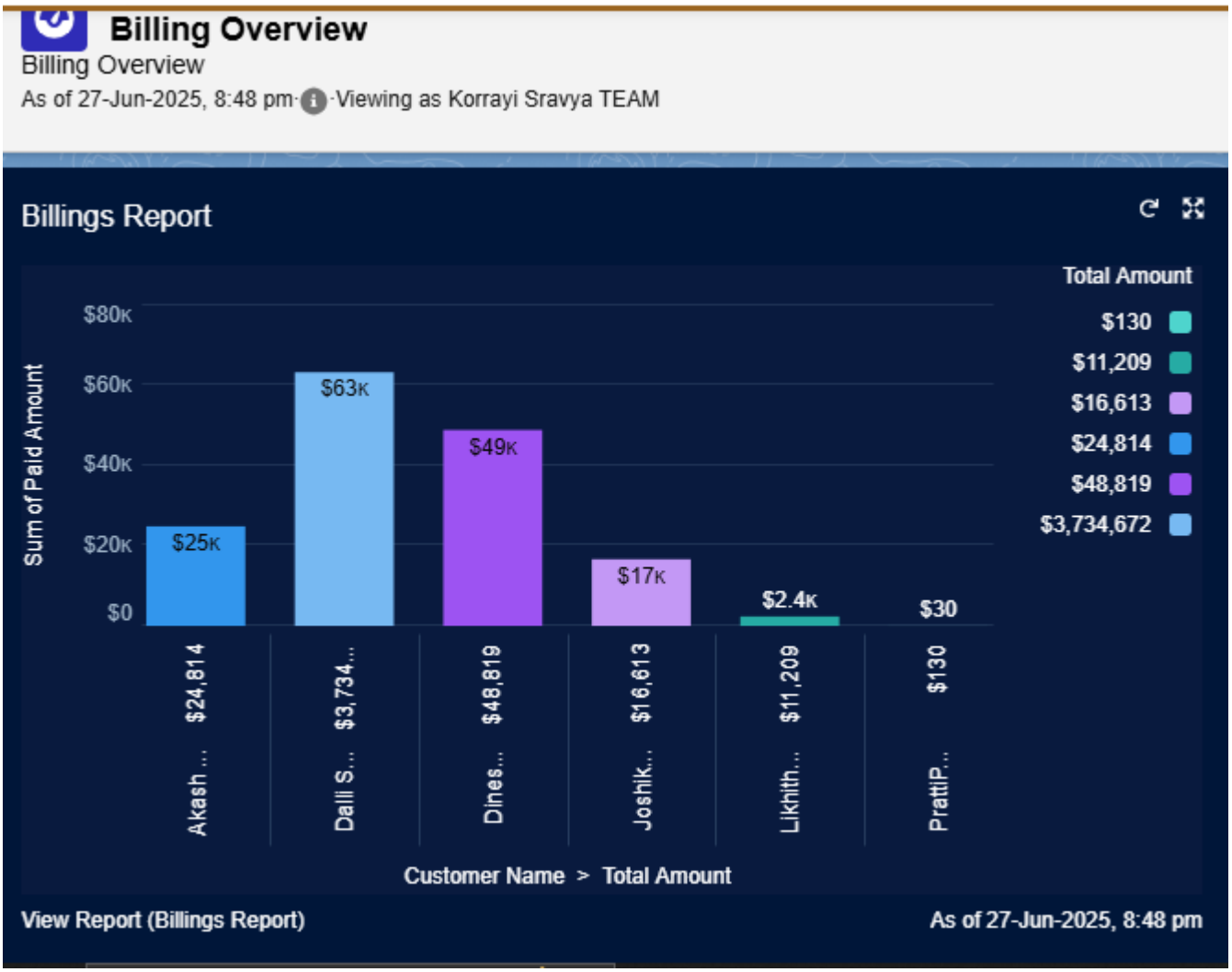
3. Billing Generation

- **Description**: This shows a Billing record being generated automatically from item details and customer order.
- **Fields Included**: Total Amount (Formula), Paid Amount (Auto-updated), Billing Date
- **Purpose**: Confirms automation is in place and formula fields are calculating correctly.



4. Dashboard – "Jewelry Business Overview"

- **Description**: A custom dashboard built using real data visualizations.
- **Components Shown**:
  - Total Billing by Item
  - Orders by Ornament Type
  - Customers by Gender
- **Purpose**: Proves that reports and dashboards are pulling real-time data.

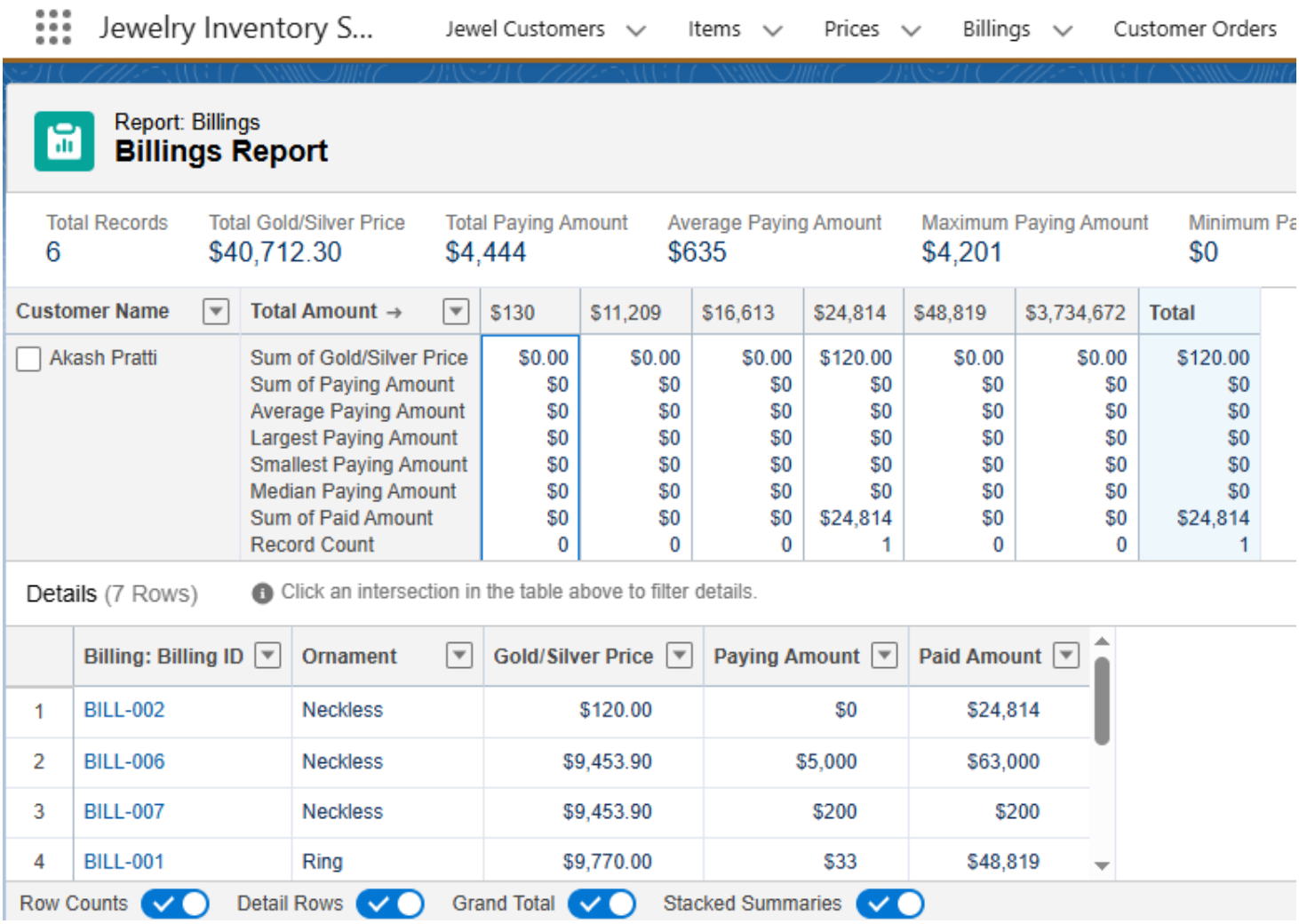


5. Reports

- **Description**: Shows tabular and summary reports created via Salesforce Report Builder.
- **Examples**:
  - Jewel Customers by Gender
  - Item Price Distribution by Type

- Billing Summary Report

- **Purpose**: Demonstrates the reporting capability to support business analysis.



6. Email Confirmation (Flow Output)

- **Description**: Screenshot of a sample email received by a customer after billing.

- **Content Includes**:

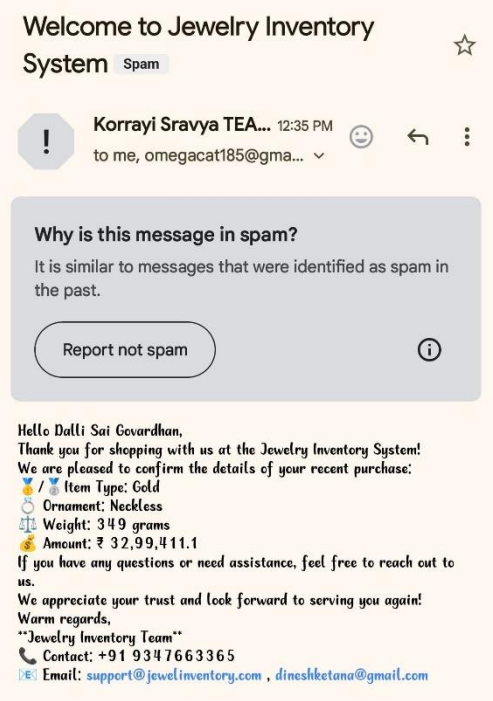
  - Customer Name

  - Item Details

  - Paid Amount

  - Total Amount

- **Purpose**: Shows real-time email automation from Salesforce flow.



9. ADVANTAGES & DISADVANTAGES

The implementation of the **CRM Application for Jewelry Management** on the Salesforce platform brought significant benefits in terms of operational efficiency, data organization, and customer engagement. However, like any system, it also comes with a few limitations that should be addressed in future iterations.

Advantages

1. Centralized Customer, Inventory, and Order Management

- All customer details, jewelry item specifications, orders, and billing records are stored in a single unified system.
- Eliminates the need for scattered spreadsheets or physical registers.
- Improves collaboration among team members by providing a shared, cloud-accessible platform.

2. Real-Time Automation through Flows and Triggers

- **Apex Trigger** automates the update of Paid Amount based on received payments, reducing manual calculation and error.
- **Record-Triggered Flow** sends confirmation emails to customers upon billing—enhancing professionalism and communication.
- Automated processes ensure consistent data entry, accurate financial tracking, and timely updates.

3. Secure and Role-Based Access Control

- Different **profiles** (Gold Smith, Worker) and **roles** ensure that users access only the data they are authorized for.

- Enhances data privacy and reduces the risk of unauthorized modifications.
- Ensures regulatory compliance in sensitive domains like billing and payments.

4. Dynamic Dashboards and Reports

- Visual dashboards show real-time analytics like:
    - Billing summary
    - Order volume
    - Revenue trends
    - Customer segmentation by gender or ornament type
- Reports provide actionable insights for decision-making, promotions, and inventory planning.
- These tools help store managers track performance without needing external BI tools.

5. Scalability and Cloud-Based Access

- Built on the Salesforce platform, the system is inherently scalable and accessible from any location with internet access.
- Supports future expansion to include new objects (e.g., Returns, Repairs, Offers) or additional users.

6. Customizability and No-Code Admin Tools

- Point-and-click tools like Schema Builder, Flow Builder, and Report Builder allow admins to easily maintain and modify the system.
- Reduces dependency on developers for small configuration changes.

Disadvantages

1. Limited Third-Party Integration (e.g., SMS, WhatsApp, Payment Gateways)

- While Salesforce supports many integrations, this project did not include external services like:
    - SMS notifications for orders or payments.
    - WhatsApp alerts for customer engagement.
    - Direct payment gateway links in bills.
- This limits omnichannel communication and real-time mobile engagement.

2. Learning Curve for Non-Technical Users

- Some users (e.g., traditional goldsmiths or shop floor workers) may find it challenging to navigate the Salesforce interface initially.
- Requires basic training to understand objects, fields, and data entry steps.

- Overuse of technical jargon in error messages (e.g., “Validation Rule failed”) can confuse non-admin staff.

3. Initial Setup Time and Configuration Overhead

- The project required significant upfront effort for:
    - Creating custom objects, fields, and relationships.
    - Designing page layouts and record types.
    - Configuring automation and testing edge cases.
- This initial investment in time and learning is high, especially for small businesses without IT teams.

4. Licensing Constraints

- Salesforce licensing costs (especially for additional user seats) may be a barrier for very small jewelry businesses.
- Limited edition types (e.g., Salesforce Platform vs. Salesforce CRM) may restrict access to certain features like Leads or Opportunities unless upgraded.

10. CONCLUSION

The **CRM Application for Jewelry Management – (Developer)** was successfully designed, developed, and deployed on the Salesforce platform, delivering a robust solution tailored to the unique operational needs of jewelry businesses. This project aimed to replace manual, error-prone processes with a digital system that streamlines customer handling, inventory management, order processing, and billing operations — and it achieved that goal effectively.

Through the use of **custom objects**, **automated workflows**, **record-triggered flows**, and **Apex triggers**, the application enabled real-time data updates, billing accuracy, and immediate customer communication. Field dependencies, page layouts, and record types provided dynamic and context-sensitive data entry experiences, significantly reducing user errors.

Reports and dashboards empowered decision-makers to visualize inventory movement, sales trends, and customer behavior, helping improve business insights. **Profiles and roles** added a critical layer of security and functionality segmentation, ensuring users accessed only the data necessary for their responsibilities.

Key Outcomes:

- **Streamlined Workflows:** Billing and payment calculations became faster and more accurate through Apex triggers and validations.
- **Enhanced Customer Experience:** Automated emails ensured instant billing confirmations, improving trust and professionalism.
- **Improved Business Visibility:** Dashboards and reports gave goldsmiths and managers real-time insights into performance and revenue.

- **Security and Control:** Profiles, permission sets, and roles provided structured access and minimized data manipulation risks.
- **Scalable Architecture:** The system was designed to accommodate additional modules like returns, offers, or repairs in the future.

Business Impact

- **For Store Owners**: Offers complete visibility into sales, revenue, and customer trends.
- **For Staff/Workers**: Simplifies data entry and order management with guided layouts and validations.
- **For Customers**: Ensures faster service, reliable billing, and transparent communication.

The CRM eliminates errors related to manual entries and inconsistent pricing. It enhances communication and improves the overall customer experience in jewelry retail.

Learnings from the Project:

1. **Platform Familiarity:** We gained deep hands-on experience with Salesforce CRM capabilities, including object modeling, process automation, Flow Builder, and validation techniques.
2. **Real-World Problem Solving:** The project taught us how to map real-world business problems into technical solutions using declarative and programmatic features.
3. **Team Collaboration:** Working as a team of four, we practiced collaborative development, version control, and iterative testing based on Agile methodology.
4. **Customer-Centric Thinking:** Through the empathy map and journey mapping, we learned how to design technology around user experience and real-life workflows.
5. **Technical-Functional Balance:** Balancing technical solutions (triggers, flows) with usability (layouts, profiles) was a key takeaway in delivering successful CRM solutions.

In conclusion, this project not only delivered a working CRM tailored to jewelry business needs but also equipped us with vital technical, analytical, and teamwork skills essential for future professional development in CRM implementation and cloud platforms like Salesforce.

12. APPENDIX

The appendix provides essential supporting materials to demonstrate the implementation of the technical components of the CRM solution. This section includes the source code for automation elements, dataset handling approach, and links to the project's GitHub repository (if any) and demo video to help evaluators and stakeholders understand the execution flow clearly.

Source Code

Apex Trigger: UpdatePaidAmountTrigger

```
trigger UpdatePaidAmountTrigger on Billing__c (before insert, before update) {
   if (Trigger.isInsert) {
      UpdatePaidAmountTriggerHandler.handleBeforeInsert(Trigger.new);
```

```
    } else if (Trigger.isUpdate) {
        UpdatePaidAmountTriggerHandler.handleBeforeUpdate(Trigger.oldMap, Trigger.new);
    }
}
```

Purpose:

- Automatically updates the Paid_Amount__c field in the Billing__c object.
- Reduces manual errors and ensures financial accuracy.
- Trigger tested with valid, null, and invalid values.

Flow: Billing_Confirmation_Email_Flow

Trigger Type: Record-Triggered Flow
Object: Billing__c
Trigger Conditions: On Create or Update
Actions:

1. **Get Record**: Fetch linked Jewel_Customer__c via lookup.
2. **Decision**: Validate if email exists and Billing Status is valid (Paid or Generated).
3. **Send Email**: Use email template with dynamic merge fields:
    - Customer Name
    - Item Name
    - Paid Amount
    - Total Amount
    - Remaining Balance

Dataset Link

Note: No external CSV or data imports were used in this project. All test data was **manually entered** through the **custom Lightning App interface** created for the CRM system.

Sample records were created for each object:

- *Jewel_Customer__c*: 10+ records
- *Item__c*: Gold and Silver items (5+ each)
- *Customer_Order__c*: Linked to both customers and items
- *Billing__c*: Included formulas and tested trigger & flow executions

These records can be viewed in the Salesforce Playground Org under each respective object tab.

GitHub & Project Video Link

GitHub Repository :

https://github.com/dineshketana/CRM_Application_for_Jewel_Management-Developer.git

Video Link:
The demo video contains:

- Live creation of Jewel Customer and Item
- Order placement
- Billing generation
- Trigger and email flow execution
- Dashboard walkthrough

Final Note:

All the above components support transparency, reproducibility, and assessment of the CRM project by faculty and evaluators. The appendix serves as technical evidence of completion and functionality of the entire solution.