**Pharmacy Delivery CRM — Salesforce Implementation Plan (2–3 day MVP)**

**1. High-level approach**

Build the app on **Salesforce Lightning** using:

* Custom objects for Order, Prescription, Order Item, Delivery (plus Contact/Account for customers)
* Screen Flows for order creation and pharmacist verification
* Record-triggered Flows for status transitions & notifications
* Files (Salesforce Files) for prescription uploads
* Lightning App & Record Pages for UX
* Reports & Dashboard for basic metrics

Aim: deliver a fully demoable org without custom Apex (optional Apex only for advanced automation).

**2. Data model (Salesforce objects & key fields)**

Use **standard Contact** (or Account+Contact) for customer info. Create the following **custom objects**:

1. **Order\_\_c**
   * Fields:
     + Order\_Number\_\_c (Auto Number: ORD-{0000})
     + Contact\_\_c (Lookup → Contact)
     + Order\_Date\_\_c (DateTime: default now)
     + Status\_\_c (Picklist: Pending, Verified, Rejected, Out\_for\_Delivery, Delivered)
     + Total\_Amount\_\_c (Currency, optional)
     + Notes\_\_c (Long Text)
   * Related lists: Order\_Items\_\_r, Prescriptions\_\_r, Delivery\_\_r
2. **Order\_Item\_\_c**
   * Fields:
     + Order\_\_c (Lookup → Order\_\_c)
     + Medicine\_Name\_\_c (Text)
     + Quantity\_\_c (Number)
     + Dosage\_Instructions\_\_c (Text Area)
3. **Prescription\_\_c**
   * Fields:
     + Order\_\_c (Lookup → Order\_\_c)
     + File (use Files related to Prescription record — you may use a File related to the Prescription record or allow direct Files on the Order)
     + Uploaded\_By\_\_c (Lookup → User)
     + Verified\_By\_\_c (Lookup → User)
     + Verified\_At\_\_c (DateTime)
     + Verification\_Notes\_\_c (Long Text)

*Implementation note:* storing the file as a Salesforce File (ContentDocument) linked to the Prescription or Order is standard. You can create a Prescription record and then attach Files.

1. **Delivery\_\_c**
   * Fields:
     + Order\_\_c (Lookup → Order\_\_c)
     + Rider\_Name\_\_c (Text)
     + Rider\_Phone\_\_c (Phone)
     + Scheduled\_Time\_\_c (DateTime)
     + Pickup\_Time\_\_c (DateTime)
     + Delivered\_Time\_\_c (DateTime)
     + Delivery\_Notes\_\_c (Long Text)

**3. Security & Access**

* Profiles / Permission Sets:
  + Pharmacist: Read/Create/Edit on Order, Prescription; verify permission.
  + Dispatcher: Read/Edit on Delivery; assign rider.
  + Rider (optional): limited access to Delivery record.
  + Manager: full view + reports.
* Record-level sharing: use org-wide defaults Recommended: Private for Order\_\_c, and share via role hierarchy or sharing rules if needed.

**4. UI & UX (Lightning pages + Flows)**

* Create a **Pharmacy Delivery** Lightning App with tabs: Orders, Create Order (Flow), Dashboard, Reports.
* **Screen Flow — Create Order**:
  + Screen 1: Customer lookup / quick-create Contact (name, phone, address)
  + Screen 2: Add Order Items (use collection variable + Repeating Section pattern or multiple screens to add items)
  + Screen 3: Upload Prescription (file upload component — stores as Salesforce Files)
  + Screen 4: Review & Submit → Create Order\_\_c, Order\_Item\_\_c rows, Prescription\_\_c record & attach Files
* **Order Record Page**:
  + Show related Order Items, Prescriptions (Files), Delivery related list, and action buttons:
    - Verify Order (calls a Flow)
    - Reject Order (Flow to capture reason)
    - Assign Delivery (quick action or Flow)
    - Mark Out for Delivery / Mark Delivered (quick actions or Flow)
* Use **Dynamic Actions** on the record page so only appropriate actions show based on Status\_\_c.

**5. Automations (Flows & Notifications)**

* **Record-Triggered Flow** on Order\_\_c (after save)
  + When Status\_\_c changes:
    - If Verified → send email to customer (or create a Task / Chatter post)
    - If Rejected → send notification with reason
    - If Out\_for\_Delivery → notify assigned rider (email or task)
    - If Delivered → update Delivered\_Time\_\_c in Delivery\_\_c (if present)
* **Screen Flow — Verify Order**
  + Display prescription file preview via record data
  + Buttons: Verify (set status to Verified; set Verified\_By\_\_c, Verified\_At\_\_c) or Reject (set status to Rejected + capture notes)
* **Screen Flow — Assign Delivery**
  + Fields: Rider Name, Rider Phone, Scheduled\_Time
  + Creates/updates Delivery\_\_c and sets Order\_\_c.Status\_\_c → Out\_for\_Delivery (or Verified→Out\_for\_Delivery)
* Use **Email Templates** for status change emails (or just show success messages).

**6. Reports & Dashboard**

Create 3 simple reports:

1. Orders by Status (summary)
2. Recent Orders (last 30 days)
3. Deliveries (with rider and delivery times)

Dashboard components:

* Status counts (Pending, Verified, Out for Delivery, Delivered)
* Recent Orders table

**7. Deployment & Data (how to set up quickly)**

* Work in a **Developer Edition** / Scratch Org / Trailhead Playground.
* Create objects & flows using **Setup** (UI).
* Load sample data using **Data Import Wizard** or **Data Loader**:
  + Create Contacts, then Orders, Order Items, Prescription placeholder records, Delivery records.
* For files: upload sample PDFs via the UI to records or use Data Loader with ContentVersion.

**8. 3-Day Implementation Plan (Salesforce tasks)**

**Day 1 — Setup & Order Creation**

* Create custom objects and fields (Order\_\_c, Order\_Item\_\_c, Prescription\_\_c, Delivery\_\_c).
* Configure page layouts & compact layouts for Order\_\_c.
* Build the **Create Order Screen Flow** (customer lookup/create, add items, file upload).
* Test: create 3 sample orders with uploaded files.

**Deliverable Day 1:** Orders can be created via Flow; files attach to records.

**Day 2 — Verification & Dashboard**

* Build **Verify Order Flow** (screen to view prescription, verify/reject).
* Build **Assign Delivery Flow** and quick actions.
* Create Orders list view & Orders Lightning page (table + filters).
* Create basic Reports & Dashboard.
* Add Dynamic Actions and conditional visibility for action buttons.

**Deliverable Day 2:** Pharmacist can verify/reject; dispatcher can assign riders; dashboard works.

**Day 3 — Status Automation, Polish & Demo**

* Create record-triggered Flows for status change notifications.
* Add validation rules (e.g., Cannot mark Out\_for\_Delivery before Verified).
* Seed demo data, tweak UI (icons, badges), and prepare demo script.
* Create README with run steps & sample credentials (or share screenshots/video).

**Deliverable Day 3:** Demo-ready org; flows handle end-to-end lifecycle.

**9. Acceptance Criteria (Salesforce-specific)**

* Orders created via the Flow with uploaded prescription files.
* Pharmacist can open an Order, preview the prescription, and mark Verified/Rejected via Flow.
* Dispatcher can assign a rider and schedule delivery; Delivery\_\_c is created and linked to Order.
* Status changes trigger notifications (email/task) and are reflected on Order record.
* Dashboard shows accurate counts by status.
* All functionality implemented without Apex (unless optional advanced features chosen).

**10. Optional stretch (if time allows)**

* Build a simple **LWC** for an improved multi-row Order Items editor.
* Integrate Twilio or SMS (requires external setup) for rider/customer SMS notifications.
* Expose a public Experience Cloud site / web form for customers to submit orders.
* Add role-based Home pages for Pharmacist / Dispatcher.

**11. Demo script (1.5–3 minutes)**

1. Launch **Pharmacy Delivery** Lightning App. Show App navigation (Orders, Create Order).
2. Click **Create Order** → run Flow: create a customer (or lookup), add a medicine, upload prescription, submit. Show the created Order record.
3. Open the Order record → show prescription file preview.
4. Click **Verify** → run Verify Flow → mark Verified.
5. Click **Assign Delivery** → add Rider details → show Delivery record created.
6. Mark **Out for Delivery** then **Delivered**. Show dashboard counts updated.
7. Show Reports & list filters.

**12. What to submit (for assignment)**

* Google Doc containing problem statement + this Salesforce implementation plan (copy-paste).
* GitHub repo (if any code or LWC used).
* Screenshots / short screen recording of:
  + Create Order Flow
  + Order record with prescription preview
  + Verification action
  + Delivery assignment and status changes
  + Dashboard
* If you used a Developer Org / Playground: share login steps (or export configuration as change set / SFDX metadata).

**13. Quick tips to finish in 2–3 days**

* Use **Screen Flows** heavily — they’re fast to design and demo.
* Keep validation light; focus on happy path for demo.
* Use sample PDF images for prescriptions (no need to capture real prescriptions).
* Avoid custom code unless a Flow cannot do it — Flows + Files + Reports are sufficient.
* Use Dynamic Actions and component visibility to simplify the UI for each user role.