

Phase 1: Problem Understanding & Industry Analysis

Problem Statement:

Pharmacy businesses often face challenges in managing customer orders, validating prescriptions, tracking deliveries, and monitoring medicine stock levels. Manual processes lead to delays, errors, poor customer experience, and regulatory compliance issues. There is a need for a centralized Pharmacy Delivery CRM that streamlines customer management, order processing, delivery tracking, and stock monitoring within a secure and scalable platform like Salesforce.

Requirement Gathering:

- Customers should be able to place medicine orders and upload prescriptions.
- Pharmacists should validate prescriptions before dispatch.
- Delivery staff should update delivery status in real time.
- Admin should manage users, medicines, stock, and view analytics.
- The system should send automated notifications (email/SMS) for order updates.

Stakeholder Analysis:

- Customers → Place orders, upload prescriptions, track deliveries.
- Pharmacists → Approve prescriptions, manage inventory, ensure compliance.
- Delivery Staff → Update delivery progress, confirm delivery completion.
- Admin/Management → Oversee operations, manage users/permissions, generate reports & dashboards.
- Customer Support → Handle order issues, complaints, and escalations.

Business Process Mapping:

1. Customer registers and places an order.
2. Pharmacist reviews and approves prescription (if required).
3. Order is confirmed, stock is updated, delivery staff assigned.
4. Delivery staff updates order status (Out for Delivery → Delivered).
5. Customer is notified throughout the process.
6. Admin monitors sales, deliveries, and stock using dashboard

Industry-Specific Use Case Analysis:

- Healthcare & Pharma Industry Challenges: Compliance, timely delivery of medicines, patient data security, and maintaining stock accuracy.
- Use Case Fit: Pharmacy Delivery CRM ensures secure prescription handling, order accuracy, real-time delivery tracking, and customer satisfaction.
- Benefits: Improved operational efficiency, reduced errors, compliance adherence, and better customer experience.

AppExchange Exploration:

- Explore Salesforce Health Cloud & Pharma-specific apps.
- Reference existing delivery/medicine tracking solutions (e.g., Veeva CRM, Field Service Lightning).

Phase 2: Org Setup & Configuration:

Introduction:

In today's fast-paced world, pharmacies are no longer limited to walk-in customers. The increasing demand for home delivery of medicines has created the need for a streamlined system to manage customers, prescriptions, orders, deliveries, and feedback efficiently.

A **Pharmacy Delivery CRM (Customer Relationship Management system)** provides a centralized platform to manage the entire medicine delivery lifecycle — from receiving a prescription, verifying stock, processing an order, assigning delivery agents, and tracking the delivery, to handling customer support and complaints.

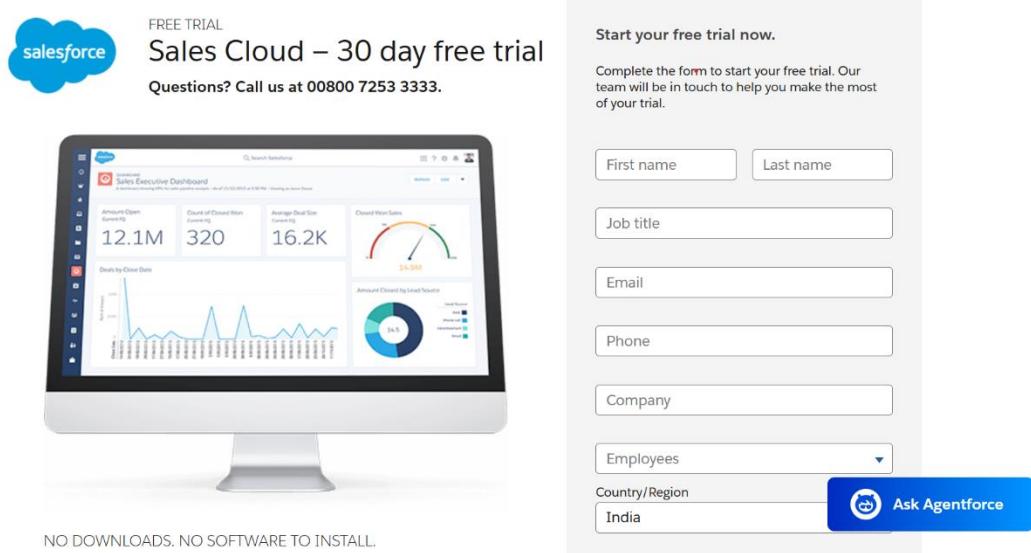
By leveraging **Salesforce CRM**, pharmacies can automate workflows, manage data securely, and provide customers with a seamless and reliable healthcare delivery experience.

Key Objectives:

1. Select the Right Salesforce Edition – Choose the edition that balances cost and features suitable for pharmacy delivery operations.
2. Establish Company Identity – Configure Company Profile (name, currency, time zone, locale) for Medi Track Pharmacy Pvt Ltd.
3. Define Business Operations – Set Business Hours & Holidays to align medicine orders, deliveries, and pharmacist availability.
4. Align Financial Tracking – Configure Fiscal Year Settings to support accurate sales, stock, and delivery reporting.
5. Enable User Access & Control – Create Users with correct Licenses, Profiles, and Roles (e.g., Pharmacist, Delivery Agent, Admin).
6. Strengthen Data Security – Apply Permission Sets, OWD, and Sharing Rules to safeguard sensitive health and delivery data.
7. Manage System Access – Define Login Access Policies to ensure only authorized staff can access CRM securely.

3.Activities Performed in Phase-2:

3.1 Salesforce Editions:



The image shows the Salesforce Sales Cloud - 30 day free trial landing page. It features the Salesforce logo at the top left, followed by a 'FREE TRIAL' button and the text 'Sales Cloud – 30 day free trial'. Below this, it says 'Questions? Call us at 00800 7253 3333.' A central image of a computer monitor displays the 'Sales Executive Dashboard' with various metrics like 'Amount Open', 'Count of Closed Won', 'Average Deal Size', and a chart titled 'Closed Won Sales'. Below the dashboard, the text 'NO DOWNLOADS. NO SOFTWARE TO INSTALL.' is visible. To the right, there's a form titled 'Start your free trial now.' with fields for 'First name', 'Last name', 'Job title', 'Email', 'Phone', 'Company', 'Employees', and 'Country/Region' (set to 'India'). A blue button at the bottom right says 'Ask Agentforce'.

Salesforce offers a variety of editions to meet the diverse needs of businesses, from small startups to large enterprises. Each edition is designed to provide unique features, scalability, and customization options, so organizations can select the version that best aligns with their requirements.

1.Essentials Edition:

Overview: Designed for small businesses, the Essentials edition provides a simple, user-friendly platform to manage customer relationships. It includes essential CRM tools like contact management, lead management, and basic sales automation features.

2.Professional Edition:

Overview: The Professional edition is ideal for growing businesses that require advanced features, more customization, and better automation. It includes everything in the Essentials edition, with added capabilities for sales teams and integrations with other business tools.

3.Enterprise Edition:

Overview: The Enterprise edition is built for businesses that need advanced customization, complex workflows, and enterprise-level functionality. It includes everything in the Professional edition and provides even more flexibility for larger organizations.

These are the few editions available on salesforce platform.

3.2 Company Profile Setup: Organization information Configuration

- **Company Name:** Medi Track Pharmacy Pvt Ltd
The official business name used across CRM for branding, reporting, and customer interactions.
- **Primary Contact:** Admin User System Administrator
The main person responsible for managing the Salesforce org setup.

- **Default Locale:** English (India)
(Ensures date, time, number, and currency formatting follows Indian standards.)
- **Default Currency:** INR (₹)
(All transactions, order values, and reports are managed in Indian Rupees.)
- **Default Time Zone:** Asia/Kolkata (IST)
(Aligns order processing, deliveries, and customer support with Indian Standard Time.)

The screenshot shows the Salesforce Setup interface with the 'Company Information' page selected. The page displays the organization's profile for 'MediTrack Pharmacy Pvt Ltd'. Key details include:

- Organization Name:** MediTrack Pharmacy Pvt Ltd
- Primary Contact:** [Placeholder]
- Division:** [Placeholder]
- Address:** IN
- Fiscal Year Starts In:** January
- Activate Multiple Currencies:**
- Enable Data Translation:**
- Newsletter:**
- Phone:** [Placeholder]
- Fax:** [Placeholder]
- Default Locale:** English (India)
- Default Language:** English
- Default Time Zone:** (GMT+05:30) India Standard Time (Asia/Kolkata)
- Currency Locale:** English (India) - INR
- Used Data Space:** 390 KB (8%) [\[View\]](#)
- Used File Space:** 94 KB (0%) [\[View\]](#)

The screenshot shows the Salesforce Setup interface with the 'Company Information' page selected. The page displays various configuration settings for the organization:

- Activate Multiple Currencies:**
- Enable Data Translation:**
- Newsletter:**
- Admin Newsletter:**
- Hide Notices About System Maintenance:**
- Hide Notices About System Downtime:**
- Locale Formats:** ICU
- Currency Locale:** English (India) - INR
- Used Data Space:** 390 KB (8%) [\[View\]](#)
- Used File Space:** 21 KB (0%) [\[View\]](#)
- API Requests, Last 24 Hours:** 0 (15,000 max)
- Streaming API Events, Last 24 Hours:** 0 (10,000 max)
- Restricted Logins, Current Month:** 0 (0 max)
- Salesforce.com Organization ID:** 00DWU00000W0vE2
- Organization Edition:** Developer Edition
- Instance:** SWE106

At the bottom, it shows the last modified by Brunda Sanjana on 23/09/2025, 3:38 pm.

3.3 Business Hours & Holidays:

Medi Track Pharmacy Pvt Ltd operates with extended service hours to meet customer needs:

- **Name:** Medi Track Pharmacy Pvt Ltd
- **Working Days:** Monday – Sunday (*7 days a week to ensure medicine availability*)
- **Start Time:** 9:00 AM IST

- **End Time:** 9:00 PM IST
- **Time Zone:** Asia/Kolkata (IST)

These hours ensure customers can place orders, request deliveries, and connect with pharmacy staff throughout the day, while delivery agents are scheduled efficiently.

3.3 Holidays:

To maintain service consistency and plan operations, certain national and regional holidays are marked as non-working days:

- Republic Day – 26th January
- Independence Day – 15th August
- Gandhi Jayanti – 2nd October

On holidays, delivery operations are paused or rescheduled, while the CRM ensures orders placed are queued for the next business days

Holiday Name	Description	Date and Time
GandhiJayanti		02/10/2025 All Day
Independence Day		15/08/2026 All Day
Republic Day		26/09/2025 All Day

3.4 Fiscal Year Settings:

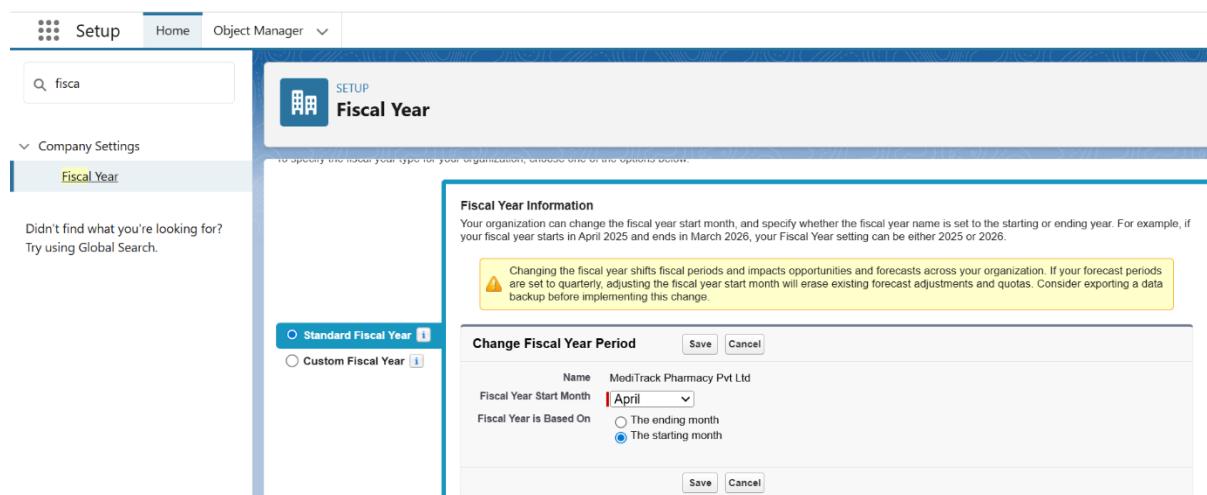
Fiscal Year Type:

- **Type:** Standard Fiscal Year
- **Start Month:** April
- **End Month:** March (next year)

This aligns with the **Indian financial year cycle (April–March)**, which is commonly followed by pharmacies, healthcare providers, and businesses for accounting and compliance.

Why Standard Fiscal Year?

- Pharmacy order management, delivery revenues, and supplier billing are reported against the financial year.
- Ensures smooth integration with accounting systems used in India.
- Reduces complexity since no custom fiscal calendar is required.



3.5 User Setup & Licenses:

User Setup & Licenses – Medi Track Pharmacy Pvt Ltd

Objective: Define the different types of users who will access the Pharmacy Delivery CRM and assign appropriate licenses for their roles in the business.

User Groups in Medi Track Pharmacy CRM

1. System Administrator:
 - Role: Manages Salesforce org setup, configuration, users, and security.
 - License: Salesforce (full access).
2. Pharmacists:
 - Role: Manage medicines, prescriptions, and stock levels.

- License: Salesforce or Salesforce Platform (depending on access needs).

3. Delivery Agents:

- Role: View assigned delivery orders, update status (delivered/pending).
- License: Salesforce Platform (lightweight access to custom apps).

4. Doctors (Optional Integration):

- Role: Upload prescriptions, verify medicine recommendations.
- License: Salesforce Platform or External Community License (if giving limited external access).

5. Customers (Optional via Portal)

- Role: Place orders, upload prescriptions, track delivery status.
- License: Customer Community License (if portal/self-service is implemented).

Key Fields for User Setup

- First Name / Last Name → Users enters his/her name
- Email & Username → Must be unique
- Role → Pharmacist, Delivery Agent, Admin, etc.
- Profile → Defines object-level permissions.
- License → Determines what functionality the user can access.

The screenshot shows the Salesforce Setup interface with the following details:

- User Detail:**
 - Name: pharmacist delivery
 - Alias: pharma
 - Email: brundajbs@gmail.com [Verify]
 - Username: brundajbs@gmail.com
 - Nickname: User17589890052989137555
 - Title: pharmacist
 - Company:
 - Department:
 - Division:
 - Address:
- Role:** Director_Direct Sales
- User License:** Salesforce
- Profile:** Standard User
- Active:** checked
- Other Options:** Edit, Sharing, Reset Password, Freeze, View Summary

Page Header: Search Setup

Page Title: SETUP Users

Page Subtitle: User pharmacist delivery

Page Footer: User ProfileHelp for this Page

3.6 Roles

Objective: Roles in Salesforce define a hierarchy of users to control record-level visibility. While profiles control permissions (what a user can do), roles control whose records a user can see. This is critical in Pharmacy Delivery CRM to ensure sensitive data (prescriptions, orders, deliveries) is visible only to the right people.

Role Hierarchy in Medi Track Pharmacy CRM:

1. System Administrator

- **Position:** Top of the hierarchy
- **Responsibilities:**
 - Full visibility across all records and objects.
 - Manages users, profiles, permissions, and overall CRM configuration.

2. Pharmacy Manager / Lead Pharmacist

- **Position:** Reports to Admin
- **Responsibilities:**
 - Oversee pharmacists and delivery agents.
 - View all orders, deliveries, prescriptions in their branch or region.
 - Approve special medicine orders.

3. Pharmacist

- **Position:** Reports to Pharmacy Manager
- **Responsibilities:**
 - Manage prescriptions, stock, and medicine orders.
 - Only sees records relevant to their branch or assigned patients.

4. Delivery Agent

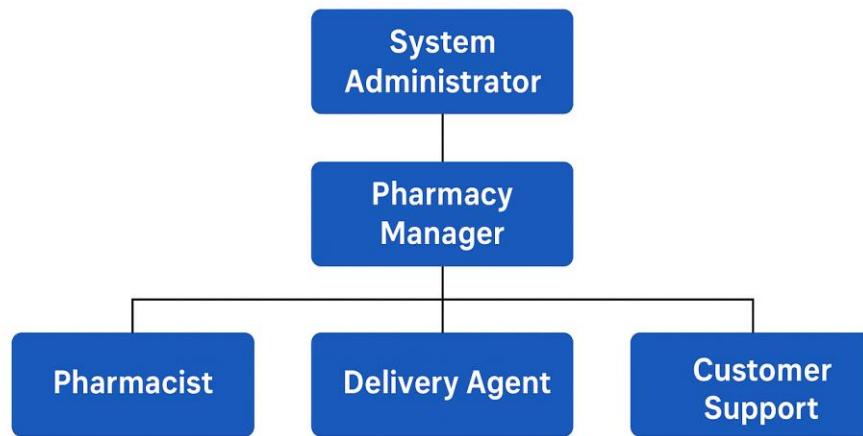
- **Position:** Reports to Pharmacy Manager / Delivery Supervisor
- **Responsibilities:**
 - View assigned orders and delivery details.
 - Update delivery status (Delivered, Pending, Delayed).
 - Cannot see full customer or prescription data beyond what is necessary for delivery.

5. Customer Support

- **Position:** Reports to Pharmacy Manager
- **Responsibilities:**
 - Handle customer queries and complaints.
 - View order and delivery status.

MediTrack Pharmacy Pvt Ltd

Roles



The screenshot shows the Salesforce Setup Roles page. The left sidebar is titled 'Setup' and includes links for 'Salesforce Mobile App', 'Lightning Usage', 'Optimizer', 'Sales Cloud Everywhere', 'ADMINISTRATION', 'Users', 'Permission Set Groups', 'Permission Sets', 'Profiles', 'Public Groups', 'Queues', 'Roles' (which is selected), 'User Management Settings', 'Users', and 'Data'. The main content area is titled 'SETUP Roles' and displays the 'Your Organization's Role Hierarchy'. It shows a tree structure of roles under 'MediTrack Pharmacy Pvt Ltd': CEO, Admin, Pharmacy Manager, Managers, manages delivery, SVP.Customer Service & Support, and SVP.Sales & Marketing. Each role has 'Edit | Del | Assign' options and an 'Add Role' link. A 'Show in tree view' dropdown is also present.

3.7 Permission Sets:

1. Access to Sensitive Patient Information

Description: Certain patient data like medical history, prescription details, and payment information are sensitive. A permission set can grant Read/Write access to these fields only to authorized roles such as pharmacists or finance staff, ensuring confidentiality while allowing necessary operations.

2. Temporary Delivery Staff Access

Description: Delivery personnel may need limited access to patient addresses, order statuses, and contact numbers during their shift. A permission set can provide Read-only access to relevant objects like Orders and Patients without exposing financial or medical data.

3. Pharmacist Access to Prescription Management

Description: Pharmacists require full access to the Prescriptions object to validate, update, and fulfil prescriptions. A permission set can grant Create, Read, Edit (CRUD) permissions on Prescriptions and related objects, without changing their base profile.

4. Marketing and Promotions Access

Description: Marketing team members need to run campaigns and send notifications to patients. A permission set can provide access to the Campaigns object and marketing fields, without exposing order or payment information.

5. Manager Access to Reports and Dashboards

Description: Pharmacy managers require analytics on sales, delivery performance, and customer trends. A permission set can enable **Report and Dashboard access** along with visibility to aggregated data, without granting record modification rights.

6. Access to Prescription Reminders

Description: Customer support or nurses managing reminders need access to the Prescription Reminder object to send notifications. A permission set can provide **Read/Write access** to this object without giving access to core patient medical records.

The screenshot shows the Salesforce Setup interface for configuring a permission set named 'Prescription Reminder'. The left sidebar navigation includes 'Setup', 'Home', 'Object Manager', and sections for 'Users' (Permission Set Groups, Permission Sets, Profiles, Public Groups, Queues, Roles, User Management Settings), 'Feature Settings', and 'Data.com' (Prospector). The main content area displays the 'Current Assignments' for the 'Prescription Reminder' permission set. A table lists one assignment: 'Pharmacy Manager' (Active, Role: Pharmacy Manager, Profile: Standard Platform User, User License: Salesforce Platform, Expires On: Not specified).

Full Name ↑	Active	Role	Profile	User License	Expires On
Pharmacy Manager	✓	Pharmacy Manager	Standard Platform User	Salesforce Platform	

3.8 OWD:

Definition:

OWD defines the baseline level of access to records for all users in the organization. It sets who can see or edit records by default, before sharing rules, roles, or permission sets are applied.

1. Patient Object:

- Recommended OWD: Private
- Reasoning: Patient information is sensitive (medical history, prescriptions, payment info). Users should only see records they own or that are explicitly shared.

2. Prescription Object

- Recommended OWD: Private
- Reasoning: Prescriptions contain sensitive health data. Access should be restricted to the prescribing pharmacist, doctor, or assigned delivery staff.

3. Orders Object

- Recommended OWD: Controlled by Parent (Patient) or Private
- Reasoning: Orders are linked to patients. Staff involved in fulfilling the order should have access, but others should not. Using "Controlled by Parent" ensures alignment with patient record visibility.

4. Delivery Object

- Recommended OWD: Private
- Reasoning: Delivery details include patient addresses. Access should be restricted to the delivery team and relevant managers.

5. Reports and Dashboards

- Recommended OWD: Public Read-Only or Private
- Reasoning: Some reports may contain aggregated sensitive data (sales, patient trends). Restrict access to managers or authorized roles.

The screenshot shows the Salesforce Sharing Settings page. At the top, there's a search bar and a toolbar with icons for Home, Object Manager, and Setup. The main area has a title 'Sharing Settings' with a 'SETUP' button. Below it is a table titled 'Default Sharing Settings' with a 'Organization-Wide Defaults' header. The table lists various objects (Lead, Account and Contract, Contact, Order, Asset, Opportunity, Case, Campaign, Campaign Member, User, Activity, Calendar) and their sharing settings. The columns are 'Object', 'Default Internal Access', 'Default External Access', and 'Grant Access Using Hierarchies'. Most objects have 'Public Read/Write/Transfer' as the default internal access, while Contact, Order, Asset, Opportunity, Case, and Activity have 'Controlled by Parent' as the default internal access. The 'Grant Access Using Hierarchies' column contains checkmarks for most objects except for Contact, Order, Asset, Opportunity, Case, and Activity.

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Lead	Public Read/Write/Transfer	Private	✓
Account and Contract	Public Read/Write	Private	✓
Contact	Controlled by Parent	Controlled by Parent	✓
Order	Controlled by Parent	Controlled by Parent	✓
Asset	Controlled by Parent	Controlled by Parent	✓
Opportunity	Public Read/Write	Private	✓
Case	Public Read/Write/Transfer	Private	✓
Campaign	Public Full Access	Private	✓
Campaign Member	Controlled by Campaign	Controlled by Campaign	✓
User	Public Read Only	Private	✓
Activity	Private	Private	✓
Calendar	Hide Details and Add Events	Hide Details and Add Events	✓

3.9 Login Access Policies:

In the Pharmacy Delivery CRM, Login Access Policies define how administrators and Salesforce Support can temporarily access a user's account to troubleshoot issues. This is especially important in a healthcare delivery environment, where different user roles such as Pharmacists, Delivery Agents, and Customer Support Representatives may face issues related to prescriptions, medicines, or deliveries.

By enabling Login-As functionality, a System Administrator can log in as another user to replicate problems, validate permissions, and resolve workflow issues without needing the user's credentials. For example, if a Delivery Agent reports that they cannot update the *Delivery Status* of an order, that can use Login-As to check the problem directly from the Delivery Agent's perspective.

Key configurations include:

- Administrators Can Log in as Any User → Allows admins to quickly log in as Pharmacists, Delivery Agents, or Support Staff to debug issues.
- Available to Salesforce Support → Ensures Salesforce Support can access the org securely if needed for technical troubleshooting.
- Force Re login After Login-As → Maintains security by requiring the admin to log back in after ending the session.

This policy ensures faster resolution of operational issues such as prescription errors, delivery tracking problems, or patient support cases, while still maintaining strict security and accountability.

The screenshot shows the Salesforce Setup interface. The left sidebar has sections for Identity (Login Flows, Login History) and Security (Login Access Policies). A message at the bottom says "Didn't find what you're looking for? Try using Global Search." The main content area is titled "Login Access Policies" and contains a sub-section "Manage Support Options". It shows a table with one row for "Salesforce.com Support" where the "Enabled" setting is checked. A green "Changes Saved" message is displayed at the top of the form.

The screenshot shows the Salesforce Setup interface. The left sidebar has sections for My Personal Information, Aprover Settings, Authentication Settings for External Systems, Change My Password, Connections, External Credentials, Grant Account Login Access (which is selected), Language & Time Zone, Login History, Personal Information, Reset My Security Token, Display & Layout, and Email. The main content area is titled "Grant Account Login Access" and contains a sub-section "Grant Account Login Access". It shows a table with one row for "Salesforce.com Support" where the "Access Duration" is set to "6 day(s) left. Expires on 06/10/2025". A green "Changes Saved" message is displayed at the top of the form.

4. Key Outcomes of Pharmacy Delivery CRM:

1. Centralized Patient & Prescription Management

- Maintain a single view of patients with their prescriptions, medical history, and contact details.

2. Efficient Delivery Tracking

- Monitor real-time delivery status (Pending, Out for Delivery, Delivered).
- Assign deliveries to specific agents and ensure timely fulfillment.

3. Medicine Inventory Control

- Track stock levels of medicines in real time.
- Get automated notifications when critical medicines (e.g., Insulin) are low in stock.

5 Challenges and Resolutions:

Implementing a Pharmacy Delivery CRM comes with unique operational, technical, and compliance-related challenges. Below are the most common ones and how they can be addressed:

1. Challenge: Managing Sensitive Patient Data

- **Issue:** Patient records, prescriptions, and medical history are highly sensitive and need strict privacy controls.
- **Resolution:**
 - Implement Profiles, Roles, Permission Sets, and Field-Level Security to ensure only authorized users can view/edit data.
 - Set Organization-Wide Defaults (OWD) to Private for patient and prescription objects.
 - Use Sharing Rules to selectively grant access to delivery agents or pharmacists

2. Challenge: Accurate Medicine Inventory Management

- **Issue:** Overstocking or stockouts can lead to delivery delays or lost revenue.
- **Resolution:**
 - Track stock levels using custom Medicine objects with fields for quantity, reorder level, and expiry date.
 - Set up workflow rules or automation to alert pharmacists when stock falls below a threshold.
 - Generate reports for inventory forecasting.

6. Conclusion:

Phase 2 of the Pharmacy Delivery CRM focused on enhancing operational efficiency, role-based access, and automation across pharmacy processes. Key accomplishments include:

- Advanced User Access & Security: Configured Profiles, Roles, Permission Sets, OWD, and Sharing Rules to ensure that pharmacists, delivery agents, and support staff have appropriate access while maintaining data privacy.
- Automation & Workflows: Implemented workflow rules, process automation, and notifications to streamline prescription approvals, delivery status updates, and stock alerts.

Phase 2 successfully enhanced the CRM's usability, security, and automation capabilities, providing a robust operational framework that supports accurate prescription tracking, timely deliveries, and improved customer support.

Phase 3: Data Modelling & Relationships:

Introduction:

Phase 3 of the Pharmacy Delivery CRM focuses on advanced analytics, reporting, and optimization of pharmacy operations to support strategic decision-making. While Phase 1 and Phase 2 focused on basic setup, user access, workflows, and delivery management, Phase 3 aims to leverage CRM data for actionable insights, predictive planning, and enhanced operational efficiency.

Key aspects of Phase 3 include:

- **Advanced Reporting & Dashboards:** Generate real-time reports on prescriptions, deliveries, medicine stock levels, and customer support cases to monitor performance and trends.
- **Predictive Analytics:** Use historical data to forecast medicine demand, identify delivery bottlenecks, and optimize inventory levels.
- **Process Optimization:** Enhance workflows and automation based on insights gained from reports and analytics.

Objectives – Phase 3: Pharmacy Delivery CRM:

- Phase 3 of the Pharmacy Delivery CRM is designed to focus on analytics, optimization, and integration to transform the CRM from an operational tool into a strategic decision-making platform. The key objectives are:

1. Enhance Reporting and Analytics

- Build custom report types for prescriptions, deliveries, and medicine inventory.
- Develop dashboards for different roles (Pharmacists, Delivery Agents, Admins) showing KPIs like:
- Deliveries by status (Pending, Out for Delivery, Delivered).

2. Implement Predictive Insights

- Use CRM data to forecast medicine demand and prevent stockouts.
- Identify delivery bottlenecks (e.g., delayed deliveries in certain regions).
- Track patient ordering trends to plan pharmacy resources more effectively.

3. Detailed Activities Performed

3.1 Object Creation:

Standard Objects:

These are objects provided by Salesforce by default. In a Pharmacy Delivery CRM, they can be repurposed to manage pharmacy-related data.

Custom Objects (Created for Pharmacy Delivery CRM needs):

Custom objects extend Salesforce beyond standard healthcare CRM functions. For pharmacy delivery, you would create:

1. **Patients** → Stores patient details (Name, Age, Address, Contact Info, Medical History).
2. **Prescriptions** → Contains prescription details (Doctor, Medicine, Dosage, Validity Period, Status).
3. **Medicines** → Medicine inventory with fields like Stock Quantity, Expiry Date, Manufacturer, and Price.
4. **Deliveries** → Tracks delivery requests (Assigned Agent, Delivery Status, Delivery Date, Patient).
5. **Pharmacists** → Details about pharmacists (License Number, Shifts, Contact Info).

SETUP > OBJECT MANAGER
Medicine

Details	
Fields & Relationships	Description (Name, Expiry, Stock, Price)
Page Layouts	API Name Medicines__c
Lightning Record Pages	Custom
Buttons, Links, and Actions	✓
Compact Layouts	Singular Label Medicine
Field Sets	Plural Label Medicines
Object Limits	Enable Reports ✓
Record Types	Track Activities ✓
Related Lookup Filters	Track Field History ✓
	Deployment Status Deployed
	Help Settings Standard salesforce.com Help Window

SETUP > OBJECT MANAGER
Pharmacist

Details	
Fields & Relationships	Description
Page Layouts	API Name Pharmacist__c
Lightning Record Pages	Custom
Buttons, Links, and Actions	✓
Compact Layouts	Singular Label Pharmacist
Field Sets	Plural Label Pharmacists
Object Limits	Enable Reports
Record Types	Track Activities
	Track Field History
	Deployment Status Deployed
	Help Settings Standard salesforce.com Help Window

Setup Home Object Manager

SETUP > OBJECT MANAGER
Order

Details

- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters

Details

Description

API Name **Orders_c**

Custom ✓

Singular Label Order

Plural Label Orders

Edit Delete

Enable Reports ✓

Track Activities ✓

Track Field History ✓

Deployment Status **Deployed**

Help Settings Standard salesforce.com Help Window

Setup Home Object Manager

SETUP > OBJECT MANAGER
Delivery

Details

- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters

Details

Description

API Name **Delivery_c**

Custom ✓

Singular Label Delivery

Plural Label Deliveries

Edit Delete

Enable Reports

Track Activities

Track Field History

Deployment Status **Deployed**

Help Settings Standard salesforce.com Help Window

Setup Home Object Manager

SETUP > OBJECT MANAGER
Prescription

Details

- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters

Details

Description (Doctor Name, Notes, Medicines prescribed)

API Name **Prescription_c**

Custom ✓

Singular Label Prescription

Plural Label Prescriptions

Edit Delete

Enable Reports ✓

Track Activities ✓

Track Field History ✓

Deployment Status **Deployed**

Help Settings Standard salesforce.com Help Window

SETUP > OBJECT MANAGER
Patient

Details

Description

API Name
Patient__c

Custom

Singular Label
Patient

Plural Label
Patients

Enable Reports

Track Activities

Track Field History

Deployment Status
Deployed

Help Settings

Standard salesforce.com Help Window

Edit **Delete**

3.2 Fields:

Medicines (Custom Object)

Field Label	Type	Description
Medicine ID	Auto Number	Unique identifier for each medicine.
Medicine Name	Text	Commercial brand name of the medicine.
Generic Name	Text	Chemical or generic drug name.
Manufacturer	Text	Company manufacturing the medicine.
Stock Quantity	Number	Available stock for the medicine.

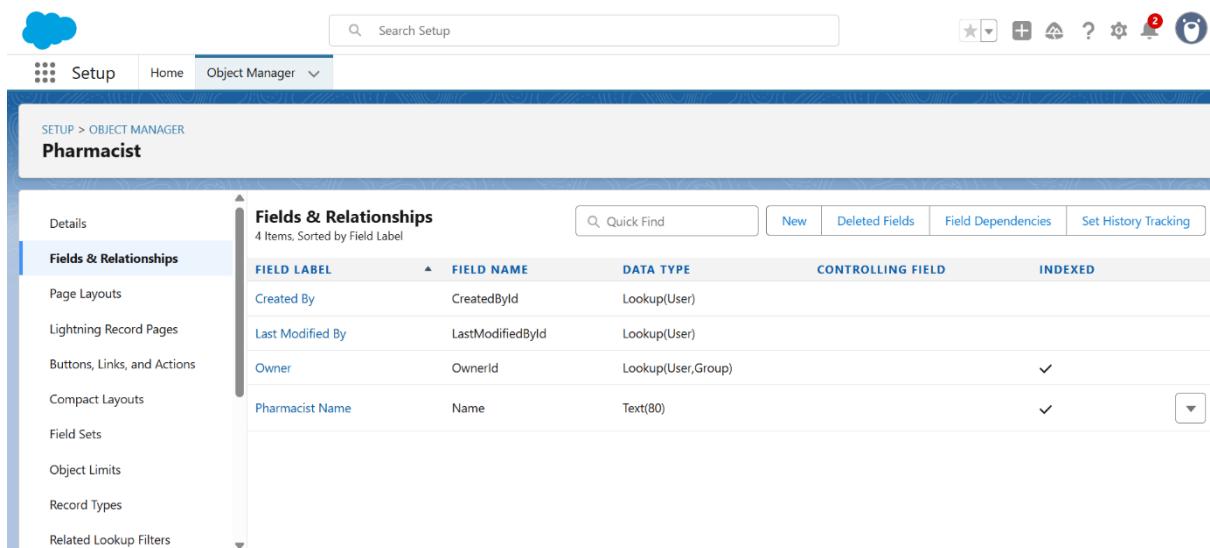
SETUP > OBJECT MANAGER
Medicine

Fields & Relationships
5 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Medicine Name	Name	Text(80)		✓
Medicine Price	Medicine_Price__c	Currency(18, 0)		
Owner	OwnerId	Lookup(User,Group)		✓

Pharmacists (Custom Object)

Field Label	Type	Description
Pharmacist ID	Auto Number	Unique identifier for each pharmacist.
Full Name	Text	Pharmacist's complete name.
License Number	Text	Professional license ID.



The screenshot shows the Salesforce Object Manager interface for the 'Pharmacist' custom object. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main content area is titled 'Fields & Relationships' and displays four fields: Created By, Last Modified By, Owner, and Pharmacist Name. The 'Owner' field is marked as indexed. The top navigation bar includes a search bar, a setup icon, and various system navigation icons.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Pharmacist Name	Name	Text(80)		✓

Delivery Agents (Custom Object)

Field Label	Type	Description
Agent ID	Auto Number	Unique identifier for each delivery agent.
Full Name	Text	Delivery agent's complete name.
Phone Number	Phone	Contact number of the delivery agent.
Email	Email	Official email address.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Delivery Status	Delivery_Status__c	Picklist		
Last Modified By	LastModifiedById	Lookup(User)		
Order Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User,Group)		✓
Record Type	RecordTypeId	Record Type		✓

3.3 Page Layouts:

Page Layouts in Salesforce determine how information is displayed for users when they view or edit records. In the Pharmacy Delivery CRM, page layouts are customized to ensure that pharmacists, delivery agents, and administrators see only the relevant fields, related lists, and actions they need for their roles.

Purpose:

- Organize fields into logical sections for easy data entry and review.
- Control visibility of related lists (e.g., prescriptions under patients, medicines under orders).
- Provide quick actions to streamline frequent tasks like creating prescriptions, marking deliveries, or updating payments.
- Ensure different user profiles (Pharmacists, Delivery Agents, Admins) have layouts tailored to their responsibilities

Key Page Layouts Designed:

1. Patient Layout

- **Sections:** Patient Info, Contact Details, Medical History.
- **Related Lists:** Prescriptions, Deliveries.
- **Quick Actions:** New Prescription, Send Email.

2. Prescription Layout

- **Sections:** Prescription Details, Medication Info.
- **Related Lists:** Deliveries, Notes & Attachments.
- **Quick Actions:** Approve Prescription, Create Delivery.

3. Medicine Layout

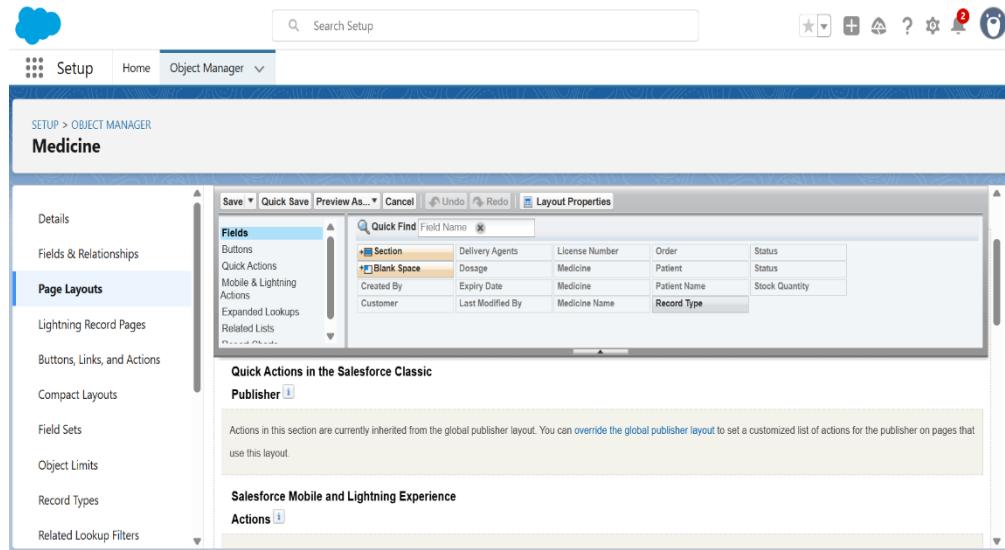
- **Sections:** Medicine Info, Stock & Inventory, Compliance & Safety.
- **Related Lists:** Prescriptions, Deliveries, Notes.
- **Quick Actions:** Update Stock, Adjust Price.

4. Order Layout

- **Sections:** Order Info, Customer Details, Prescription/Medicines, Delivery Info.
- **Related Lists:** Order Line Items, Payments, Deliveries.
- **Quick Actions:** Create Delivery, Update Payment, Cancel Order, Send Invoice.

The screenshot shows the Salesforce Setup interface for the 'Object Manager' under 'SETUP > OBJECT MANAGER'. The left sidebar is expanded, showing various configuration options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The 'Page Layouts' section is currently selected. In the main content area, the 'Fields' section is displayed, showing a table with columns for 'Section' and 'Field Name'. The table contains several rows, including 'Section' (Delivery Status, Order Status), 'Blank Space' (Last Modified By, Owner), 'Created By' (Order Date, Payment Status), 'Delivery Method' (Order Name, Record Type), and a 'Section' row for 'Information' (Header visible on edit only) containing fields like Order Name, Delivery Status, Order Date, Order Status, Payment Status, and Delivery Method, all set to 'Sample Text'. Below this is a 'System Information' section with similar sample data. The top navigation bar includes 'Save', 'Quick Save', 'Preview As...', 'Cancel', 'Undo', 'Redo', and 'Layout Properties'.

The screenshot shows the Salesforce Setup interface for the 'Object Manager' under 'SETUP > OBJECT MANAGER'. The left sidebar is expanded, showing various configuration options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The 'Page Layouts' section is currently selected. In the main content area, the 'Prescription Layout' section is displayed, showing a table with columns for 'Section' and 'Field Name'. The table contains several rows, including 'Section' (Owner, Record Type), 'Blank Space' (Prescription End ..., Prescription Name, Prescription Status), 'Created By' (Prescription Name, Prescription Status), and 'Last Modified By' (Prescription Status). Below this is a 'Prescription Sample' section, a 'Highlights Panel' (with a note to 'Customize the highlights panel for this page layout...'), and a 'Quick Actions in the Salesforce Classic' section. The top navigation bar includes 'Save', 'Quick Save', 'Preview As...', 'Cancel', 'Undo', 'Redo', and 'Layout Properties'. A toolbar at the top right includes icons for Custom Console Components, Mini Page Layout, Mini Console View, Video Tutorial, Help for this Page, and a question mark.



3.4 Compact Layouts:

Compact Layouts in Pharmacy Delivery CRM:

Compact Layouts in Salesforce determine which fields appear in the Highlights Panel of Lightning record pages and in record previews on the Salesforce Mobile App. They allow users to quickly view the most important details of a record without scrolling through the full page layout.

In the Pharmacy Delivery CRM, Compact Layouts are defined for key objects such as Order, Medicine, Delivery, and Prescription to improve usability and ensure critical information is visible at a glance.

1. Order Compact Layout

The Order compact layout provides delivery agents, pharmacists, and support staff with the essential status of an order.

Fields included:

- Order ID (Auto Number)
- Order Status (Picklist: Pending, In-Progress, Completed, Cancelled)
- Patient Name (Lookup → Patient)
- Total Amount (Currency)
- Estimated Delivery Date (Date)

Purpose: Helps staff instantly identify the order, track its progress, and check delivery timelines.

2. Medicine Compact Layout

The Medicine compact layout gives pharmacists quick access to inventory and compliance information.

Fields included:

- Medicine Name (Text)
- Stock Quantity (Number)
- Expiry Date (Date)
- Price (Currency)
- Medicine Type (Picklist: OTC, Prescription Required, Controlled)

Purpose: Enables fast validation of stock levels, expiry checks, and sales readiness.

3. Delivery Compact Layout

The Delivery compact layout helps delivery agents and managers track logistics in real time.

Fields included:

- Delivery ID (Auto Number)
- Delivery Status (Picklist: Assigned, In-Transit, Delivered, Returned)
- Assigned Agent (Lookup → User)
- Delivery Date (Date/Time)
- Tracking ID (Text)

Purpose: Provides an immediate overview of delivery progress, assigned personnel, and tracking details.

4. Prescription Compact Layout

The Prescription compact layout highlights key medical and status information for pharmacists and patients.

Fields included:

- Prescription ID (Auto Number)
- Patient Name (Lookup → Patient)
- Medicine Prescribed (Lookup → Medicine)
- Prescription Status (Picklist: Active, Expired, Cancelled)
- Start Date (Date)

Purpose: Ensures pharmacists can quickly verify prescription validity, patient details, and medicine information before processing orders.

Setup Home Object Manager

SETUP > OBJECT MANAGER
Prescription

Prescription Compact Layout
Prescription ID
« Back to Prescription

Help for this Page ?

Compact Layout Detail

Label	Prescription ID	Object Name	Prescription
API Name	Prescription_ID		
Included Fields	Prescription Name, Prescription Status, Prescription End Date, Record Type		

Created By Brunda Sanjana, 01/10/2025, 11:41 am Modified By Brunda Sanjana, 01/10/2025, 11:41 am

Help for this Page ?

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters

Setup Home Object Manager

SETUP > OBJECT MANAGER
Delivery

Delivery Compact Layout
Delivery Compact Layout
« Back to Delivery

Help for this Page ?

Compact Layout Detail

Label	Delivery Compact Layout	Object Name	Delivery
API Name	Delivery_Compact_Layout		
Included Fields	Delivery Name, Owner		

Created By Brunda Sanjana, 01/10/2025, 12:05 pm Modified By Brunda Sanjana, 01/10/2025, 12:05 pm

Help for this Page ?

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters

Setup Home Object Manager

SETUP > OBJECT MANAGER
Medicine

Medicine Compact Layout
Medicine Compact
« Back to Medicine

Help for this Page ?

Compact Layout Detail

Label	Medicine Compact	Object Name	Medicine
API Name	Medicine_Compact		
Included Fields	Medicine Name, Record Name, Record Type, Status, Dosage, Customer		

Created By Brunda Sanjana, 01/10/2025, 12:01 pm Modified By Brunda Sanjana, 01/10/2025, 12:01 pm

Help for this Page ?

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters

The screenshot shows the Salesforce Setup interface with the following details:

- Header:** Includes the Salesforce logo, a search bar labeled "Search Setup", and various navigation icons.
- Breadcrumbs:** "SETUP > OBJECT MANAGER Order".
- Left Sidebar (Compact Layouts):** A list of options including Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, and Compact Layouts (which is selected).
- Central Content:**
 - Section Header:** "Order Compact Layout Order compact layout < Back to Order".
 - Compact Layout Detail:** A table with the following data:

Label	Order compact layout	Object Name	Order
API Name	Order_compact_layout		
Included Fields	Order Name, Order Date, Order Status, Payment Status		
 - Buttons:** Edit, Clone, Delete, Compact Layout Assignment.
 - Timestamps:** Created By: Brunda Sanjana, 01/10/2025, 12:03 pm; Modified By: Brunda Sanjana, 01/10/2025, 12:03 pm.
 - Actions:** Edit, Clone, Delete, Compact Layout Assignment.

3.5 Schema Builder:

The Delivery object in Pharmacy Delivery CRM represents the tracking and management of medicine deliveries to patients. It links orders, delivery agents, and patients to ensure timely fulfilment of prescriptions and accurate record-keeping.

Purpose:

- Track the status of medicine deliveries from the pharmacy to the patient.
- Assign delivery agents to specific orders and monitor delivery progress.
- Maintain a history of completed, pending, or failed deliveries for reporting and auditing.

Relationships:

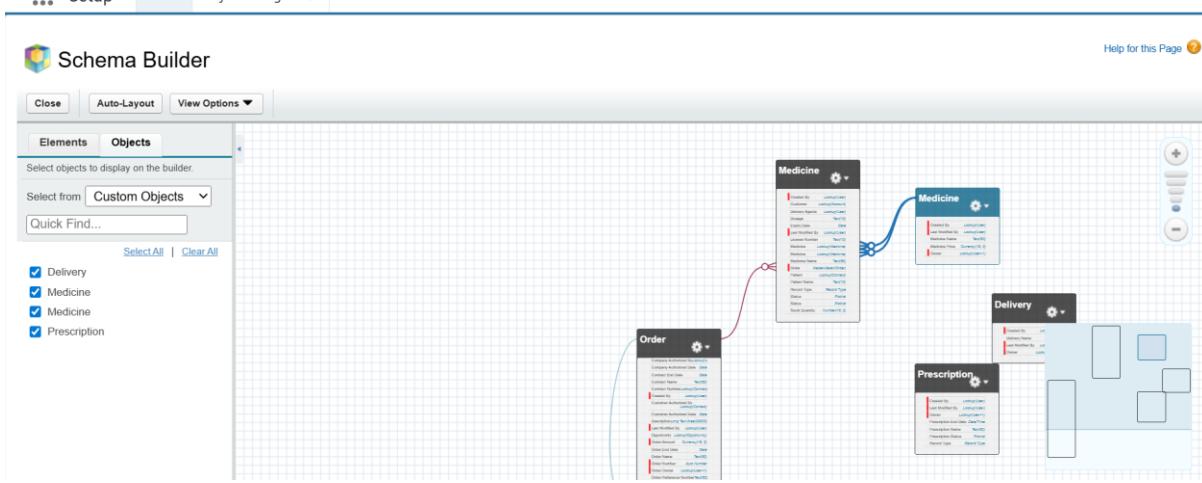
- Delivery → Order (Lookup): Each delivery is linked to a specific order to track which items are being delivered.
- Delivery → Delivery Agent (Lookup): Assigns a responsible user or agent to carry out the delivery.
- Order → Delivery (Related List): Allows orders to show all associated deliveries for tracking purposes.

Related Lists:

- Orders
- Prescriptions (via linked orders)
- Notes & Attachments (for delivery-related documents or receipts)

Benefits:

- Provides a centralized view of all deliveries in the pharmacy system.
- Helps monitor delivery performance, identify bottlenecks, and ensure customer satisfaction.
- Supports automation and reporting: e.g., generate reports for pending deliveries, completed deliveries, and agent performance.
- Facilitates integration with mobile apps or tracking systems for delivery updates.



3.6 Lookup vs Master-Detail vs Hierarchical Relationships:

Implementing Relationships in Pharmacy Delivery CRM

In Pharmacy Delivery CRM, relationships between objects define how data is connected and ensure proper workflow, reporting, and data integrity. The main relationship types used are Lookup, Master-Detail, and Hierarchical Relationships.

1. Lookup Relationship

A Lookup Relationship creates a flexible connection between two objects where the child record can exist independently of the parent. Deleting the parent does not delete the child. Lookup relationships are ideal for references or optional links between objects.

Implementation in Pharmacy Delivery CRM:

- **Prescription → Medicine:** Each prescription references a medicine. Prescriptions remain even if the medicine record is deleted.
- **Order → Prescription:** Orders can link to prescriptions but may exist independently (for OTC orders).
- **Delivery → Order:** Each delivery references an order, but an order can exist without a delivery.

2. Master-Detail Relationship

A Master-Detail Relationship creates a strong dependency between a parent and child. The child record cannot exist without the parent, and deleting the parent deletes the child. Child records inherit security and sharing settings from the parent. Master-Detail relationships are used when strict ownership and roll-up summaries are needed.

Implementation in Pharmacy Delivery CRM:

- **Prescription Medication → Prescription:** Medications are linked to prescriptions; deleting a prescription deletes its medications.

3. Hierarchical Relationship

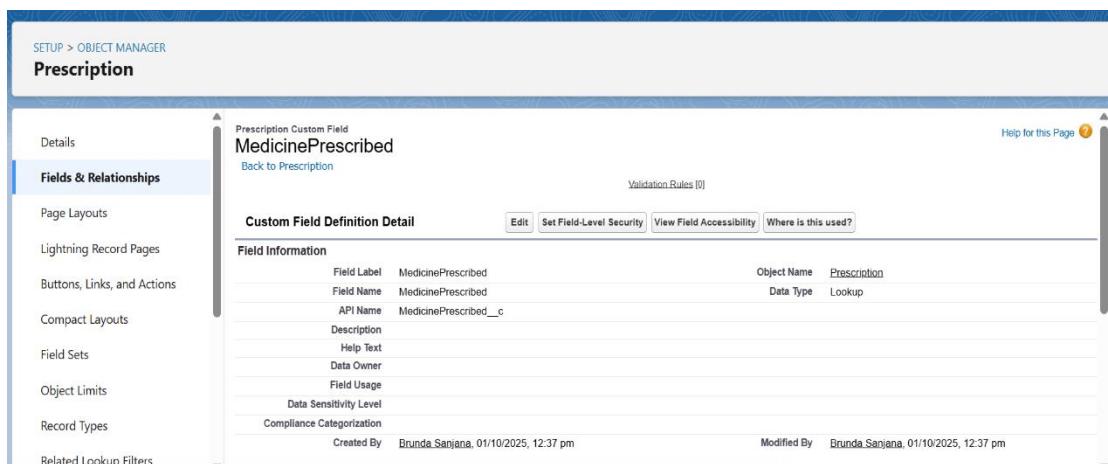
A Hierarchical Relationship is a special type of relationship available only on the User object. It links a user to another user to define reporting or managerial structures.

Implementation in Pharmacy Delivery CRM:

- User → Manager (User): Defines the reporting hierarchy of pharmacists, delivery agents, and supervisors

Benefits of Implementing Relationships

- Lookup: Provides flexible linking between objects where dependency is optional.
- Master-Detail: Enforces parent-child ownership and supports roll-up summaries.
- Hierarchical: Supports user reporting hierarchy, approvals, and access management.
- Ensures data integrity, workflow automation, and accurate reporting in Pharmacy Delivery CRM.



Conclusion: Phase 3 successfully strengthened the CRM's data architecture and integration capabilities, ensuring that all pharmacy delivery operations—from prescription management to order fulfillment and delivery tracking—are accurate, efficient, and scalable. This phase laid a solid foundation for further automation, advanced reporting, and seamless user experiences in subsequent phases.

Phase 4: Process Automation (Admin):

Introduction:

Phase 4 of the Pharmacy Delivery CRM focuses on automation, validation, and streamlined business processes to improve efficiency and accuracy. Key components include Validation Rules to enforce data integrity and Workflow Rules, Process Builder, and Flow Builder to automate tasks and approvals. The phase also incorporates Approval Processes for prescription and order validation, along with Email Alerts, Field Updates, Tasks, and Custom Notifications to keep users informed. By implementing these tools, the system minimizes manual work, reduces errors, and ensures timely communication. Overall, Phase 4 enhances operational efficiency and supports a more intelligent, responsive pharmacy delivery system.

Objectives:

1. Implement Validation Rules to ensure accurate and consistent data entry across all objects.
2. Automate routine tasks using Workflow Rules, Process Builder, and Flow Builder to reduce manual effort.
3. Configure Approval Processes for prescriptions, orders, and deliveries to maintain compliance and accountability.
4. Enable Email Alerts, Tasks, and Custom Notifications to keep pharmacists, delivery agents, and support staff informed in real time.
5. Use Field Updates and automated actions to maintain up-to-date records and improve operational efficiency.

Key Aspects :

1. **Validation Rules:** Enforce data accuracy and prevent incorrect or incomplete entries.
2. **Workflow Rules:** Automate standard processes like notifications, field updates, and task assignments.
3. **Process Builder:** Manage multi-step automation and record updates across related objects.
4. **Flow Builder:** Create Screen, Record-Triggered, Scheduled, and Auto-Launched Flows for complex business logic.
5. **Approval Processes:** Streamline prescription, order, and delivery approvals with defined hierarchies.

4.1 Validation Rules in Pharmacy Delivery CRM

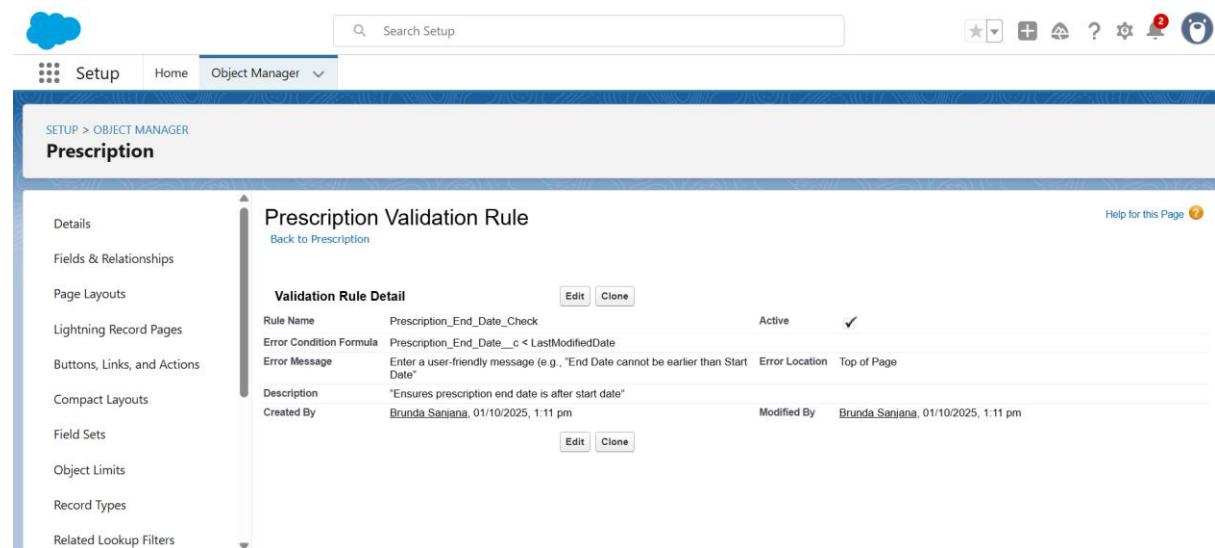
Definition:

Validation Rules in Salesforce are used to enforce data integrity by preventing users from entering incorrect, incomplete, or inconsistent information. They ensure that the data entered into the system meets defined business criteria before a record can be saved.

Purpose in Pharmacy Delivery CRM:

- Ensure accurate patient, prescription, and order data.
- Prevent errors in critical fields like Medicine Stock, Prescription Dates, and Delivery Status.
- Enforce compliance with pharmacy regulations and operational policies.
- Reduce manual checks and improve overall data quality

Validation Rules in Pharmacy Delivery CRM are essential for ensuring that all critical data entered is correct and consistent, supporting safe and efficient pharmacy operations while reducing risks of errors or non-compliance.



The screenshot shows the Salesforce Object Manager interface for a 'Prescription' object. The left sidebar lists various setup options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main content area is titled 'Prescription Validation Rule' and displays the 'Validation Rule Detail' for the rule named 'Prescription_End_Date_Check'. The rule is active and has an error message: 'Enter a user-friendly message (e.g., "End Date cannot be earlier than Start Date")'. The description is: 'Ensures prescription end date is after start date'. The rule was created by Brunda Sanjana on 01/10/2025 at 1:11 pm and modified by the same user on the same date and time. There are 'Edit' and 'Clone' buttons for the rule.

The screenshot shows the Salesforce Object Manager interface. At the top, there's a navigation bar with icons for Setup, Home, and Object Manager. A search bar says "Search Setup". On the right, there are various system status icons. Below the navigation is a header bar with "SETUP" and "Object Manager". The main content area is titled "Order Validation Rule" and has a sub-link "Back to Order Validation Rules". It displays a "Validation Rule Detail" section with the following fields:

Rule Name	Order_Quantity_Check	Active	<input checked="" type="checkbox"/>
Error Condition Formula	ISBLANK(Id)	Error Location	Top of Page
Error Message	"Order ID cannot be blank. Please enter a valid Order ID."		
Description	Prevents saving an order if ordered quantity exceeds stock		
Created By	Brunda Sanjana, 01/10/2025, 2:56 pm	Modified By	Brunda Sanjana, 01/10/2025, 2:56 pm

At the bottom of the detail section are "Edit" and "Clone" buttons.

4.2 Approval Process:

Definition:

An Approval Process in Salesforce is an automated sequence that allows records to be reviewed and approved by designated users based on predefined criteria. It ensures that critical business actions comply with organizational policies before being finalized.

Purpose in Pharmacy Delivery CRM:

- Maintain compliance and accountability for critical operations such as prescriptions, orders, and deliveries.
- Streamline review and approval workflows for pharmacists, supervisors, and managers.
- Reduce manual intervention and ensure timely decision-making.
- Provide a clear audit trail of approvals and rejections for regulatory purposes.

Key Components of an Approval Process:

1. Entry Criteria:

- Defines which records enter the approval process.
- Example: Only prescriptions marked as “Ready for Approval” or orders above a certain amount.

2. Approvers:

- Specify **users, roles, or queues** responsible for approving records.
- Example: Senior pharmacist approves high-value orders or controlled prescriptions.

3. Approval Steps:

- Multi-step approvals can be configured for hierarchical approvals.

- Example: Prescription approved first by a pharmacist, then by a supervisor.

Example Use Cases in Pharmacy Delivery CRM:

- **Prescription Approval:** Ensures prescriptions are verified by a pharmacist before being processed.
- **High-Value Orders Approval:** Orders above a certain monetary threshold require supervisor approval before fulfillment.
- **Medicine Stock Update:** Changes to critical medicine stock quantities require managerial approval.

The screenshot shows the Salesforce Setup interface with the following details:

- Left Navigation:** Shows sections like Data, Feature Settings, Process Automation, and the selected **Approval Processes**.
- Search Bar:** Displays "Search Setup" and a search term "appro".
- Header:** Shows "SETUP" and the title "Approval Processes".
- Page Content:**
 - Process Definition Detail:**

Process Name	Prescription_Approval_Process	Active	<input type="checkbox"/>
Unique Name	Prescription_Approval_Process	Next Automated Approver Determined By	Manager or Record Owner
Description	"Approval workflow for verifying prescriptions before processing."		
Entry Criteria	MedicinePrescribed__c = "Ready for Approval"		
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests	<input type="checkbox"/>
Approval Assignment Email Template	Sales_New_Customer_Email		
Initial Submitters	Prescription Owner		
Created By	Brunda Sanjana, 01/10/2025, 3:23 pm	Modified By	Brunda Sanjana, 01/10/2025, 3:29 pm
 - Help and Buttons:** Includes "Help for this Page" and standard edit, clone, delete, and activate buttons.

The screenshot shows the detailed configuration of the approval process:

- Action Type:** Record Lock
- Description:** Lock the record from being edited
- Approval Steps:**

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit Del	1	Pharmacist_Review	This step requires a senior pharmacist to review and approve prescriptions marked as "Ready for Approval."		Manager	Final Rejection
- Final Approval Actions:**

Action	Type	Description
Edit	Record Lock	Lock the record from being edited
- Final Rejection Actions:**

Action	Type	Description
Edit	Record Lock	Unlock the record for editing

4.3 Flow Builder in Pharmacy Delivery CRM

Definition:

Flow Builder is a powerful automation tool in Salesforce that allows you to create complex business processes without writing code. It automates tasks, updates records, sends notifications, and guides users through multi-step processes using a visual interface.

Purpose in Pharmacy Delivery CRM:

- Automate prescription approvals, order processing, and delivery tracking.
- Replace repetitive manual tasks with dynamic, automated workflows.
- Ensure data accuracy and consistency across objects like Prescriptions, Orders, Patients, Deliveries, and Medicines.

Types of Flows:

1. **Screen Flows:**
 - Interactive flows that guide users through screens to collect or update information.
 - Example: A pharmacist enters prescription verification details step by step.
2. **Record-Triggered Flows:**
 - Automatically run when a record is created, updated, or deleted.
 - Example: Update delivery status and notify the patient when an order record is updated to “Dispatched.”
3. **Scheduled Flows:**
 - Run at specific times or intervals.
 - Example: Daily check of medicine stock and send low-stock alerts to pharmacy staff.
4. **Autolaunched Flows:**
 - Triggered automatically via Process Builder, Apex, or other flows.
 - Example: Automatically assign a delivery agent when a prescription is approved.

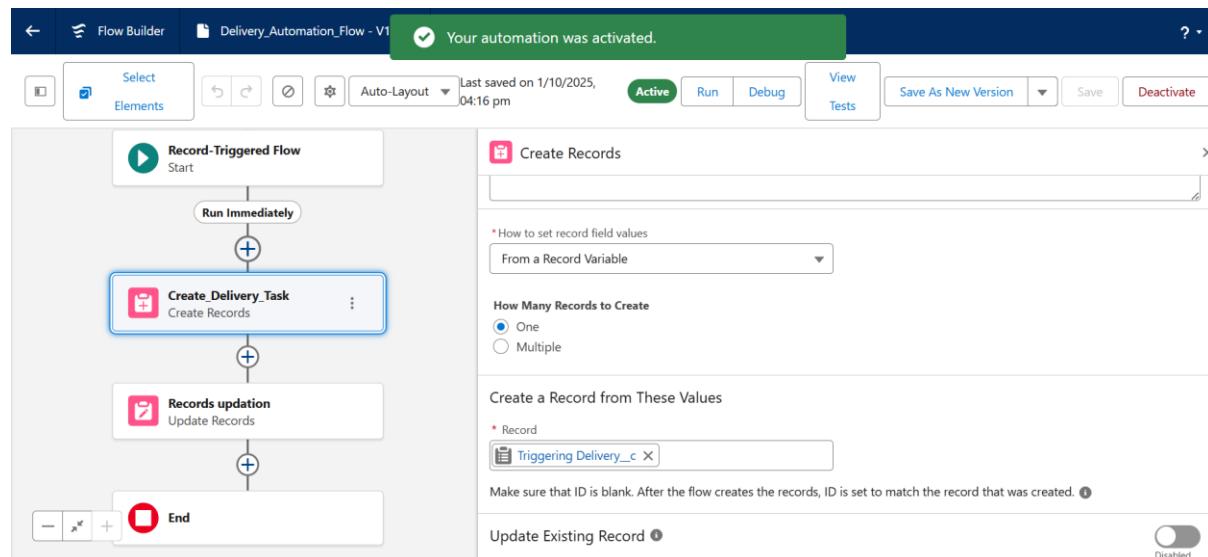
Key Components of a Flow

- **Elements:** Actions, decisions, assignments, loops, and screens that define the flow logic.
- **Resources:** Variables, collections, constants, formulas used within the flow.
- **Connectors:** Define the sequence of steps between elements.

- **Start:** Defines how the flow begins (manual, record-triggered, scheduled).
- **Debug & Test:** Validate that the flow behaves as expected before activating.

Example Use Cases in Pharmacy Delivery CRM

Flow Type	Use Case
Screen Flow	Pharmacist verifies prescription details step by step before approval.
Record-Triggered Flow	Automatically update delivery status when order is marked “Dispatched.”
Scheduled Flow	Send daily low-stock alerts for medicines below threshold.
Autolaunched Flow	Assign delivery agents automatically when prescription is approved.



4.4 Email Alerts in Pharmacy Delivery CRM:

Purpose:

The purpose of setting up Email Alerts is to reduce manual follow-ups, provide real-time status updates, and improve customer satisfaction by ensuring that all stakeholders are informed about delivery progress.

Configuration Steps:

1. **Create Email Templates:** Define standardized templates (HTML/Text) with placeholders for delivery details (Order ID, Customer Name, Address, Scheduled Time).
2. **Set Up Email Alerts:** Define alerts with chosen templates, Assign recipients (Delivery Agent, Manager, or Customer)

Classic Email Templates

Delivery_Notification

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Delivery_Notification
Template Unique Name	Delivery_Notification
Encoding	Unicode (UTF-8)
Author	Brunda Sanjana [Change]
Description	
Created By	Brunda Sanjana, 01/10/2025, 4:30 pm
Modified By	Brunda Sanjana, 01/10/2025, 4:30 pm

Email Alert

Alert for Delivery Assignment

Email Alert Detail

Description	Alert for Delivery Assignment
Unique Name	Alert_for_Delivery_Assignment
From Email Address	Current User's email address
Recipients	User_Brunda Sanjana
Additional Emails	
Created By	Brunda Sanjana, 01/10/2025, 4:35 pm
Modified By	Brunda Sanjana, 01/10/2025, 4:35 pm

Key Outcomes:

1. End-to-End Automation

- Automated delivery workflows (assignment, dispatch, tracking, and completion).
- Email Alerts and Notifications configured for customers, delivery agents, and managers.
- Reduced manual intervention through Process Builder and Flow automation.

2. Enhanced Reporting & Dashboards

- Real-time dashboards for delivery performance, customer satisfaction, and order fulfillment rates.
- Custom reports on delivery times, agent productivity, and delayed orders.
- Data-driven decision-making for pharmacy managers.
- Higher customer trust and satisfaction.

5 Challenges and Resolutions:

1. Automation Complexity

- **Challenge:**

Configuring multiple Flows, Email Alerts, and Process Automations led to overlaps and execution errors.

- **Resolution:**

Conducted a flow consolidation exercise, replaced redundant workflows with Record-Triggered Flows, and implemented error handling paths for failed automation.

2. Email Deliverability Issues

- **Challenge:**

Email Alerts were sometimes blocked by spam filters or failed due to incorrect customer email addresses.

- **Resolution:**

- Configured Organization-Wide Email Addresses and Email Relays.
- Implemented email validation rules at data entry.
- Used standard Salesforce templates with trusted domains.

Conclusion:

Phase 4 of the Pharmacy Delivery CRM implementation focuses on advanced automation, reporting, and optimization to ensure the system delivers measurable business value. By the end of this phase, the CRM is fully integrated into pharmacy operations, with improved efficiency, compliance, and customer engagement.

Phase 5: Apex Programming (Developer):

Introduction:

Phase 5 focuses on extending the Pharmacy Delivery CRM with Apex programming for advanced automation and customization.

Developers implement classes, triggers, and design patterns to handle complex business logic beyond point-and-click tools.

Key features include SOQL/SOSL queries, collections, control statements, and exception handling.

Key Objectives:

1. Custom Business Logic Implementation

- Develop Apex Classes & Objects to handle pharmacy-specific delivery workflows that cannot be achieved using declarative tools.

2. Trigger-Based Automation

- Implement Apex Triggers (before/after insert, update, delete) to enforce business rules such as order validation, delivery assignment, and stock updates.

3. Efficient Data Handling

- Use SOQL & SOSL queries along with Collections (List, Set, Map) for optimized data retrieval and processing.

4. Asynchronous Processing

- Introduce Batch Apex, Queueable, Scheduled Apex, and Future Methods to manage high-volume data, background jobs, and scheduled notifications.

Detailed Activities Performed

5.1 Classes & Objects:

Use Case in Pharmacy Delivery CRM

In your CRM, you'll have key business entities:

- **Customer** (who orders medicines)
- **Medicine** (the product being delivered)
- **Delivery** (the process of dispatching orders)
- **Delivery Agent** (who delivers medicines)

a) **Customer Class**

```
public class Customer {  
    public String customerName;  
    public String phone;
```

```
public String address;

// Constructor
public Customer(String name, String phoneNumber, String addr) {
    customerName = name;
    phone = phoneNumber;
    address = addr;
}

// Method
public void displayDetails() {
    System.debug('Customer: ' + customerName + ', Phone: ' + phone + ', Address: '
+ address);
}
```

b) Medicine Class

```
public class Medicine {
    public String medName;
    public Decimal price;
    public Integer quantity;

    // Constructor
    public Medicine(String name, Decimal medPrice, Integer qty) {
        medName = name;
        price = medPrice;
        quantity = qty;
    }

    // Method
    public Decimal calculateTotal() {
        return price * quantity;
    }
}
```

c) Delivery Class

```
public class Delivery {
    public Customer cust;
    public List<Medicine> medicines;
    public String deliveryStatus;

    // Constructor
}
```

```

public Delivery(Customer c, List<Medicine> meds) {
    cust = c;
    medicines = meds;
    deliveryStatus = 'Pending';
}

// Method

public Decimal calculateOrderAmount() {
    Decimal total = 0;
    for(Medicine m : medicines) {
        total += m.calculateTotal();
    }
    return total;
}

public void markAsDelivered() {
    deliveryStatus = 'Delivered';
}

```

5.2 Apex Triggers (before/after insert/update/delete):

Definition:

A trigger is an Apex script that executes automatically when a record is inserted, updated, deleted, or undeleted.

- **Before Insert:** Validate a medicine order (e.g., quantity > 0).
- **Before Update:** Prevent delivery status from being changed directly by unauthorized users.
- **After Insert:** Automatically create a delivery task once an order is placed.
- **After Update:** Send notification email/SMS when delivery status changes to "Delivered".
- **After Delete:** Log canceled deliveries for audit.

Example Trigger – Auto-create a Delivery Task

```
trigger DeliveryTrigger on Delivery__c (after insert) {  
    List<Task> tasks = new List<Task>();  
  
    for(Delivery__c d : Trigger.new) {  
        Task t = new Task(  
            Subject = 'Deliver Medicines',  
            WhatId = d.Id,  
            OwnerId = d.Assigned_Agent__c,  
            Status = 'Not Started'  
        );  
        tasks.add(t);  
    }  
  
    if(!tasks.isEmpty()) {  
        insert tasks;  
    }  
}
```

2. Trigger Design Pattern

Problem: If you write all logic in the trigger, it becomes bulky and hard to maintain.

Solution: Use a **Handler Class** (Trigger Design Pattern).

Structure:

- **Trigger File:** Only event declarations.
- **Handler Class:** Contains logic.

Example

Trigger File (DeliveryTrigger.trigger):

```
trigger DeliveryTrigger on Delivery__c (before insert, after insert, after update) {  
    DeliveryTriggerHandler handler = new DeliveryTriggerHandler();
```

```

if(Trigger.isBefore && Trigger.isInsert){
    handler.beforeInsert(Trigger.new);
}

if(Trigger.isAfter && Trigger.isInsert){
    handler.afterInsert(Trigger.new);
}

if(Trigger.isAfter && Trigger.isUpdate){
    handler.afterUpdate(Trigger.new, Trigger.oldMap);
}

```

Handler Class (DeliveryTriggerHandler.cls):

```

public class DeliveryTriggerHandler {

    public void beforeInsert(List<Delivery__c> newList){
        for(Delivery__c d : newList){
            if(d.Quantity__c <= 0){
                daddError('Medicine quantity must be greater than 0.');
            }
        }
    }

    public void afterInsert(List<Delivery__c> newList){
        List<Task> tasks = new List<Task>();
        for(Delivery__c d : newList){
            Task t = new Task(
                Subject = 'Deliver Medicines',
                WhatId = d.Id,
                Status = 'Not Started'
            );
        }
    }
}
```

```

        tasks.add(t);
    }
    if(!tasks.isEmpty()) insert tasks;
}

public void afterUpdate(List<Delivery__c> newList, Map<Id, Delivery__c> oldMap){
    for(Delivery__c d : newList){
        if(d.Status__c == 'Delivered' && oldMap.get(d.Id).Status__c != 'Delivered'){
            System.debug('Delivery completed for Order: ' + d.Id);
            // Can add email alert logic here
        }
    }
}
}

```

This keeps triggers clean, reusable, and easier to maintain.

5.3 SOQL (Salesforce Object Query Language):

Used to retrieve records from Salesforce database.

Use Cases in Pharmacy Delivery CRM

- Get all pending deliveries for a delivery agent.
- Find all medicines below stock threshold.
- Get total orders by a customer.

Examples

```

// Get all pending deliveries for Agent
List<Delivery__c> pendingDeliveries = [
    SELECT Id, Customer__c, Status__c
    FROM Delivery__c
    WHERE Status__c = 'Pending' AND Assigned_Agent__c = :UserInfo.getUserId()
];

```

```
// Get all medicines with low stock  
List<Medicine__c> lowStock = [  
    SELECT Name, Stock__c  
    FROM Medicine__c  
    WHERE Stock__c < 10  
];
```

5.4 SOSL (Salesforce Object Search Language):

Used to perform text-based search across multiple objects.

Use Cases in Pharmacy Delivery CRM

- Search a medicine by name across product catalog.
- Search customer by phone number or name.

Example

```
// Search customer or medicine by keyword  
List<List<SObject>> searchList = [  
    FIND 'Paracetamol' IN ALL FIELDS  
    RETURNING  
        Customer__c(Id, Name, Phone__c),  
        Medicine__c(Id, Name, Stock__c)  
];
```

```
List<Customer__c> customers = (List<Customer__c>) searchList[0];
```

```
List<Medicine__c> medicines = (List<Medicine__c>) searchList[1];
```

Summary:

- **Triggers** → Automate business logic (before/after events).
- **Trigger Design Pattern** → Clean, reusable trigger logic.
- **SOQL** → Query structured records.
- **SOSL** → Search text across objects.

5.5 Collections:

Salesforce Apex has three main collection types:

a) List

- Ordered collection, allows duplicates.
- Use case: Store all medicines in a single delivery.

```
List<Medicine__c> medicines = new List<Medicine__c>();  
medicines.add(new Medicine__c(Name='Paracetamol', Stock__c=50));  
medicines.add(new Medicine__c(Name='Vitamin C', Stock__c=30));
```

```
System.debug('Medicines in Delivery: ' + medicines);
```

b) Set

- Unordered collection, no duplicates.
- Use case: Store unique customer IDs who placed orders today.

```
Set<Id> uniqueCustomers = new Set<Id>();  
uniqueCustomers.add('0035g00000XXXXXX'); // customer Id  
uniqueCustomers.add('0035g00000YYYYYY');  
System.debug('Unique Customers: ' + uniqueCustomers);
```

c) Map

- Key-value pairs, fast lookup.
- Use case: Map each delivery ID to its assigned delivery agent.

```
Map<Id, Id> deliveryAgentMap = new Map<Id, Id>();  
deliveryAgentMap.put('a0B5g00000XXXXXX', '0055g00000AAAAA'); // DeliveryId ->  
AgentId
```

```
System.debug('Assigned Agent: ' + deliveryAgentMap.get('a0B5g00000XXXXXX'));
```

5.6 Control Statements:

Control Statements direct the flow of code.

Examples in Pharmacy Delivery CRM:

```
List<Delivery__c> deliveries = [SELECT Id, Status__c FROM Delivery__c];
```

```

for(Delivery__c d : deliveries){ // for loop
    if(d.Status__c == 'Pending'){ // if statement
        System.debug('Pending Delivery: ' + d.Id);
    } else if(d.Status__c == 'Delivered'){ // else if
        System.debug('Already Delivered: ' + d.Id);
    } else {
        System.debug('Other Status: ' + d.Id);
    }
}

```

```

Integer count = 0;
while(count < deliveries.size()){ // while loop
    System.debug('Delivery #' + count + ': ' + deliveries[count].Id);
    count++;
}

```

5.6 Batch Apex:

Purpose: Process large data sets asynchronously (more than 50,000 records).

Use Case:

Send notifications for all pending deliveries at the end of the day.

Example:

```

global class PendingDeliveryBatch implements Database.Batchable<SObject> {

    global Database.QueryLocator start(Database.BatchableContext BC){
        return Database.getQueryLocator(
            'SELECT Id, Customer__c, Status__c FROM Delivery__c WHERE Status__c = '
            '\'Pending\''
        );
    }
}

```

```
global void execute(Database.BatchableContext BC, List<Delivery__c> scope){  
    for(Delivery__c d : scope){  
        System.debug('Notify Customer for Delivery: ' + d.Id);  
        // Add email/SMS logic  
    }  
}
```

```
global void finish(Database.BatchableContext BC){  
    System.debug('Batch Process Completed.');//  
}
```

Execute Batch:

```
PendingDeliveryBatch batch = new PendingDeliveryBatch();  
Database.executeBatch(batch, 200); // 200 records per batch
```

5.7 Queueable Apex:

- Asynchronous, like Batch Apex, but lightweight.
- Use case: Process a single large delivery order in the background.

```
public class DeliveryQueueable implements Queueable {  
    private Id deliveryId;
```

```
    public DeliveryQueueable(Id dId){  
        deliveryId = dId;  
    }
```

```
    public void execute(QueueableContext context){  
        Delivery__c d = [SELECT Id, Status__c FROM Delivery__c WHERE Id = :deliveryId];  
        d.Status__c = 'Delivered';  
        update d;
```

```

        System.debug('Delivery marked delivered for: ' + deliveryId);
    }
}

```

Enqueue Job:

```
ID jobId = System.enqueueJob(new DeliveryQueueable('a0B5g00000XXXXX'));
```

5.8 Scheduled Apex:

- Automatically runs Apex at a scheduled time.
- Use case: Send daily summary of deliveries at 8 PM.

```

global class DailyDeliverySummary implements Schedulable {
    global void execute(SchedulableContext sc){
        List<Delivery__c> deliveries = [SELECT Id, Status__c FROM Delivery__c];
        System.debug('Total Deliveries Today: ' + deliveries.size());
        // Add email logic
    }
}

```

```
// Schedule in Salesforce
```

```
String cronExp = '0 0 20 * * ?'; // Every day at 8 PM
```

```
System.schedule('Daily Delivery Summary', cronExp, new DailyDeliverySummary());
```

Summary:

Feature	Use Case
List	Store medicines in a delivery
Set	Track unique customers/orders
Map	Link delivery → delivery agent
Control Statements	Check delivery status, loop through orders
Batch Apex	Send notifications for all pending deliveries
Queueable Apex	Update large single deliveries asynchronously

```

1 * public class DeliveryHandler {
2 *     public static void sayHello() {
3 *         System.debug('Hello from Pharmacy Delivery CRM!');
4 *     }
5 }

```

Logs

User	Application	Operation	Time	Status	Read	Size
Brinda Sanjana	Unknown	ApexTestHandler	10/1/2025, 4:56:43 PM	Success	Unread	989 bytes

```

1 * public class DeliveryTriggerHandler {
2
3 *     public void beforeInsert(List<Delivery__c> newList){
4 *         for(Delivery__c d : newList){
5 *             if(d.quantity__c <= 0){
6 *                 d.addError('Medicine quantity must be greater than 0.');
7 *             }
8 *         }
9 *     }
10
11     public void afterInsert(List<Delivery__c> newList){
12         List<Task> tasks = new List<Task>();
13         for(Delivery__c d : newList){
14             Task t = new Task{

```

```

1 * trigger DeliveryTrigger on Delivery__c (before insert, after insert, after update) {
2     DeliveryTriggerHandler handler = new DeliveryTriggerHandler();
3
4     if(Trigger.isBefore && Trigger.isInsert){
5         handler.beforeInsert(Trigger.new);
6     }
7     if(Trigger.isAfter && Trigger.isInsert){
8         handler.afterInsert(Trigger.new);
9     }
10    if(Trigger.isAfter && Trigger.isUpdate){
11        handler.afterUpdate(Trigger.new, Trigger.oldMap);
12    }
13 }

```

Key Outcomes for Pharmacy Delivery CRM:

1. Apex Classes & Objects Implementation

- o Model real-world entities: Customer, Medicine, Delivery, Delivery Agent.
- o Handle attributes like order quantity, delivery status, and customer contact efficiently.

2. Automated Processes with Apex Triggers

- **Before Insert/Update:** Validate medicine orders, prevent invalid delivery status changes.
- **After Insert/Update:** Auto-create delivery tasks, notify delivery agents, update delivery status.

3. Trigger Design Pattern Application

- Separate trigger logic into **handler classes** for maintainability.
- Easily extend CRM functionality without cluttering triggers.

4. Data Management with SOQL & SOSL

- **SOQL:** Query pending deliveries, low-stock medicines, and customer orders.
- **SOSL:** Search across Customers and Medicines for faster retrieval.

6 Challenges and Resolutions:

1. Handling Large Volumes of Orders:

In a busy pharmacy, hundreds or thousands of orders may come in daily. Processing all these orders at once can exceed Salesforce governor limits, causing errors or delays.

Resolution: Implement Batch Apex to process orders in bulk and Queueable Apex for background processing. Use Collections (Lists, Sets, Maps) to efficiently manage multiple records without hitting limits.

2. Real-time Delivery Tracking:

Keeping delivery statuses updated in real-time is critical for both customers and delivery agents. Without automation, updates can be delayed or inconsistent.

Resolution: Use Apex Triggers (before/after update) to automatically update delivery statuses as changes occur. Scheduled Apex can periodically synchronize data between the CRM and delivery agents or external systems.

7 Conclusion:

Phase 5 focused on implementing Apex programming to automate and streamline pharmacy delivery operations.

By using Classes, Triggers, and Trigger Design Patterns, the CRM can handle orders, deliveries, and notifications efficiently.

SOQL and SOSL enabled fast retrieval and search of customer, medicine, and delivery data. Collections, Control Statements, and Asynchronous Apex (Batch, Queueable, Scheduled) improved scalability for high-volume operations.

Overall, Phase 5 made the Pharmacy Delivery CRM robust, automated, and capable of managing large-scale delivery workflows effectively.

Phase 6: User Interface Development:

Introduction:

Phase 6 focuses on enhancing the user interface and interactivity of the Pharmacy Delivery CRM using Salesforce Lightning and Lightning Web Components (LWC). The Lightning App Builder in Salesforce is a powerful tool used to design and customize the user interface using components. It supports building Record Pages, Home Page Layouts, and Tabs tailored for user needs. Developers can enhance functionality using Lightning Web Components (LWC), which interact with Apex via wire adapters or imperative calls. Events in LWC enable component communication, while the Navigation Service allows seamless page transitions within the app. The Utility Bar adds persistent tools at the bottom of the screen for improved accessibility.

Key Objectives – Pharmacy Delivery CRM:

1. Enhance User Interface with Lightning App Builder
 - Create intuitive apps for pharmacy staff, delivery agents, and administrators.
 - Organize functionality using custom record pages, tabs, and home page layouts.
2. Improve Navigation and Accessibility
 - Use Tabs and Utility Bar for quick access to orders, deliveries, inventory, and customer information.
 - Ensure important tools are readily available to increase productivity.
3. Develop Responsive and Interactive Components
 - Build Lightning Web Components (LWC) for dynamic features like live order trackers, medicine search widgets, and delivery dashboards.
4. Integrate Apex with LWC
 - Connect server-side logic to LWCs to fetch, update, and process CRM data in real-time.
 - Enable automation and business logic execution from the UI.

6.1 Lightning App Builder in Pharmacy Delivery CRM:

The Lightning App Builder in Salesforce is a point-and-click tool that allows administrators to design customized user interfaces without writing code. For a Pharmacy Delivery CRM, it helps create intuitive and efficient pages that improve the workflow for pharmacists, delivery agents, and customer service representatives.

Key Uses in Pharmacy Delivery CRM

1. Record Pages

- Customize the Delivery Object Page to show patient details, prescription information, delivery address, and agent assignment in one view.
- Configure Pharmacy Account Pages to include prescription history, billing details, and delivery preferences.

2. Home Pages

- Create a Pharmacist Home Page that displays pending prescription approvals, high-priority deliveries, and inventory alerts.
- Build a Delivery Agent Home Page with today's assigned deliveries, maps integration for route optimization, and customer contact details.

3. App Pages

- Create a Delivery Management App Page that combines components like active orders, delivery status dashboards, and quick actions for updating delivery stages.
- Provide customer service teams with an Order Support App Page that highlights escalated cases and communication history.

4. Tabs & Navigation

- Add custom tabs such as "Deliveries," "Prescriptions," "Customers," and "Pharmacy Partners" to make navigation simple for users.

5. Utility Bar

- Include utilities like Chat Support, Quick Notes, and Delivery Notifications so that agents can access them from anywhere in the app.

6. Integration with Lightning Web Components (LWC)

- Embed custom LWCs for Real-time Delivery Tracking, Prescription Verification, or Customer Signature Capture directly into record or app pages.

The screenshot shows the Lightning App Builder interface with the URL <https://wise-koala-f746j2-dev-ed.trailblaze.lightning.force.com/visualEditor/appBuilder.app?id=02uWU00000Iru9lYAB&retUrl=https%3A%2F%2Fwise-koala-f746j2-dev-ed.trailblaze.lightning.force.com/visualEditor/appBuilder.app>. The page title is "Pharmacy Delivery CRM". The left sidebar under "App Settings" has "App Details & Branding" selected. The main content area shows "App Details" and "App Branding" sections. In "App Details", fields include "App Name" (Pharmacy Delivery CRM), "Developer Name" (Pharmacy_Delivery_CRM), and "Description" (This app provides easier access related to pharmacy issues). In "App Branding", there is an "Image" preview, a color picker set to "#0070D2", and a checked checkbox for "Use the app's image and color instead of the org's custom theme".

6.2 Record Pages in Pharmacy Delivery CRM:

Record Pages in Salesforce are customized layouts built using the Lightning App Builder that display detailed information about a specific record (such as a Delivery, Prescription, Customer, or Pharmacy Account). They allow users to see all the related data, components, and actions in one place, improving productivity and user experience.

In a Pharmacy Delivery CRM, Record Pages play a vital role by providing role-specific views and contextual data for pharmacists, delivery agents, and support staff.

Key Features of Record Pages

1. Dynamic Layouts

- Record Pages can show different layouts depending on the profile (e.g., Pharmacist vs. Delivery Agent).
- Example: A Delivery Record Page shows route maps for delivery agents, but pharmacists see prescription verification details.

2. Components on Record Pages

- Record Details → Core details like order ID, medicine list, dosage instructions, delivery status.
- Related Lists → Linked prescriptions, customer details, billing info.
- Charts & Reports → Delivery performance metrics or stock usage.
- Custom Components (LWCs) → Live GPS tracking, digital signature, customer feedback form.

3. Quick Actions

- Agents can update delivery status (*Out for Delivery*, *Delivered*, *Failed Attempt*).
- Pharmacists can approve prescriptions or flag errors directly from the record page.

4. Contextual Information

- Delivery Record Page → Shows assigned delivery agent, route, and customer contact.
- Prescription Record Page → Shows prescribed medicines, dosage, and pharmacy fulfillment status.
- Customer Record Page → Displays customer's prescription history, preferred pharmacy, and delivery preferences.

6.3 Home Page Layouts in Pharmacy Delivery CRM:

Home Page Layouts in Salesforce allow administrators to design personalized landing pages for different users based on their role. They provide an at-a-glance view of the most important information, helping users work efficiently without navigating multiple records or reports.

Use in Pharmacy Delivery CRM

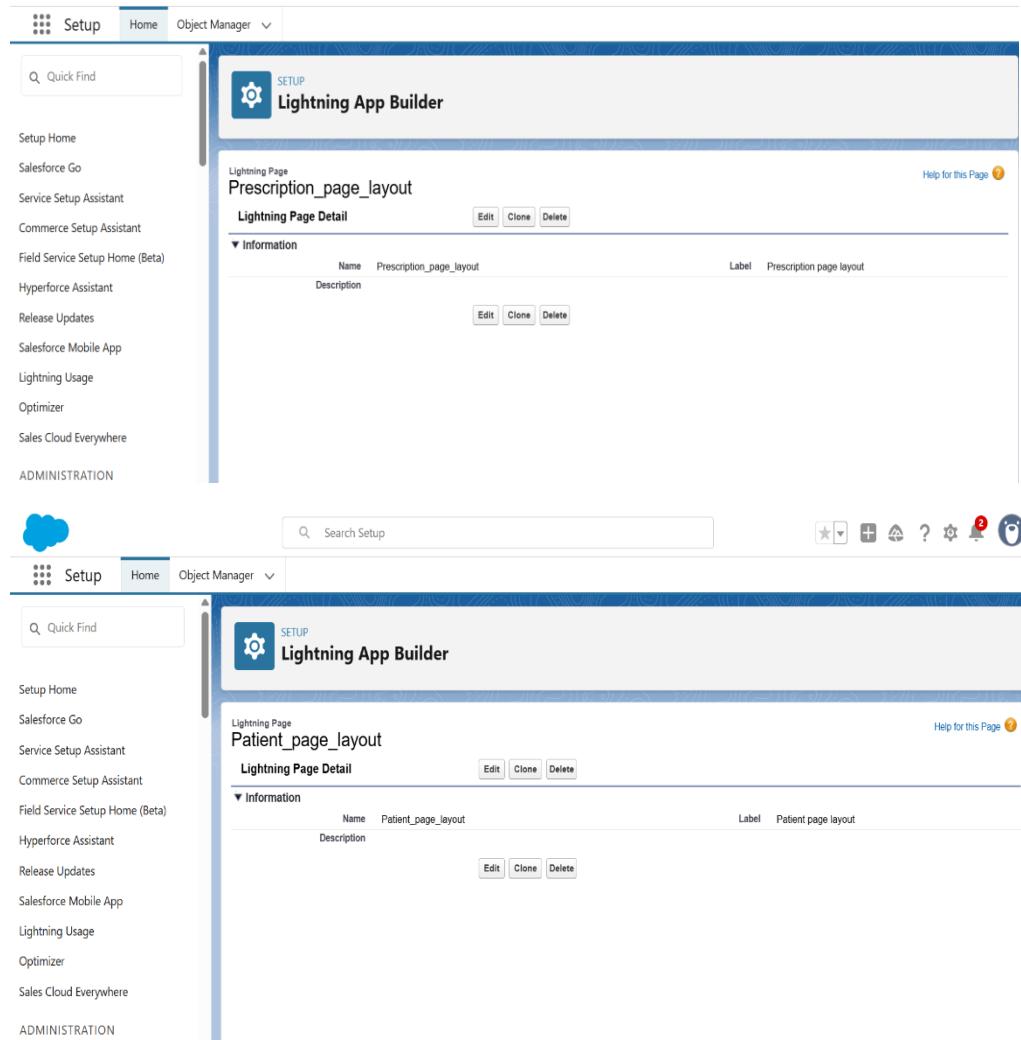
- Pharmacist Home Page
 - Pending prescription approvals.
 - Inventory alerts for medicines running low.
 - Quick links to patient records and pharmacy accounts.
 - Notifications for new customer orders.
- Delivery Agent Home Page
 - List of today's assigned deliveries.
 - Map integration with optimized delivery routes.
 - Quick action buttons to update delivery status (*Out for Delivery*, *Delivered*).
 - Alerts for urgent or high-priority deliveries.
- Admin/Manager Home Page
 - Dashboards showing KPIs (delivery performance, customer satisfaction, stock usage).
 - Reports on prescription fulfillment times.

- Shortcuts to user management and pharmacy partner accounts.

The screenshot shows the Lightning App Builder interface. The left sidebar has a search bar and categories: Lightning Usage, Email, Apps, Objects and Fields, User Interface, and Custom Code. Under User Interface, 'Lightning App Builder' is selected. The main area displays a page layout named 'Medicine_PageLayout'. It includes sections for 'Information' (Name: Medicine_PageLayout, Label: Medicine PageLayout), 'Assignments By App' (No Assignments to display), and 'Assignments By App, Record Type, and Profile' (No Assignments to display). There are 'Edit', 'Clone', and 'Delete' buttons at the top right of each section.

The screenshot shows the Object Manager interface under the 'Delivery' object. The left sidebar lists options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts (which is selected), Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main area shows a compact layout named 'Delivery Compact Layout'. It includes sections for 'Compact Layout Detail' (Label: Delivery Compact Layout, API Name: Delivery_Compact_Layout, Object Name: Delivery, Created By: Brunda Sanjana, Modified By: Brunda Sanjana), 'Included Fields' (Delivery Name, Owner), and 'Created By' and 'Modified By' details. There are 'Edit', 'Clone', 'Delete', and 'Compact Layout Assignment' buttons at the bottom.

The screenshot shows the Lightning App Builder interface. The left sidebar has a search bar and components: Quip Center Console home, Quip Associated Documents, Quip Document, Quip Notifications, Recent Items, Recent Records, Recommendations, Report Chart (which is selected), Rich Text, setup.cdp:cdpEducationalGuidance, Tableau Pulse, and Tableau View. The main area shows a 'Report Chart' component. It includes a 'Recent Records' section with a list of items like 'Sample Flow Report Screen Flows', 'Brunda Sanjana', 'Pharmacy Manager', and 'benrey spm'. Below it are three cards: 'All' (Nothing nothing in All yet. When records are added to the first view you'll see them here.), 'Assistant' (Nothing needs your attention right now. Check back later.), and 'Sample Flow Report: Screen Flows' (We can't draw this chart because there's no data.). On the right, there are configuration panels for 'Page > Report Chart': 'Label' (Leave blank for default...), 'Report' (Sample Flow Report: Screen Flows), 'Filter By' (None), 'Show Refresh Button' (checked), 'Cache Age (in minutes)' (1440), 'Set Component Visibility', and 'Filters'. At the bottom, there is a 'Get more on the AppExchange' button.



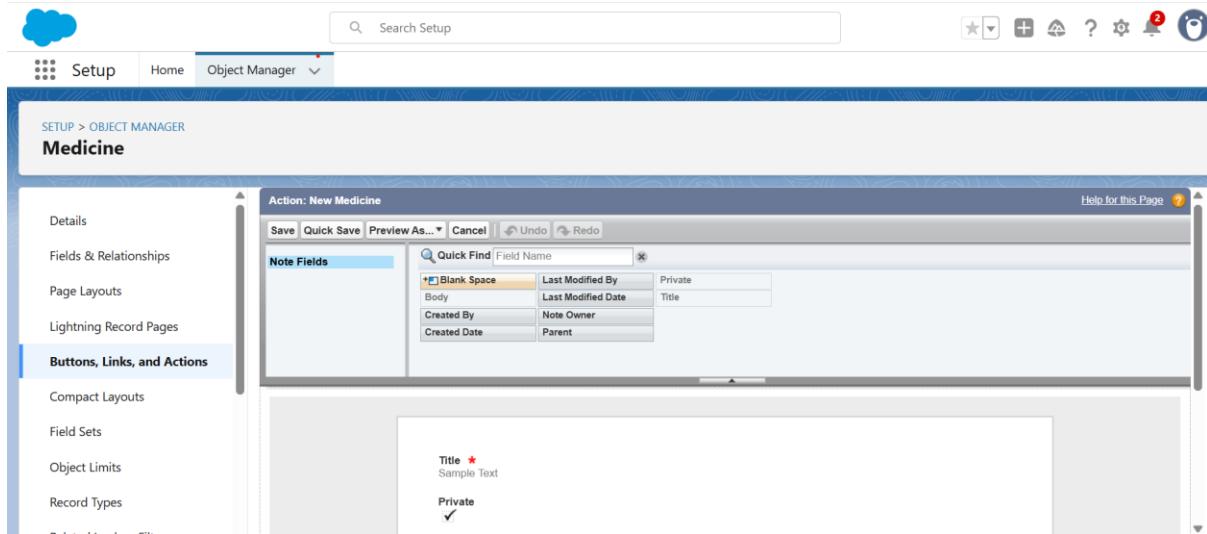
6.4 Utility Bar in Pharmacy Delivery CRM:

The Utility Bar is a persistent toolbar at the bottom of a Salesforce Lightning app that provides users with quick access to frequently used tools and utilities. It ensures that users can perform common tasks without leaving their current page.

Use in Pharmacy Delivery CRM

- **Quick Notes** → Delivery agents and pharmacists can capture notes about prescriptions, customers, or delivery challenges instantly.
- **Chat Support** → Agents can connect with pharmacy staff or customer support for real-time issue resolution.
- **Notifications** → Displays alerts for urgent deliveries, prescription approvals, or order delays.

- **Recent Items** → Allows quick navigation back to recently viewed delivery, prescription, or customer records.
- **Phone Integration (CTI)** → If integrated, agents can quickly call customers to confirm delivery times.



Key Outcomes:

- **Custom User Interfaces:** Created role-specific Record Pages (Delivery, Prescription, Customer, Pharmacy Partner) to streamline workflows.
- **App Navigation:** Configured Tabs for quick access to Deliveries, Prescriptions, Customers, Agents, and Reports.
- **Personalized Dashboards:** Designed Home Page Layouts for Pharmacists, Delivery Agents, and Admins to provide role-specific insights.
- **Utility Bar Integration:** Added utilities like Quick Notes, Notifications, Chat, and Recent Items for fast accessibility.
- **Custom Lightning Web Components (LWCs):** Built reusable components such as Delivery Tracker, Prescription Verification, and Digital Signature Capture.
- **Apex with LWC:** Implemented back-end logic (using Apex) to handle complex processes like delivery status updates, order validation, and prescription checks.
- **Event Handling in LWC:** Enabled seamless communication between components for real-time updates (e.g., delivery status change triggers notification to pharmacists).

Challenges and Resolutions

1. **Challenge:** Users faced cluttered interfaces due to too much information on record pages

Resolution: Implemented Dynamic Lightning Pages, showing relevant fields and components only for specific profiles (Pharmacist vs Delivery Agent).

2. **Challenge:** Delivery agents needed faster access to real-time order updates.

Resolution: Embedded **Live Delivery Tracker LWC** and connected it with Apex for real-time updates.

3. **Challenge:** Lack of smooth navigation between objects caused workflow delays.

- **Resolution:** Added **custom tabs** and organized them in logical order (Deliveries → Prescriptions → Customers → Agents → Reports).

Conclusion:

Phase 6 successfully transformed the Pharmacy Delivery CRM into a role-specific, user-friendly, and efficient Lightning Experience. By leveraging Lightning App Builder, Record Pages, Tabs, Home Layouts, Utility Bar, and custom LWCs integrated with Apex, the CRM now:

- Provides personalized experiences for pharmacists, delivery agents, and administrators.
- Enhances real-time tracking and communication between teams.
- Reduces manual work with automated actions and streamlined navigation.

Phase 7: Integration & External Access

Introduction:

In today's healthcare and delivery ecosystem, seamless integration is critical for efficient operations. A Pharmacy Delivery CRM often needs to interact with multiple external systems such as hospital management software, pharmacy inventory systems, courier services, payment gateways, and government healthcare APIs. Salesforce provides powerful tools to achieve this through Named Credentials, External Services, Web Services, Callouts, Platform Events, Change Data Capture, Salesforce Connect, API limits, OAuth, and Remote Site Settings.

Key Objectives

1. Named Credentials

- Store authentication details securely for external systems.
- Simplify making callouts to pharmacy partner APIs (e.g., stock verification, order placement).

2. External Services

- Connect Salesforce with external healthcare or pharmacy APIs without complex coding.
- Enable pharmacists to fetch drug availability or prescription details directly from integrated systems.

3. Web Services (REST/SOAP)

- Use REST/SOAP APIs to exchange data between CRM and external systems like hospital databases, courier tracking services, or payment processors.
- Example: Integrating courier tracking APIs to show real-time delivery updates in the CRM.

7.1 Web Services (REST/SOAP) in Pharmacy Delivery CRM:

Web Services (REST and SOAP) enable the Pharmacy Delivery CRM to integrate with external systems, allowing secure data exchange between Salesforce and third-party applications such as pharmacy inventory systems, hospital management software, courier tracking platforms, and payment gateways.

Use in Pharmacy Delivery CRM

- REST Services
 - Used for lightweight, real-time integrations.

- Example: Fetching delivery tracking details from a courier service API and displaying them on the Delivery Record Page.
- Example: Checking stock availability in partner pharmacy systems before order confirmation.
- **SOAP Services**
 - Used when strict contracts, structured data, or healthcare compliance (like HL7/FHIR-based systems) are required.
 - Example: Exchanging prescription details with hospital management systems using SOAP-based services.
 - Example: Securely submitting insurance claim details for prescriptions through SOAP APIs.

7.2 Callouts in Pharmacy Delivery CRM:

Callouts allow Salesforce to make outbound HTTP requests to external services (REST/SOAP APIs) to retrieve or send data. They are essential for real-time interactions with external systems used in pharmacy delivery operations.

Use in Pharmacy Delivery CRM

- **Delivery Updates** → CRM makes callouts to a courier partner's system to send delivery details and fetch live tracking updates.
- **Prescription Validation** → CRM calls out to a pharmacy's verification API to confirm whether a prescription is valid before dispatching.
- **Inventory Check** → Real-time callouts to pharmacy stock management systems to ensure medicines are available before scheduling delivery.
- **Payment Processing** → Secure callouts to payment gateways to process customer payments for medicine orders.

Example 1: Delivery Callout to Courier REST API

Trigger: Delivery Trigger

```
trigger Delivery Trigger on Delivery __c (after insert, after update) {
    if (Trigger is After && (Trigger is Insert || Trigger is Update)) {
        for (Delivery c delivery: Trigger. new) {
            if (delivery Status c == 'Out for Delivery') {
                // Call Queueable class to perform API callout
                System. Enqueue Job (new Delivery Callout Queueable (delivery Id));
```

```
    }
}
}
}
```

Example 2: Prescription Validation via SOAP Callout

```
trigger Prescription Trigger on Prescription__c (after insert) {
    for (Prescription__c pres : trigger.new) {
        if (pres.Status == 'Pending Verification') {
            System.enqueueJob(new PrescriptionValidationQueueable(pres.Id));
        }
    }
}

public class PrescriptionValidationQueueable implements Queueable, Database.AllowCallouts {
    private Id id;
    public PrescriptionValidationQueueable(Id id) {
        this.id = id;
    }
    public void execute(QueueableContext context) {
        Prescription__c pres = [SELECT Id, Patient__r.Name, Medicine__c FROM
            Prescription__c WHERE Id = :id LIMIT 1];
        try {
            // Example SOAP Stub (Generated from WSDL)
            PharmacyServiceSoap service = new PharmacyServiceSoap();
            PharmacyServiceSoap.PrescriptionValidationRequest req =
                new PharmacyServiceSoap.PrescriptionValidationRequest();
            req.medicineName = pres.Medicine__c;
        }
    }
}
```

Pharmacy Service Soap. Prescription Validation Response = service. Validate Prescription

```
// Update record based on response
```

```
}
```

```
}
```

```
}
```

The screenshot shows the Salesforce Setup interface with the following details:

- Setup Home:** The user is on the "Remote Site Settings" page under the "Custom Code" category.
- Page Title:** SETUP Remote Site Settings
- Section:** Remote Site Details
- Form Fields:**
 - Remote Site Name: PharmacyDeliveryAPI
 - Remote Site URL: https://pharmacydeliverycrm.com
 - Disable Protocol Security:
 - Description: (empty)
 - Active:
 - Created By: Brunda Sanjana, 03/10/2025, 11:18 pm
- Buttons:** Edit, Delete, Clone
- Page Footer:** Help for this Page ?

Key Outcomes

- Seamless API Integrations: Connected Salesforce CRM with external pharmacy systems, courier services, hospital management systems, and payment gateways using REST/SOAP web services.
- Real-Time Communication: Implemented Callouts, Platform Events, and Change Data Capture (CDC) for instant updates across systems (e.g., delivery status notifications, prescription approvals).
- Secure Authentication: Configured OAuth 2.0 and Named Credentials to ensure secure data exchange with external APIs.
- Data Accessibility: Used Salesforce Connect to view live inventory and prescription data from external pharmacy systems without duplicating records.
- Resilient Integration Management: Monitored and optimized API Limits, ensuring scalability for high transaction volumes.
- Trusted Connections: Added Remote Site Settings to whitelist external domains (courier APIs, pharmacy partners).

Challenges and Resolutions

1. Challenge: Handling large volumes of API calls led to API limit constraints.
 - o Resolution: Optimized integrations by batching requests, using Change Data Capture (CDC) instead of frequent polling, and monitoring usage with API dashboards.
2. Challenge: Authentication failures during callouts to external systems.
 - o Resolution: Configured Named Credentials with OAuth to store authentication details securely and reduce manual setup errors.
3. Challenge: Data sync issues between Salesforce and external pharmacy systems (e.g., prescription updates delayed).
 - o Resolution: Used Platform Events for real-time event-driven updates and Salesforce Connect for on-demand external data access.
4. Challenge: Error handling in callouts (e.g., courier system downtime).
 - o Resolution: Implemented Queueable Apex with retry logic and built an Integration Log object to capture failures and allow manual reprocessing.
5. Challenge: Ensuring compliance with healthcare and data security standards.
 - o Resolution: Enforced SSL connections, OAuth tokens, field-level encryption, and access control through permission sets and profiles.

Conclusion

Phase 7 successfully established a robust integration framework for the Pharmacy Delivery CRM. By leveraging Salesforce's integration tools—REST/SOAP web services, callouts, platform events, CDC, Salesforce Connect, and OAuth authentication—the system now enables:

- Secure, real-time data sharing across pharmacy partners, couriers, hospitals, and payment systems.
- Scalable integrations that handle high transaction volumes without exceeding API limits.
- Reliable workflows that ensure prescriptions, deliveries, and payments are always up-to-date.

Phase 8: Data Management & Deployment:

Introduction:

Efficient data management is crucial for the smooth operation of a Pharmacy Delivery CRM. Accurate customer, prescription, delivery, and pharmacy partner data ensures timely deliveries, reduces errors, and improves customer satisfaction. Salesforce provides tools like Data Import Wizard, Duplicate Rules, and the concept of Managed vs Unmanaged Packages to simplify data handling, maintain data integrity, and extend CRM functionality.

Key Objectives

1. Data Import Wizard

- Enable bulk import of Customers, Prescriptions, Deliveries, and Pharmacy Partner records.
- Simplify onboarding of new data without requiring complex ETL tools.
- Map external data fields to Salesforce fields accurately during import.
- Reduce errors and ensure data consistency across objects.

2. Duplicate Rules

- Detect and prevent duplicate Customer, Prescription, or Delivery records.
- Define **matching rules** (e.g., matching on email, phone number, or prescription ID).
- Automatically alert users or block duplicate record creation during data entry.
- Maintain **data integrity** to avoid delivery errors or billing issues.

3. Unmanaged vs Managed Packages

- **Unmanaged Packages**
 - Contain customizable components (objects, fields, workflows).
 - Ideal for **one-time deployments** or sharing solutions across orgs without future upgrades.
- **Managed Packages**
 - Contain locked components that can be upgraded centrally.
 - Ideal for **app distribution** via AppExchange and controlled updates

8.1 Data Import Wizard in Pharmacy Delivery CRM:

The Data Import Wizard is a user-friendly, point-and-click tool in Salesforce that allows administrators and users to import large volumes of data into standard or custom objects without requiring code. It is particularly useful for onboarding new customers, recording prescription histories, updating delivery information, or adding pharmacy partner details in bulk.

By using the Data Import Wizard, the Pharmacy Delivery CRM ensures that data is accurate, complete, and consistent, which is essential for timely deliveries, correct prescription fulfilment, and efficient customer management.

Key Features and Uses

1. Import Multiple Objects

- Supports **standard objects** like Accounts, Contacts, and Leads.
- Supports **custom objects** such as Deliveries, Prescriptions, Customers, and Pharmacy Partners.

2. User-Friendly Interface

- Guided, step-by-step wizard for uploading data files (CSV format).
- Provides options to map external data fields to Salesforce fields.

3. Data Mapping & Validation

- Automatically suggests field mappings, but allows manual adjustments.
- Validates data for required fields and correct formats before importing.

4. Duplicate Handling

- Integrates with **Duplicate Rules** to prevent creation of duplicate customer, prescription, or delivery records.
- Users can choose to block duplicates or allow but flag them for review.

5. Partial Imports & Error Reports

- Imports valid records even if some records fail validation.
- Generates detailed error reports to help correct issues in the source data.

8.2 Duplicate Rules in Pharmacy Delivery CRM:

Duplicate Rules in Salesforce help maintain clean, accurate, and reliable data by identifying and managing duplicate records during data entry or import. In a Pharmacy Delivery CRM, maintaining unique records for customers, prescriptions, deliveries, and pharmacy partners is critical to ensure timely delivery, correct prescription fulfilment, and efficient customer service.

Duplicate Rules work in tandem with Matching Rules, which define how Salesforce identifies potential duplicates (e.g., by email, phone number, prescription ID).

The screenshot shows the Salesforce Setup interface with a search bar at the top left containing "duplica". On the left, a sidebar menu under "Data" is expanded, showing "Duplicate Management", "Duplicate Error Logs", "Duplicate Rules" (which is selected and highlighted in blue), and "Matching Rules". A message at the bottom of the sidebar says, "Didn't find what you're looking for? Try using Global Search." The main content area is titled "d SETUP Duplicate Rules" and shows a "Medicine Duplicate Rule" for "Medicine Name". The "Duplicate Rule Detail" section includes fields for Rule Name (Medicine Name), Description (Medicine), Record-Level Security (Enforce sharing rules), Action On Create (Allow), Action On Edit (Allow), Alert Text (Use one of these records?), Active (unchecked), Conditions (empty), Created By (Brunda Sanjana, 03/10/2025, 11:58 pm), and Modified By (Brunda Sanjana, 03/10/2025, 11:58 pm). Buttons for Edit, Delete, Clone, and Activate are available. The status bar at the bottom right indicates "Order 1 of 1 [Reorder]".

This screenshot shows the same Salesforce Setup interface as the previous one, but the "Duplicate Rules" page is now for "Delivery Duplicate Rule" and "Correct Address". The "Duplicate Rule Detail" section includes fields for Rule Name (Correct Address), Description (Delivery), Record-Level Security (Enforce sharing rules), Action On Create (Allow), Action On Edit (Allow), Alert Text (Use one of these records?), Active (unchecked), Conditions (empty), Created By (Brunda Sanjana, 03/10/2025, 11:59 pm), and Modified By (Brunda Sanjana, 03/10/2025, 11:59 pm). Buttons for Edit, Delete, Clone, and Activate are available. The status bar at the bottom right indicates "Order 1 of 1 [Reorder]".

Key Features and Uses

1. Real-Time Duplicate Prevention

- Prevents users from creating duplicate records while entering new data manually.
- Example: Stops adding a new customer if the email or phone number already exists.

2. Duplicate Detection on Data Import

- Works with **Data Import Wizard** or other bulk import tools to flag or block duplicates.
- Ensures that bulk imports of prescriptions, deliveries, or customer lists do not create redundant records.

3. Customizable Actions

- **Block:** Prevent creation of duplicates entirely.
- **Allow but Alert:** Let the record be created but display a warning to review potential duplicates.

4. Integration with Matching Rules

- Define **matching criteria** for each object:
 - Customers → Email, Phone, or Name.
 - Prescriptions → Prescription ID, Patient Name, Medicine Name.
 - Deliveries → Delivery ID, Customer, or Address.
- Ensures duplicates are identified based on **business-critical fields**.

5. Reporting & Review

- Duplicate records are tracked and can be **reviewed and merged** by administrators.
- Supports **data hygiene** initiatives to maintain CRM accuracy.

8.3 Unmanaged vs Managed Packages in Pharmacy Delivery CRM:

Salesforce provides the concept of packages to bundle and deploy applications, components, or customizations across different Salesforce environments. In a Pharmacy Delivery CRM, packages are essential for reusing solutions, sharing custom components, and maintaining consistent functionality across development, testing, and production orgs.

There are two main types of packages: Unmanaged Packages and Managed Packages.

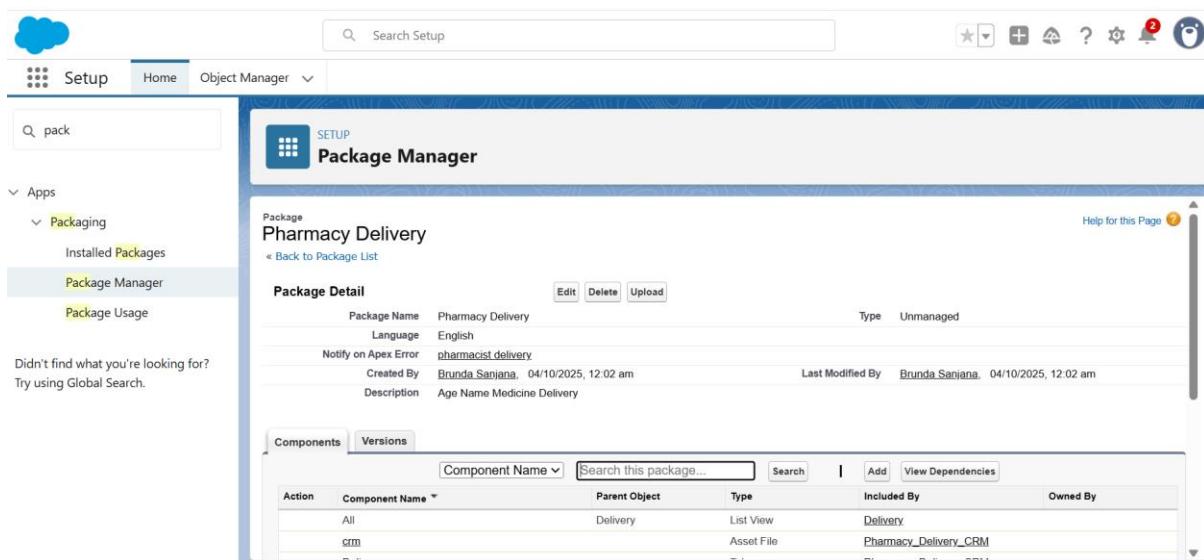
1. Unmanaged Packages

- Definition: Packages that are fully editable after installation. They are often used to share open-source solutions, templates, or one-time deployments.
- Key Characteristics:
 - Components (objects, fields, workflows, Apex code) can be modified after installation.
 - No version control or upgrade capability.
 - Ideal for learning, prototyping, or deploying one-time customizations.

- Use Case in Pharmacy Delivery CRM:
 - Sharing a set of custom objects and fields for deliveries and prescriptions with another Salesforce org without locking changes.
 - Deploying custom dashboards or reports templates to multiple branches of a pharmacy network.

2. Managed Packages

- Definition: Packages developed and maintained by Salesforce partners or ISVs that are version-controlled and upgradeable.
- Key Characteristics:
 - Components are locked to prevent unintended edits.
 - Supports upgrades and patches without affecting existing data or configurations.
 - Can be listed on AppExchange for distribution.
- Use Case in Pharmacy Delivery CRM:
 - Installing a third-party prescription verification app from AppExchange.
 - Deploying a delivery tracking module that can be upgraded centrally without manual intervention.



The screenshot shows the Salesforce Package Manager interface. The left sidebar has a search bar with 'pack' typed in and a 'Component Name' dropdown set to 'All'. Under 'Apps', 'Packaging' is expanded, showing 'Installed Packages', 'Package Manager' (which is selected), and 'Package Usage'. A note says 'Didn't find what you're looking for? Try using Global Search.' The main area is titled 'SETUP Package Manager' and contains a table with the following data:

Action	Component Name	Parent Object	Type	Included By	Owned By
All		Delivery	List View	Delivery	
crm			Asset File	Pharmacy_Delivery_CRM	
Delivery			Tab	Pharmacy_Delivery_CRM	
Delivery			Custom Object	Delivery All Delivery DeliveryTrigger	
Delivery			Change Event: Delivery	Apex Trigger	Delivery
Delivery agent		Delivery	Page Layout	Delivery	
Delivery Layout		Delivery	Page Layout	Delivery	
Delivery_Compact_Layout		Delivery	Compact Layout	Delivery	
DeliveryTrigger			Change Event: Delivery	Apex Trigger	Delivery
Remove	Pharmacy_Delivery_CRM			App	User Selected
	Pharmacy_Delivery_CRM_UtilityBar			Lightning Page	Pharmacy_Delivery_CRM

At the bottom, there's a 'Previous (1 - 11 of 11) Next' link.

Key Differences (Summary Table)

Feature	Unmanaged Package	Managed Package
Editable Components	Yes	No (locked)
Upgrade Capability	No	Yes
Version Control	No	Yes
Use Case	Prototyping, one-time deployment	Production apps, AppExchange solutions
Deployment Scope	Simple sharing between orgs	Distributed, upgradeable solutions

Key Outcomes

- Efficient Data Import & Onboarding
 - Successfully imported bulk customer, prescription, delivery, and pharmacy partner data using Data Import Wizard.
 - Reduced manual data entry and minimized errors during onboarding of large datasets.
- Data Accuracy & Integrity
 - Implemented Duplicate Rules and Matching Rules to prevent redundant records.
 - Maintained clean, accurate, and reliable CRM data for customers, prescriptions, and deliveries.
- Scalable & Modular Functionality

- Deployed custom solutions and reusable components using Unmanaged and Managed Packages.
- Enabled consistent functionality across different orgs and environments (development, testing, production).

4. Improved Operational Efficiency

- Streamlined workflows with accurate data, preventing delivery errors and prescription duplications.
- Reduced confusion for pharmacists and delivery agents by ensuring unique records for every customer and prescription.

5. Deployment & Upgrade Readiness

- Used Managed Packages for apps and modules that require version control and upgrades.
- Used Unmanaged Packages for one-time deployments or prototyping within the Pharmacy Delivery CRM.

Conclusion

- **Data quality** is maintained across all objects, reducing errors in deliveries and prescription fulfilment.
- **Packages** allow scalable and reusable deployment of custom functionality and third-party apps.
- The CRM is now **ready for growth**, supporting multiple pharmacy branches, external integrations, and ongoing enhancements with minimal risk to data integrity.

Phase 9: Reporting, Dashboards & Security Review

Introduction

Effective reporting and data security are critical for a Pharmacy Delivery CRM to monitor operations, track performance, and protect sensitive patient and prescription data. Salesforce provides a robust reporting framework with Reports, Report Types, and Dashboards, while Field-Level Security ensures that only authorized users can access sensitive information.

Key Objectives

1. Reports (Tabular, Summary, Matrix, Joined)

- Tabular Reports: Simple lists of deliveries, prescriptions, or customers for quick reference.
- Summary Reports: Aggregate data (e.g., total deliveries per agent, prescriptions per pharmacy) for performance analysis.
- Matrix Reports: Compare data across two dimensions (e.g., delivery status by region and by agent).
- Joined Reports: Combine data from multiple report types (e.g., linking prescriptions with deliveries) for comprehensive insights.
- Objective: Enable flexible, role-specific reporting for operational monitoring, decision-making, and KPI tracking.

2. Report Types

- Define which objects and relationships are reportable (e.g., Delivery with Customer, Prescription with Pharmacy).
- Support custom report types for specialized reporting requirements, such as tracking high-priority prescriptions or delayed deliveries.
- Objective: Provide structured reporting options tailored to the business needs of pharmacy operations.

3. Dashboards

- Visualize report data through charts, graphs, and gauges.
- Create role-specific dashboards for:
 - Pharmacists: pending prescriptions, inventory alerts.
 - Delivery Agents: daily assigned deliveries, route efficiency.
 - Managers: KPIs like delivery success rate, average prescription fulfillment time.

- Objective: Deliver real-time insights for quick decision-making and operational efficiency.

9.1 Reports (Tabular, Summary, Matrix, Joined):

Reports in Salesforce are tools that allow users to summarize CRM data to make informed business decisions. In a Pharmacy Delivery CRM, reports are critical for monitoring deliveries, prescriptions, customer interactions, and pharmacy operations. They help stakeholders like pharmacists, delivery agents, and managers track performance, identify bottlenecks, and improve service efficiency.

Types of Reports and Their Uses

1. Tabular Reports

- Simple list-style reports showing records in rows.
- Example Use Case: A list of all deliveries scheduled for the day or all prescriptions pending approval.
- Benefit: Quick reference and easy export for operational tasks.

2. Summary Reports

- Aggregate data and allow grouping by fields (e.g., by delivery agent or pharmacy).
- Example Use Case: Total number of prescriptions fulfilled per pharmacy or per day.
- Benefit: Helps managers track performance metrics and identify trends.

3. Matrix Reports

- Display data grouped by rows and columns, allowing comparison across two dimensions.
- Example Use Case: Comparing delivery success rate by region and delivery agent.
- Benefit: Enables multi-dimensional analysis for operational planning.

4. Joined Reports

- Combine multiple report types or objects into a single view.
- Example Use Case: Linking Prescriptions with Deliveries to see which prescriptions are pending delivery.
- Benefit: Provides comprehensive insights across multiple CRM objects.

Key Features in Pharmacy Delivery CRM

- Filtering & Sorting: Filter records by delivery status, prescription type, customer region, or pharmacy partner.
- Conditional Highlighting: Highlight high-priority prescriptions or delayed deliveries for quick attention.
- Scheduled Reports: Automatically send reports to users or managers at regular intervals (daily, weekly).
- Export & Sharing: Export reports to Excel or PDF or share them with other Salesforce users.

9.2 Report Types in Pharmacy Delivery CRM:

Report Types in Salesforce define which objects and fields can be included in a report. They determine the relationships between objects (like Deliveries, Prescriptions, Customers, and Pharmacy Partners) and provide the structure for building meaningful reports.

In a Pharmacy Delivery CRM, carefully designed report types enable pharmacists, delivery agents, and managers to operational data accurately and generate actionable insights.

Standard Report Types

- Predefined by Salesforce for standard objects (e.g., Accounts, Contacts, Opportunities).
- Example Use Case: Viewing all Customer Accounts with their related Deliveries.
- Quick to use but limited to the fields and relationships predefined by Salesforce.

Custom Report Types

- Created by administrators to include custom objects and specific relationships.
- Example Use Case: Linking Prescriptions with Deliveries to track which prescriptions are pending delivery.

- Supports complex reporting requirements specific to Pharmacy Delivery CRM operations.

Key Features

1. Primary Object Selection
 - Decide the main object for the report (e.g., Delivery, Prescription, Customer).
2. Related Object Inclusion
 - Include related objects (e.g., include Customer data in a Delivery report).
 - Supports one-to-many relationships for detailed insights.
3. Field Selection
 - Choose which fields are available in the report (e.g., Delivery Status, Prescription Type, Customer Contact).
4. Deployment and Security
 - Report types respect object-level and field-level security, ensuring users only see data they are authorized to access.

9.3 Dashboards in Pharmacy Delivery CRM:

Dashboards in Salesforce are visual representations of CRM data that consolidate information from multiple reports into charts, tables, gauges, and metrics. In a Pharmacy Delivery CRM, dashboards provide real-time insights into operations, delivery performance, prescription fulfilment, and customer interactions, enabling pharmacists, delivery agents, and managers to make quick, informed decisions.

Key Features

1. Visual Components
 - Charts (bar, pie, line) to track trends such as daily deliveries or prescription counts.
 - Gauges to monitor KPIs like delivery success rate or prescription turnaround time.
 - Tables and metrics for quick reference to key operational data.
2. Multiple Data Sources
 - Pulls data from multiple Reports (Tabular, Summary, Matrix, Joined) to provide comprehensive views.
3. Role-Specific Dashboards
 - Pharmacist Dashboard: Pending prescriptions, low-stock alerts, prescription fulfilment rates.

- Delivery Agent Dashboard: Assigned deliveries, completed deliveries, route efficiency metrics.
- Manager Dashboard: Overall KPIs, delivery success rate, prescription turnaround time, customer satisfaction metrics.

4. Real-Time Updates

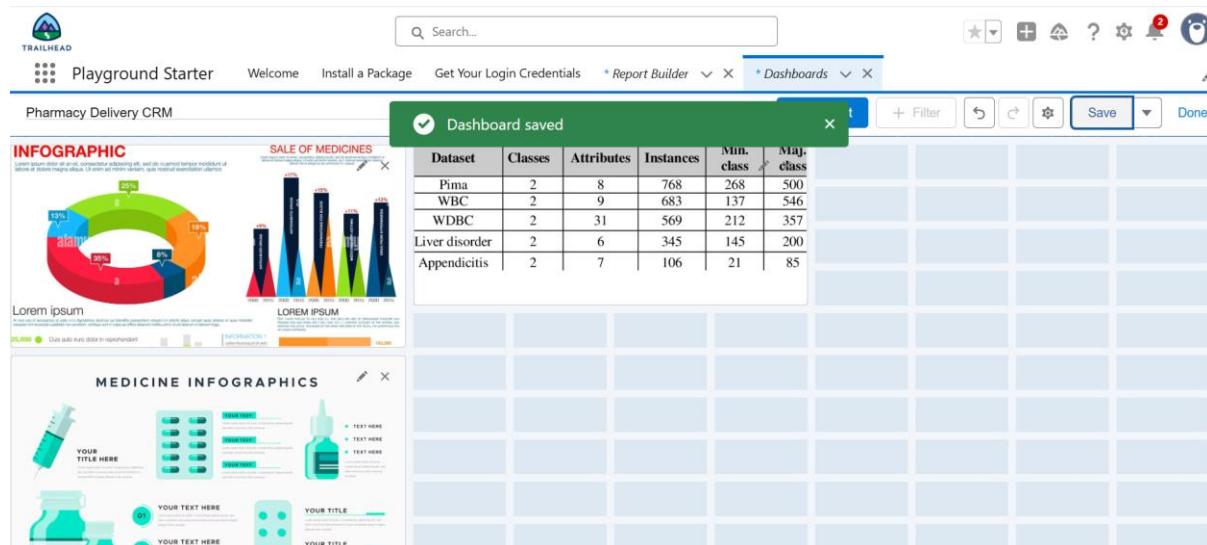
- Dashboards reflect real-time data to support timely decisions.
- Can be refreshed manually or scheduled for automatic updates.

5. Interactive Components

- Clickable charts and tables allow users to drill down into underlying reports for more detailed analysis.

6. Sharing & Security

- Dashboards respect profile permissions and field-level security, ensuring sensitive data is visible only to authorized users.



9.4 Field-Level Security (FLS) in Pharmacy Delivery CRM

Field-Level Security (FLS) is a Salesforce feature that controls user access to individual fields on objects. In a Pharmacy Delivery CRM, FLS ensures that sensitive information—such as patient details, prescription data, and financial information—is protected, while still allowing users to access the fields necessary to perform their roles.

FLS works alongside profiles and permission sets, enabling role-based access control and ensuring compliance with healthcare data privacy standards.

Sharing Settings

Object	Sharing Rule	Access Level	Controlled By
Streaming Channel	Public Read/Write	Private	✓
Tableau Host Mapping	Public Read Only	Private	✓
Thanks	Public Read Only	Private	<input type="checkbox"/>
User Provisioning Request	Private	Private	✓
Web Cart Document	Private	Private	✓
Work Order	Private	Private	✓
Work Plan	Private	Private	✓
Work Plan Template	Private	Private	✓
Work Step Template	Private	Private	✓
Delivery	Public Read/Write	Private	✓
Medicine	Public Read/Write	Private	✓
Medicine	Controlled by Parent	Controlled by Parent	
Prescription	Public Read/Write	Private	✓

Key Features:

1. Visibility Control
 - o Determines whether a user can view a field.
 - o Example: Only pharmacists and admins can view fields like Prescription Notes or Medical History.
2. Edit Permissions
 - o Determines whether a user can edit a field.
 - o Example: Delivery agents can update Delivery Status but cannot modify Prescription Details.
3. Role-Based Access
 - o Configured at the Profile or Permission Set level to grant or restrict access based on user roles.
 - o Example: Managers may have read-only access to all prescription fields, while pharmacists have full edit rights.

Key Outcomes:

1. Enhanced Reporting Capabilities
 - o Successfully implemented Tabular, Summary, Matrix, and Joined reports for Deliveries, Prescriptions, Customers, and Pharmacy Partners.
 - o Enabled role-specific insights for pharmacists, delivery agents, and managers.
2. Custom Report Types

- Created custom report types to track complex relationships (e.g., Prescriptions linked with Deliveries, Customers linked with Pharmacy Partners).
- Allowed comprehensive and actionable data analysis for operational efficiency.

Challenges and Resolutions:

1. Challenge: Complex reporting requirements across multiple objects.
 - Resolution: Implemented custom report types and joined reports to consolidate data for multi-object analysis.
2. Challenge: Users accessing sensitive information inadvertently.
 - Resolution: Configured Field-Level Security (FLS) and role-based access, ensuring that only authorized users can view or edit critical fields.
3. Challenge: Dashboards not reflecting real-time data for operational decisions.
 - Resolution: Set up real-time dashboard refresh and scheduled report updates to ensure accurate, up-to-date metrics.

Phase 9 successfully established a robust reporting and security framework in the Pharmacy Delivery CRM. Key outcomes include:

- Comprehensive visibility into deliveries, prescriptions, customer interactions, and pharmacy operations.
- Role-specific dashboards and reports for quick and informed decision-making.
- Data protection and compliance via Field-Level Security, safeguarding sensitive information.

Quality Assurance (QA) Testing for Pharmacy Delivery CRM:

Quality Assurance (QA) ensures that the Pharmacy Delivery CRM functions accurately, reliably, and efficiently according to business requirements. QA testing validates that all features—such as deliveries, prescriptions, customer management, dashboards, reports, integrations, and data security—work as expected before deployment to production.

The screenshot displays the 'New Prescription' screen of the Pharmacy Delivery CRM. On the left, there's a sidebar titled 'Recently Viewed' with two items: 'skin issue' and 'Fever issue'. The main area has a title 'New Prescription'. Below it, there's a section labeled 'Information' with a required field 'Prescription Name' containing the value 'Hair issue'. To the right of this field is an 'Owner' field set to 'Brunda Sanjana'. At the bottom of the screen are three buttons: 'Cancel', 'Save & New', and 'Save'.

