Complaint Management System (CRM)

Prepared by: Praveenkumar Menasinakai

# Phase 1: Problem Understanding & Industry Analysis

## Problem Statement: Complaint Management System

Organizations, especially those in banking, telecom, retail, and utilities, receive a large number of customer complaints every day. These may be related to billing errors, service interruptions, product issues, or general inquiries. Currently, complaints are often handled manually or via email, which leads to delays, missed follow-ups, lack of accountability, and poor customer satisfaction. A centralized Complaint Management CRM is required to track, assign, and resolve complaints efficiently while providing transparency to customers and management.

## Requirement Gathering

Core Functionality:

- Customers (via agents or portal) should be able to log complaints linked to Accounts & Contacts.  
- Complaints must have type, subject, description, and priority.  
- Complaints should be auto-assigned to agents with SLA due dates.  
- Agents must be able to update status, resolution, and add notes.  
- Duplicate complaints for the same customer/type should be prevented.  
- Customers should receive notifications when complaint status changes.

Reporting & KPIs:

- Number of complaints by type, status, and branch.  
- SLA compliance (% complaints resolved within SLA).  
- Average resolution time.  
- Agent workload distribution.

Automation:

- Auto-assign agent and calculate SLA due date on complaint creation.  
- Escalate complaint if SLA overdue.  
- Notify customers when status updates.  
- Prevent duplicate open complaints.

## Stakeholder Analysis

Customers: Need a transparent way to track their complaints and receive timely updates.  
Agents: Need a queue of assigned complaints, visibility of priorities, and ability to update.  
Managers: Need dashboards to monitor SLA performance and escalated complaints.  
Admin: Needs to configure objects, fields, validation, automation, and permissions.

## Business Process Mapping

Step 1: Customer/Agent logs complaint linked to Contact & Account.  
Step 2: System auto-assigns agent and sets SLA due date.  
Step 3: Agent works on complaint and updates status/resolution.  
Step 4: If SLA overdue, system escalates complaint.  
Step 5: Complaint closed with resolution notes and notification sent.  
Step 6: Complaint data feeds into reports & dashboards.

Simple Process Flow Diagram (placeholder for screenshot):

[Submit Complaint] → [Auto Assign + SLA] → [Agent Works] → [Escalation if overdue] → [Close Complaint] → [Reports]

## Industry-Specific Use Case Analysis

Complaint Management CRM is essential for industries with large customer bases. For telecom companies, it ensures quick response to service outages. For banks, it ensures compliance with regulatory complaint handling. For retail, it reduces dissatisfaction from defective products. Overall, it improves customer loyalty, reduces churn, and ensures service excellence.

## AppExchange Exploration

Explore existing Service Cloud-based apps such as Complaint Handling for Financial Services, ServiceMax, or Customer Service apps on AppExchange. These apps demonstrate best practices in case management, escalation, and SLA tracking. They can be used for benchmarking reports, dashboards, and automation flows for the Complaint Management CRM.

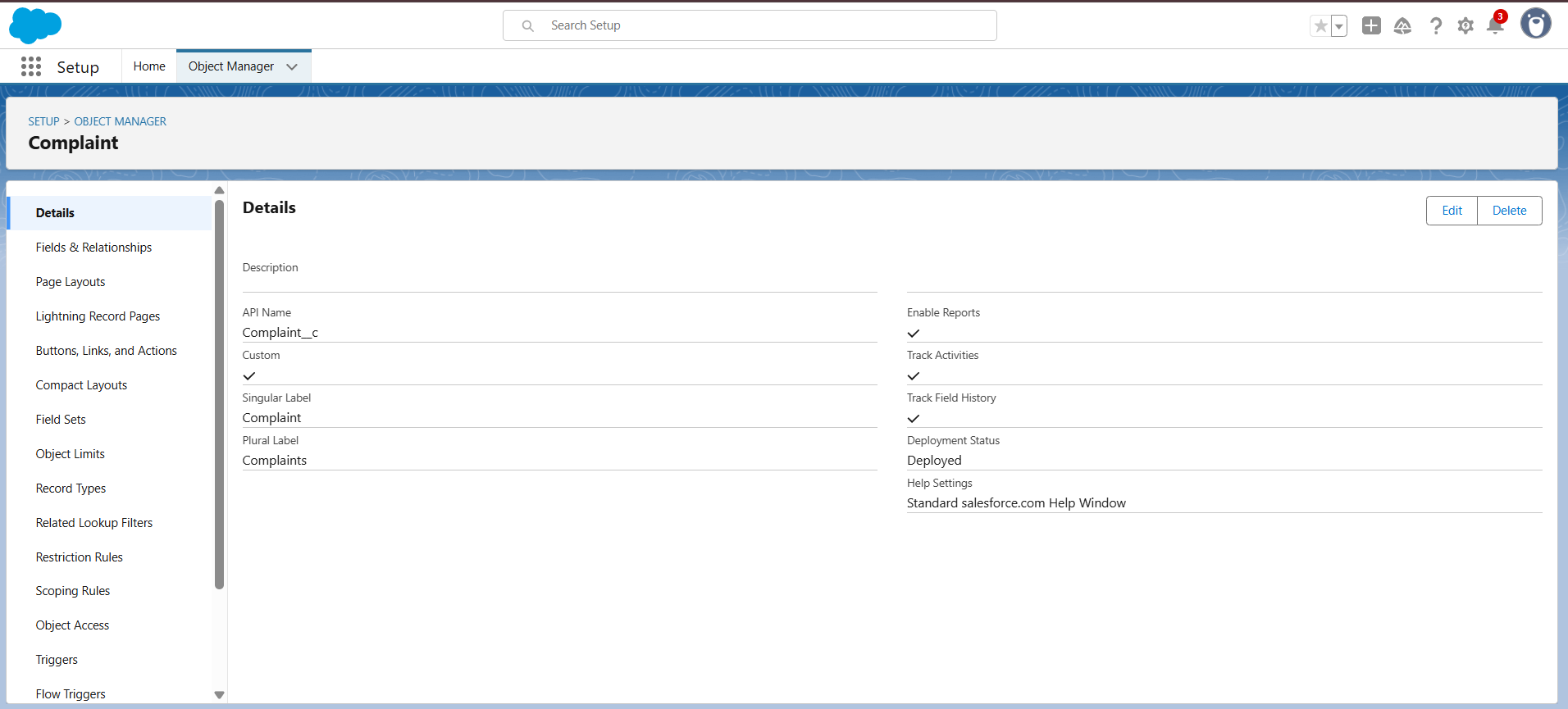
# Phase 2: Org Setup & Configuration

The following steps outline the detailed setup and implementation of the Complaint Management CRM.

**Step 1 — Create the Complaint Object and Fields**

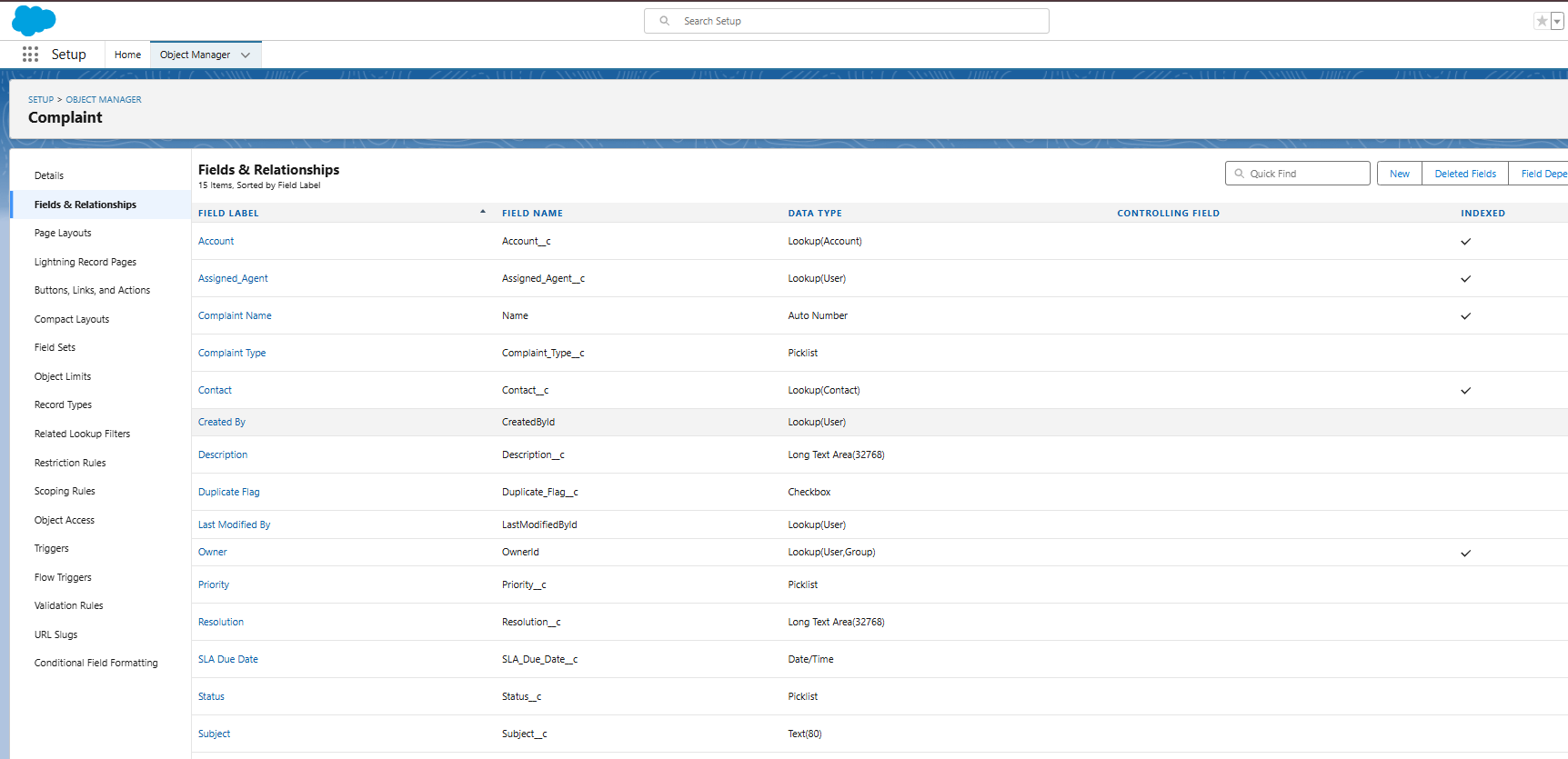
**Setup → Quick Find → Object Manager → Create → Custom Object**

* **Label**: Complaint
* **Plural Label**: Complaints
* **API Name**: Complaint\_\_c
* **Record Name**: Auto Number (Format: CMP-{0000})
* **Check**: Launch New Custom Tab (optional).



**Add Fields**:

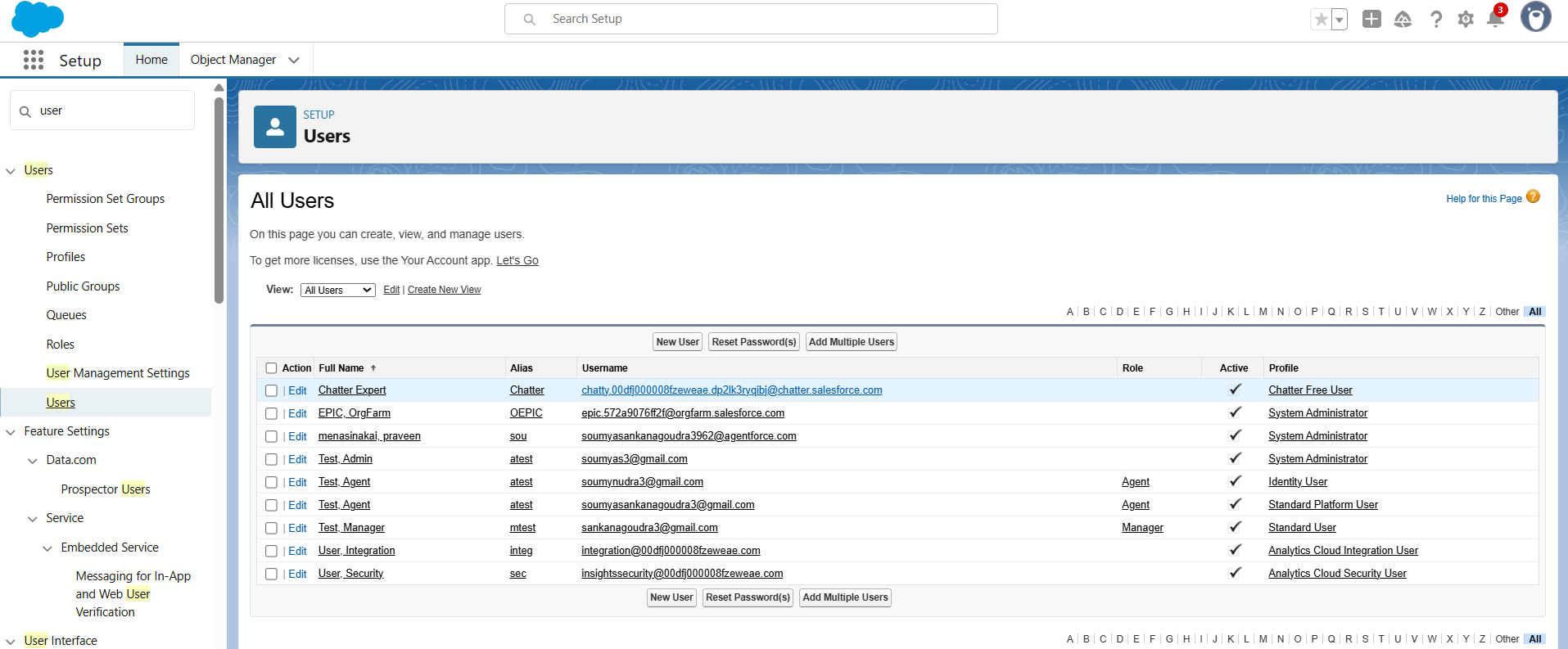
* Contact\_\_c — Lookup (Contact)
* Account\_\_c — Lookup (Account)
* Subject\_\_c — Text (80)
* Description\_\_c — Long Text Area (1000+)
* Complaint\_Type\_\_c — Picklist: Billing, Service, Product, Other
* Status\_\_c — Picklist: New, In Progress, Escalated, Resolved, Closed
* Priority\_\_c — Picklist: Low, Medium, High
* Assigned\_Agent\_\_c — Lookup (User)
* SLA\_Due\_Date\_\_c — Date/Time
* Resolution\_\_c — Long Text Area
* Duplicate\_Flag\_\_c — Checkbox (optional)



**Step 2 — Basic Data Setup: Profiles, Users, Roles & OWD**

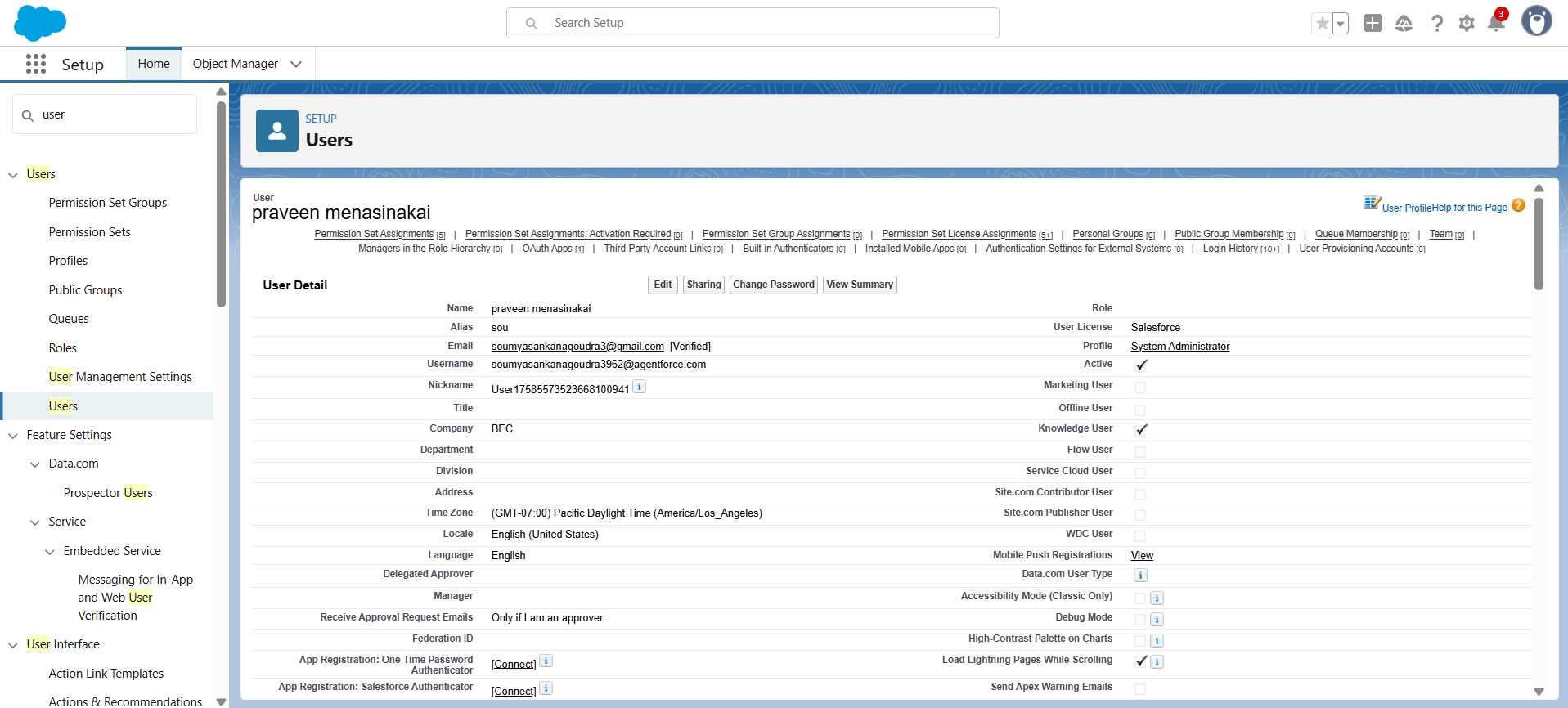
**Users**:

* Admin (System Administrator)
* Manager (Standard User)
* Agent (Standard User)



**Roles**:

* Manager (parent role: higher-level visibility)
* Agent (reports to Manager)



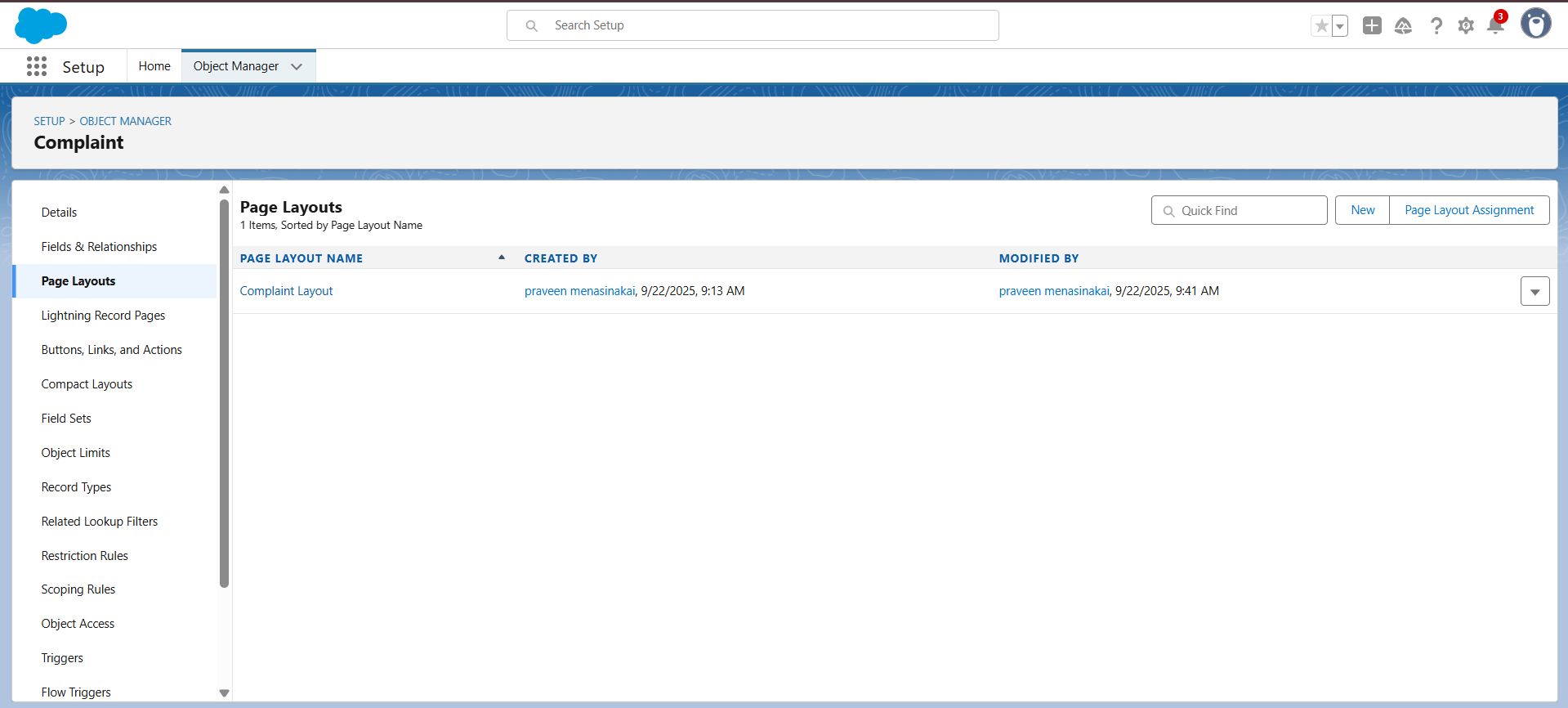
**OWD (Org-Wide Defaults)**:

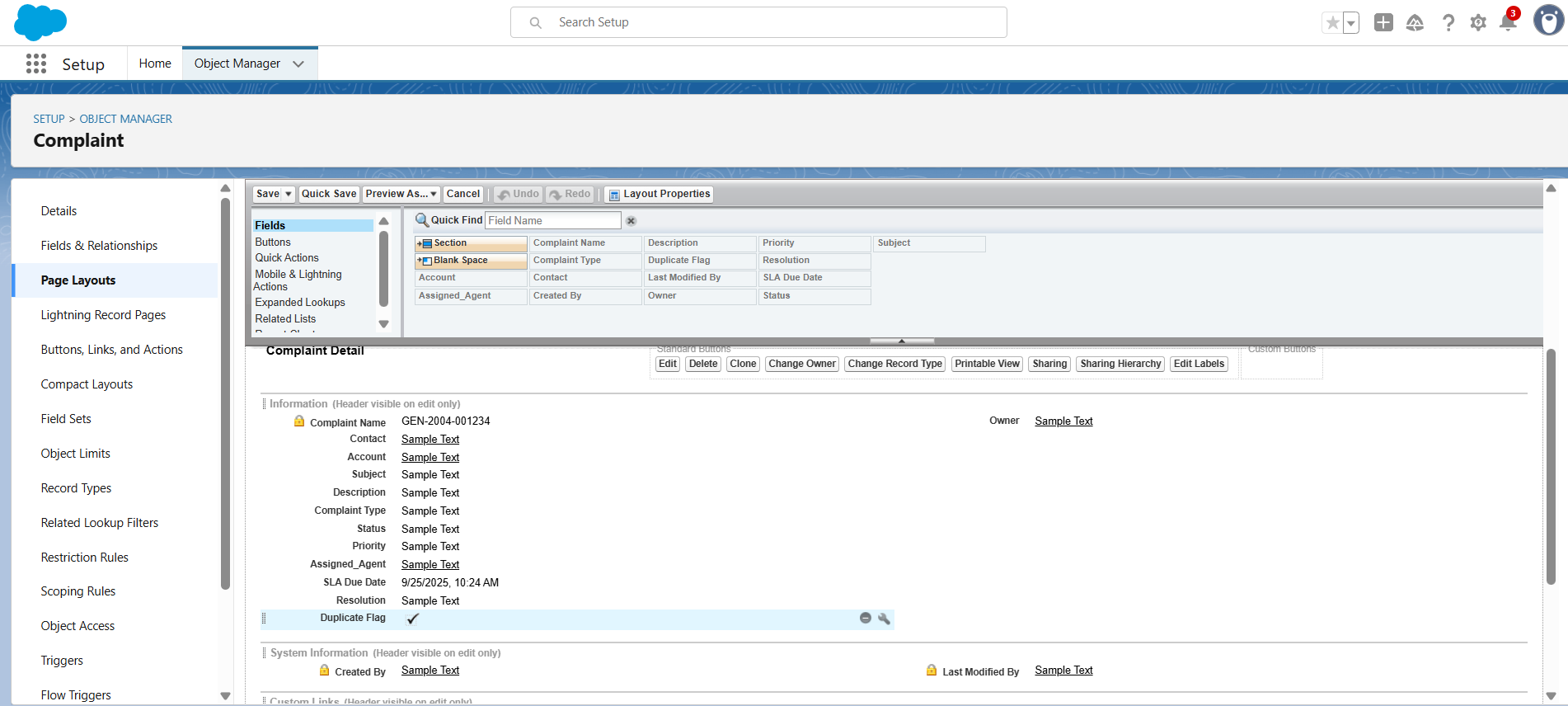
* Complaint = **Private**

**Step 3 — Page Layouts, Compact Layouts & List Views**

**Page Layouts**: Add key fields (Assigned\_Agent\_\_c, SLA\_Due\_Date\_\_c, Resolution\_\_c). Add Related Lists.  
**Compact Layout**: Subject, Status, Priority, Assigned Agent.  
**List Views**:

* *Open Complaints*: Status = New OR In Progress.
* *My Complaints*: Assigned\_Agent\_\_c = Current User.

****

****

**Step 4 — Validation Rules**

**Rule 1: Require Complaint Type**  
Formula:

ISBLANK(Complaint\_Type\_\_c)

Error: Please select a Complaint Type.

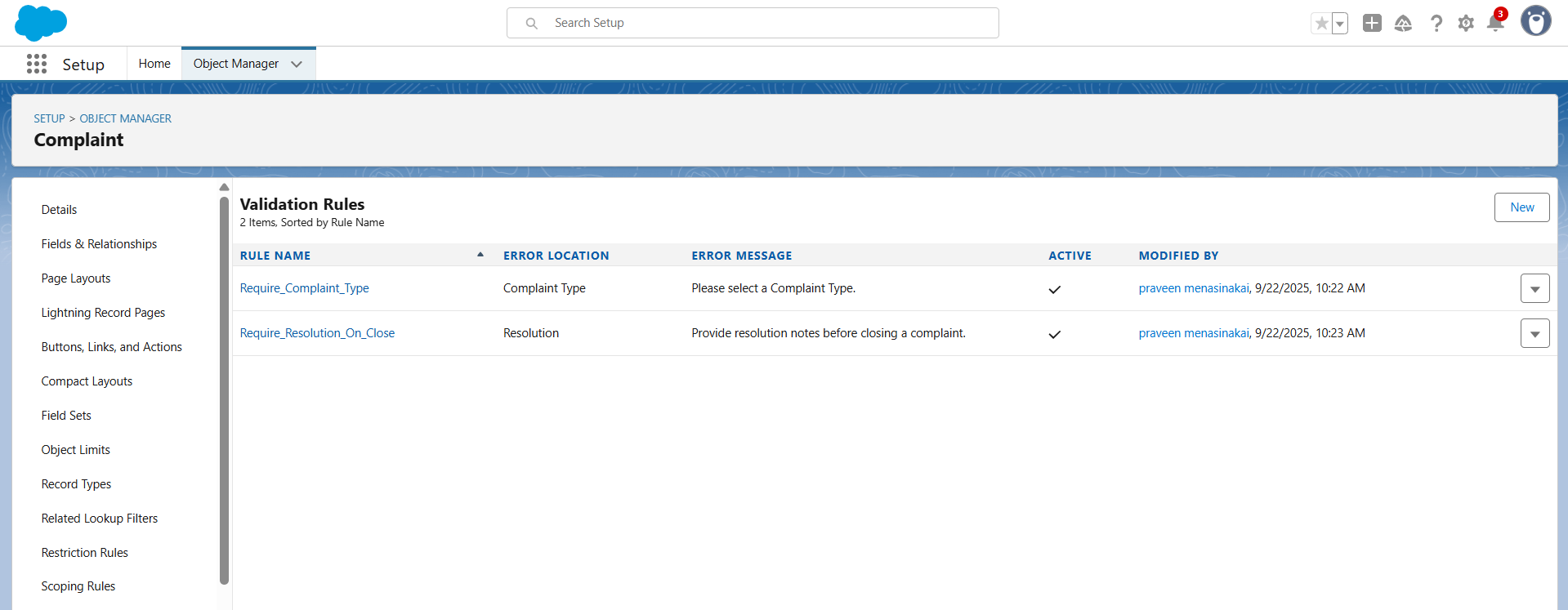
**Rule 2: Cannot close without Resolution**  
Formula:

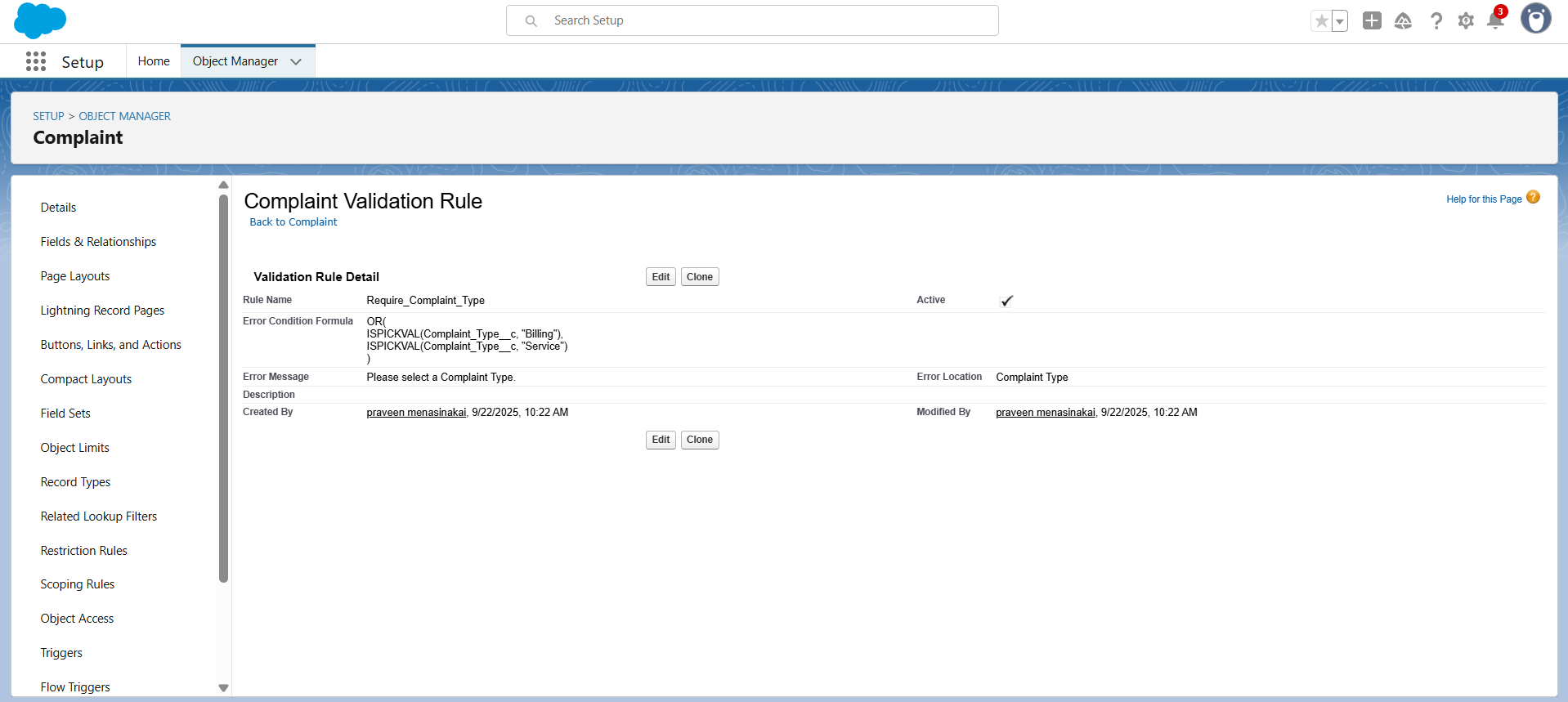
AND(

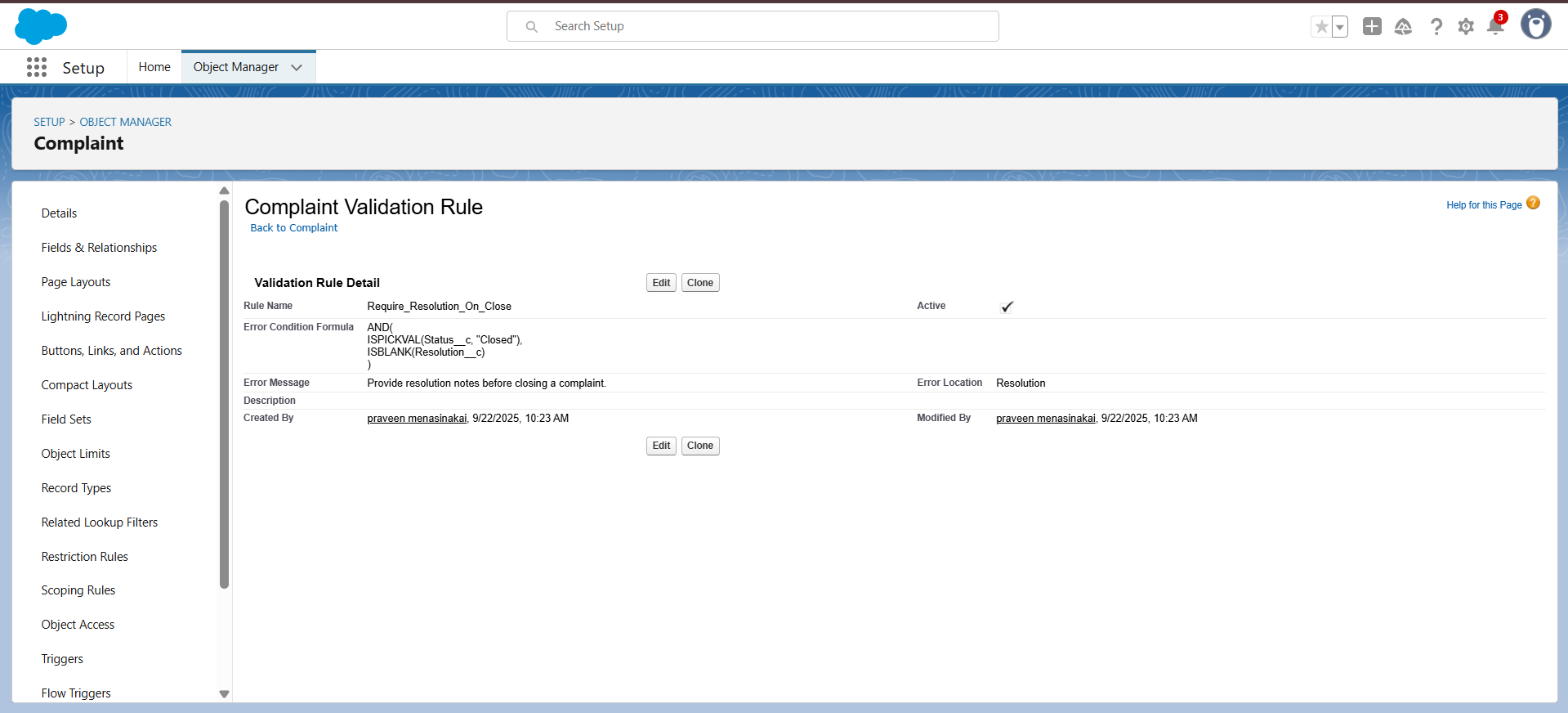
ISPICKVAL(Status\_\_c, "Closed"),

ISBLANK(Resolution\_\_c)

)







**Step 5 — Flows (Automations)**

**Flow A — Auto Assign & SLA Due Date** (Record-Triggered Flow, On Create).

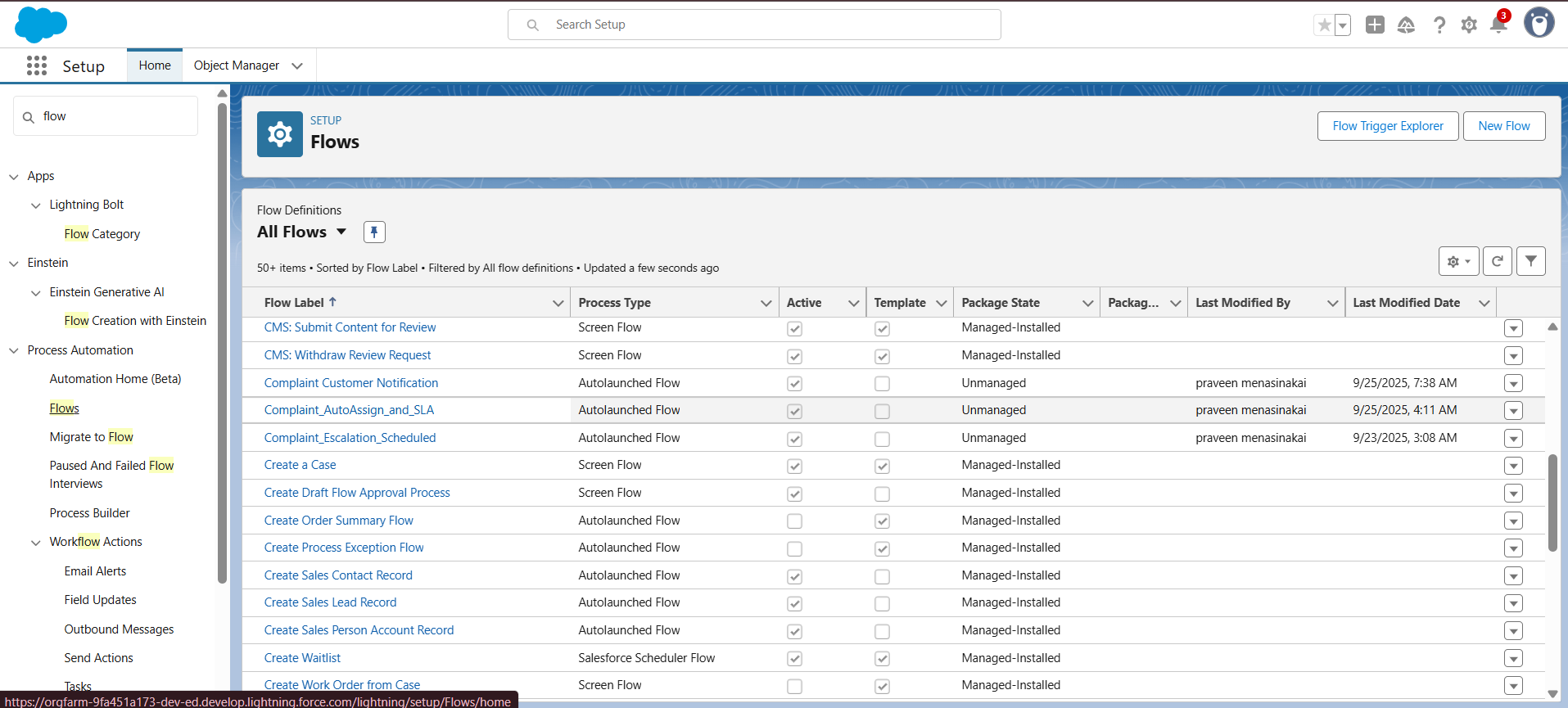
* Assign to agent automatically.
* Set SLA = CreatedDate + 24 hours.

**Flow B — Escalation if SLA Overdue** (Scheduled Flow).

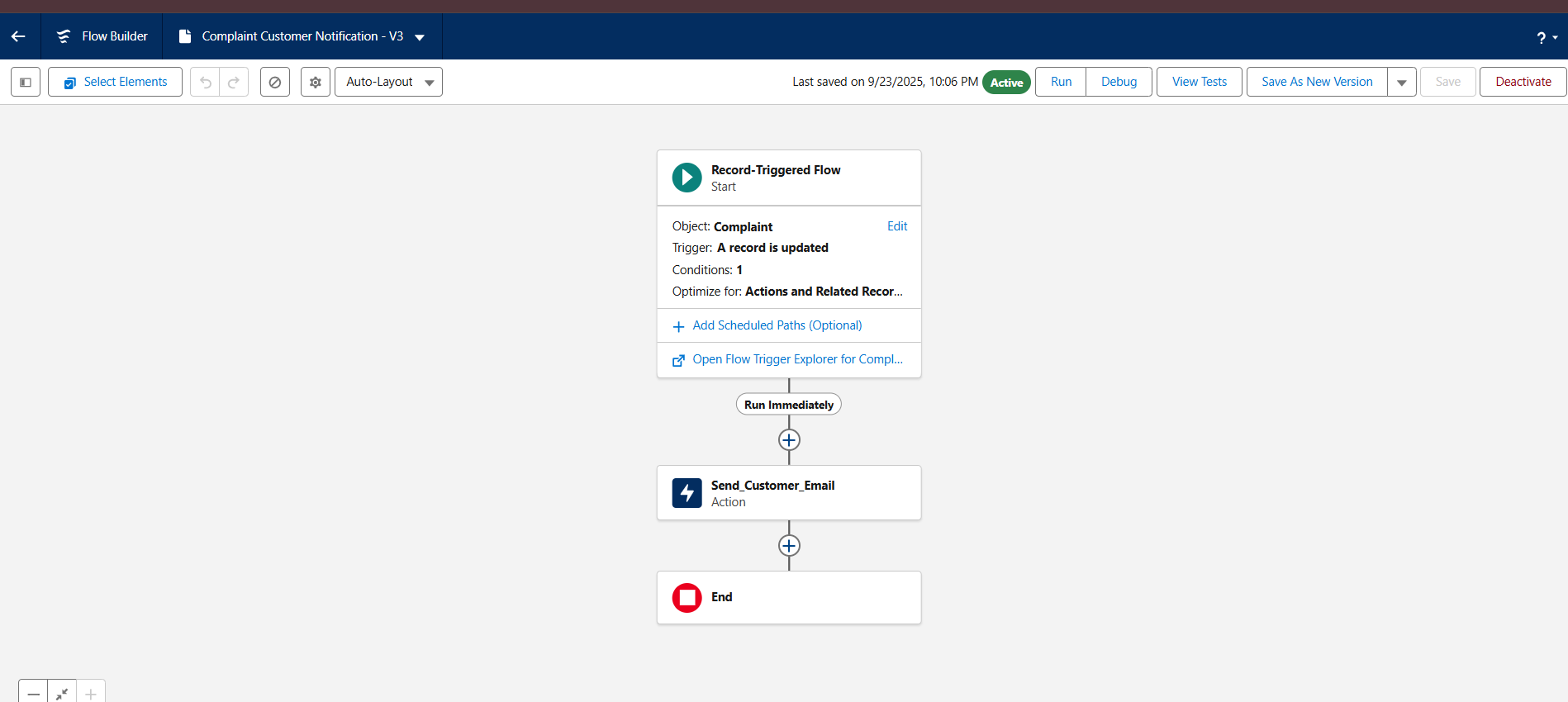
* Run hourly/daily.
* If SLA < Now and Status in New/In Progress → set Status = Escalated.

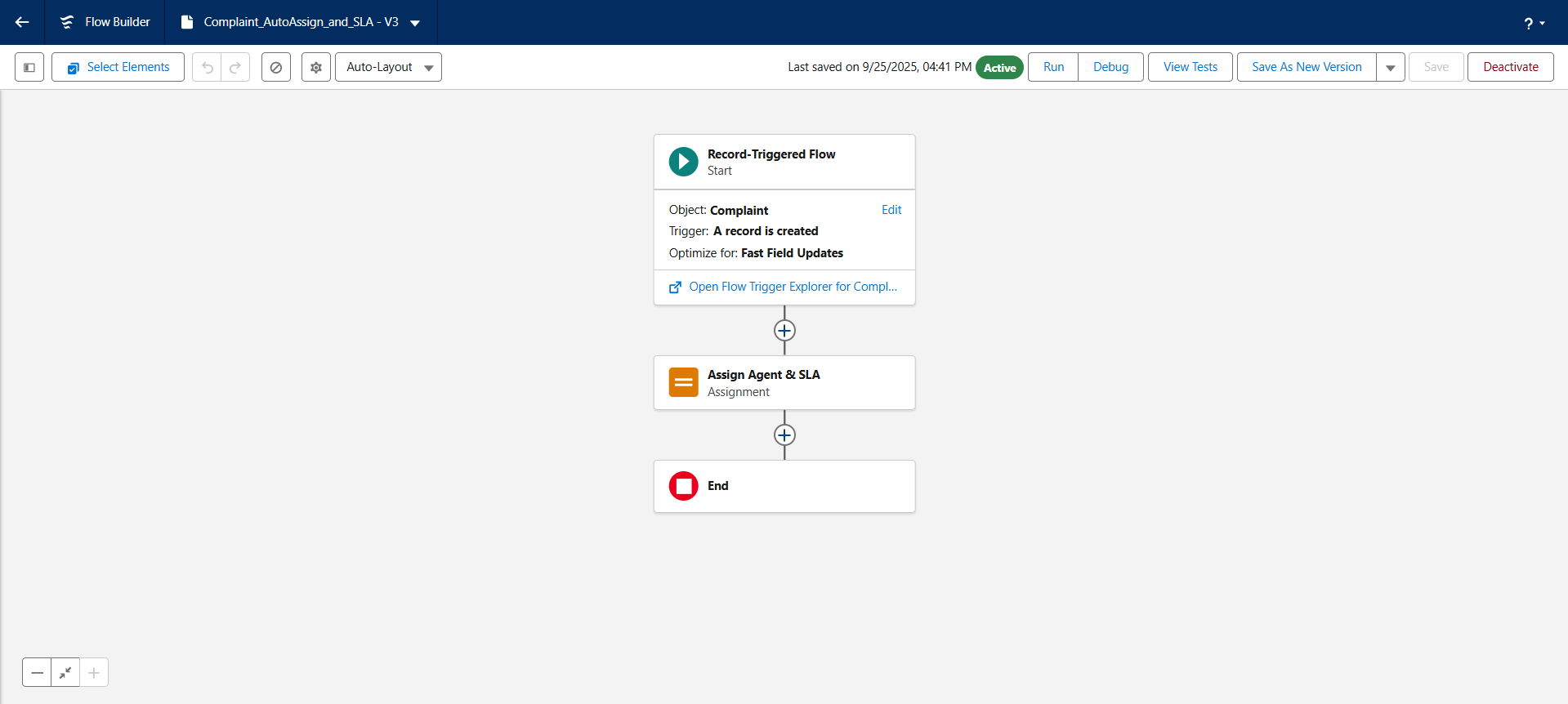
**Flow C — Notify Customer** (Record-Triggered Flow, On Status Change).

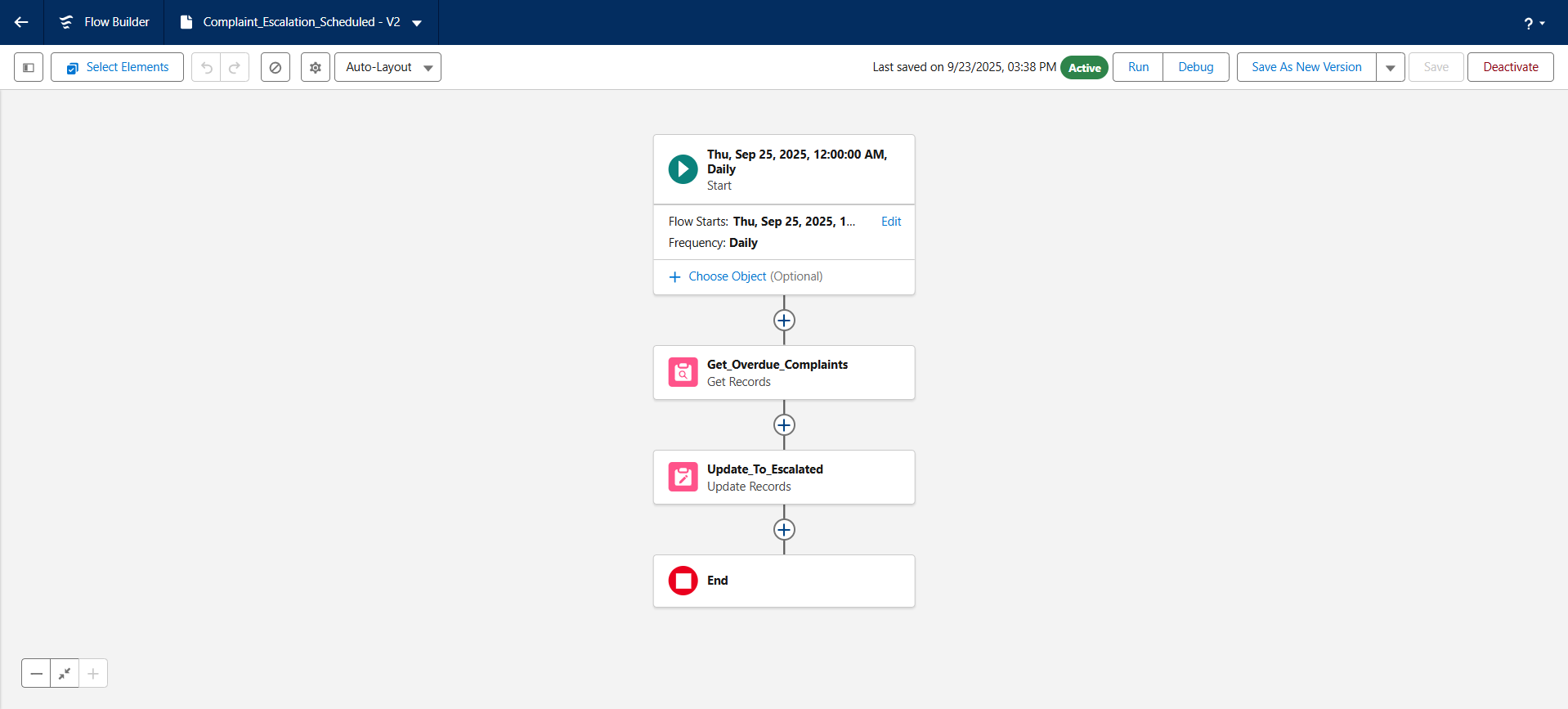
* Send Email Notification.



FLOW :A



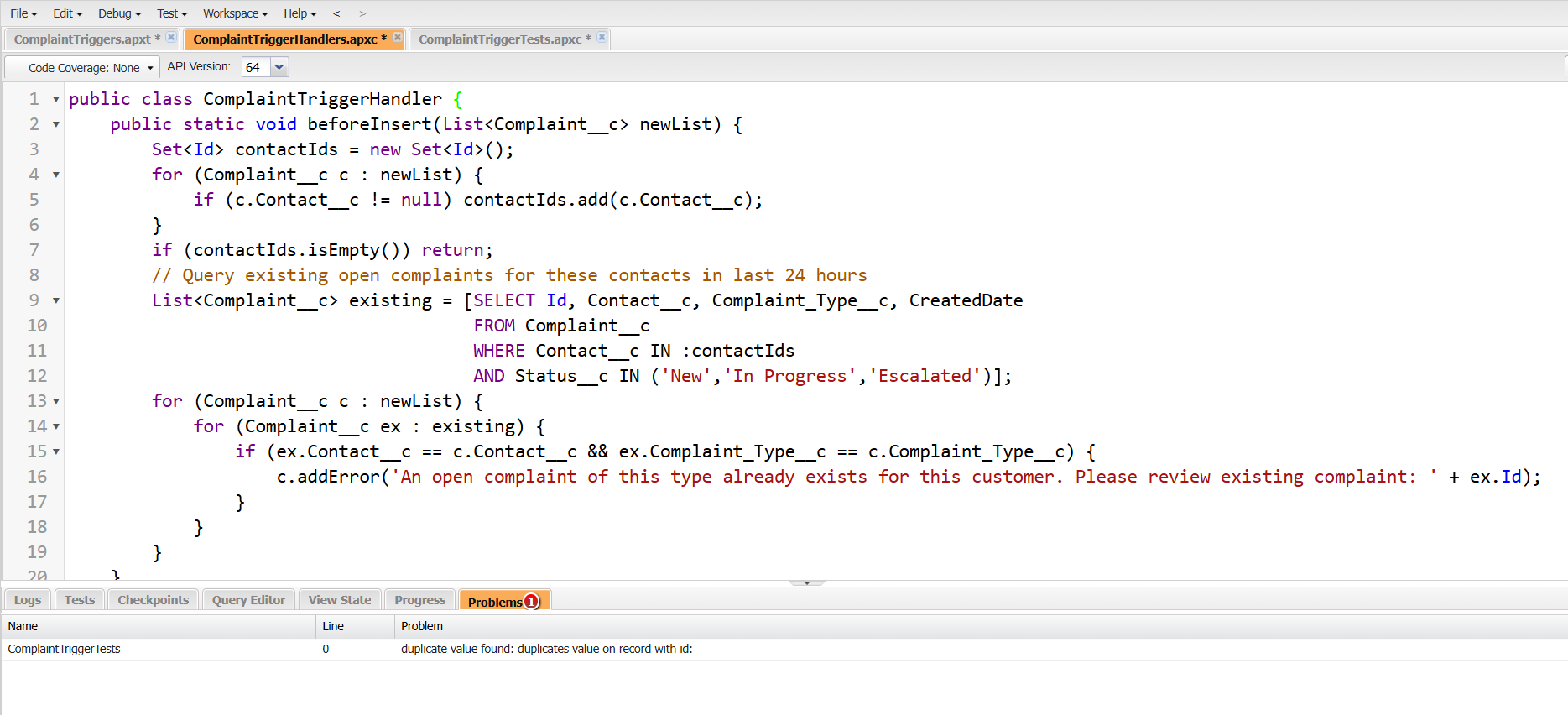
FLOW :B

FLOW :C

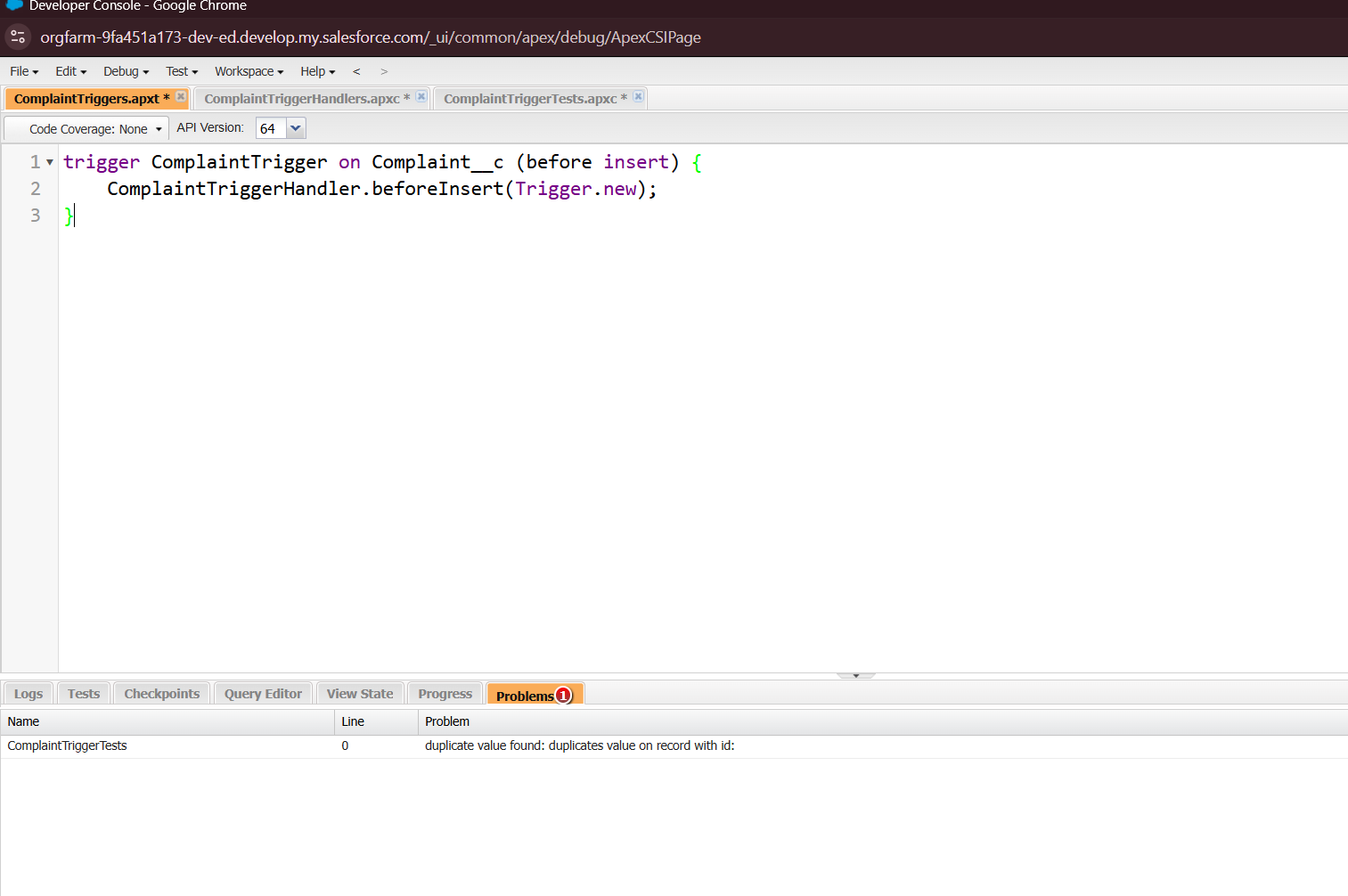
**Step 6 — Apex Trigger & Test Class**

**Trigger**: Prevent duplicate complaints (same Contact + Type + open status).  
**Test Class**: Try inserting duplicate → must fail.

**Apex Trigger (ComplaintTrigger.trigger):**



**Handler Class (ComplaintTriggerHandler.cls):**



**Test Class (ComplaintTriggerTest.cls):**



**How to deploy/test Apex:**

1. Developer Console → File → New → Apex Class → paste handler and test class.
2. Developer Console → New → Apex Trigger → paste trigger.
3. Run Apex tests (Setup → Apex Test Execution) and ensure green.

**Step 7 — Permission Sets & Sharing**

**Permission Sets**:

* Complaint\_Admin → Full CRUD
* Complaint\_Agent → Create, Read, Edit
* Complaint\_Manager → Read, Edit, Delete

**Field-Level Security**: Restrict Resolution\_\_c to Manager/Admin only.

**Sharing**: OWD Private. Create Sharing Rule → share Agents’ records with Managers.

**Step-by-Step Setup:**

**Permission Sets, Field-Level Security & Sharing Rules for Complaint\_\_c**

**🧩 1. Create Permission Sets**

**✅ Step 1: Go to Permission Sets**

* Navigate to:  
  Setup → Permission Sets

**✅ Step 2: Create Each Permission Set**

**A. Complaint\_Admin**

* Click **New**
* Label: Complaint\_Admin
* Save
* Go to: Object Settings → Complaint\_\_c → Edit
* Check: **Read, Create, Edit, Delete**
* Enable **View All** if needed
* Save

**B. Complaint\_Agent**

* Click **New**
* Label: Complaint\_Agent
* Save
* Go to: Object Settings → Complaint\_\_c → Edit
* Check: **Read, Create, Edit**
* Leave **Delete** unchecked
* Save

**C. Complaint\_Manager**

* Click **New**
* Label: Complaint\_Manager
* Save
* Go to: Object Settings → Complaint\_\_c → Edit
* Check: **Read, Edit, Delete**
* Enable **View All Records**
* Save

**🙍‍♂️ 2. Assign Permission Sets to Users**

**✅ Step 1: Open Permission Set**

* Go to: Setup → Permission Sets
* Select one (e.g. Complaint\_Agent)

**✅ Step 2: Add Assignments**

* Click: Manage Assignments → Add Assignments
* Select users (e.g. test Agent user)
* Click **Assign**

➡ Repeat for Complaint\_Admin and Complaint\_Manager sets.

**🛡️ 3. Configure Field-Level Security (FLS) on Resolution\_\_c**

**✅ Step 1: Go to Field Settings**

* Navigate to:  
  Setup → Object Manager → Complaint\_\_c → Fields & Relationships
* Click on: Resolution\_\_c

**✅ Step 2: Set Field-Level Security**

* Click **Set Field-Level Security**
* **Uncheck** visibility for **Agent** profile
* **Check** visibility for **Manager** and **Admin** profiles
* Save

🔒 This hides the Resolution\_\_c field from Agents.

**🔁 4. Create Sharing Rules (for Managers to View Complaints)**

**✅ Step 1: Go to Sharing Settings**

* Navigate to:  
  Setup → Sharing Settings → Complaint Sharing Rules

**✅ Step 2: Create New Rule**

Click **New Sharing Rule**

**Option A: Criteria-Based Sharing**

* Rule Type: **Criteria-based**
* Rule Name: Share\_All\_to\_Managers
* Criteria: e.g. Status ≠ Closed (or leave blank to share all)
* Share with: Public Group = Managers
* Access Level: **Read Only** or **Read/Write**
* Save

**Option B: Owner-Based Sharing**

* Rule Type: **Owner-based**
* Rule Name: Share\_Agent\_Complaints
* Owned By: Role = Agent
* Share with: Role = Manager
* Access Level: **Read Only** or **Read/Write**
* Save

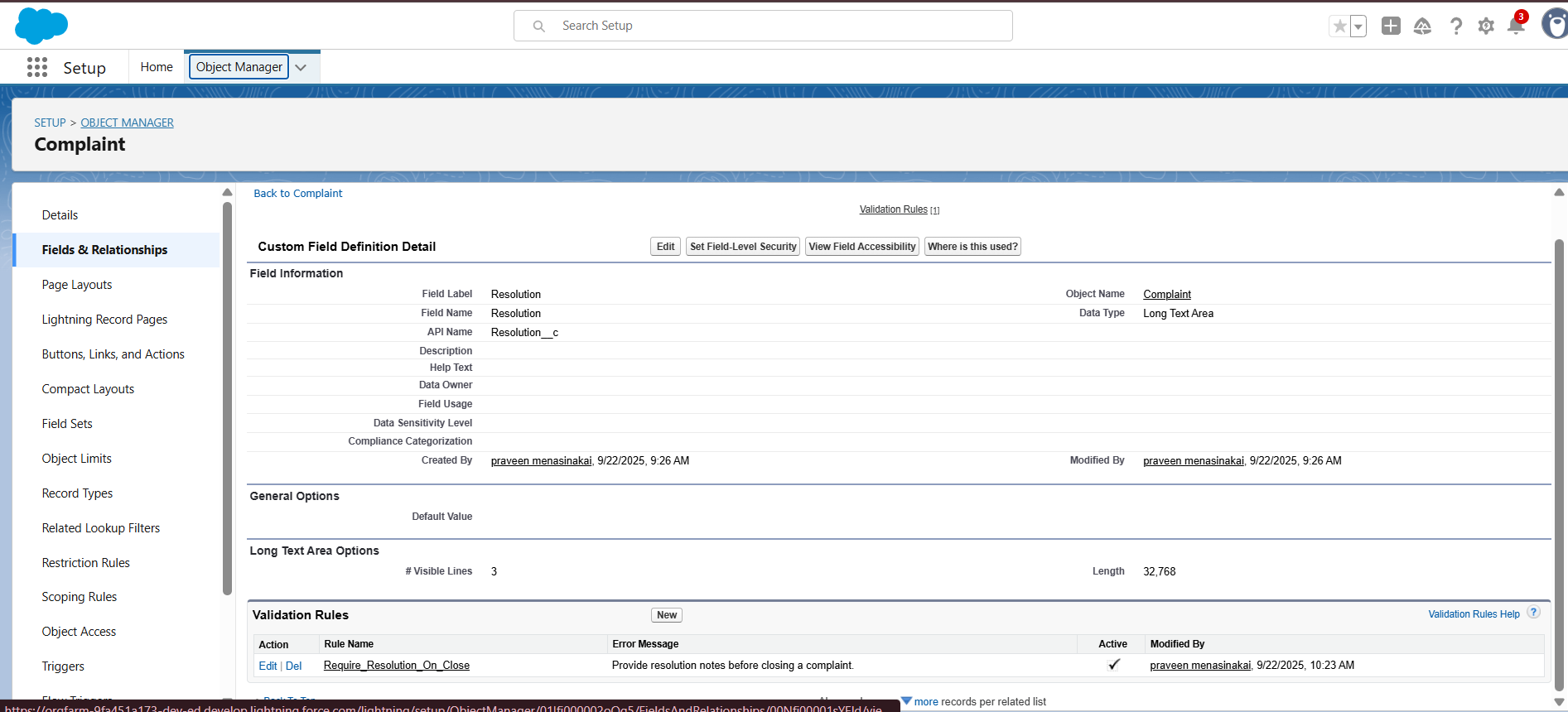
**✅ 5. Test Access**

**✅ Step 1: Login As Test Users**

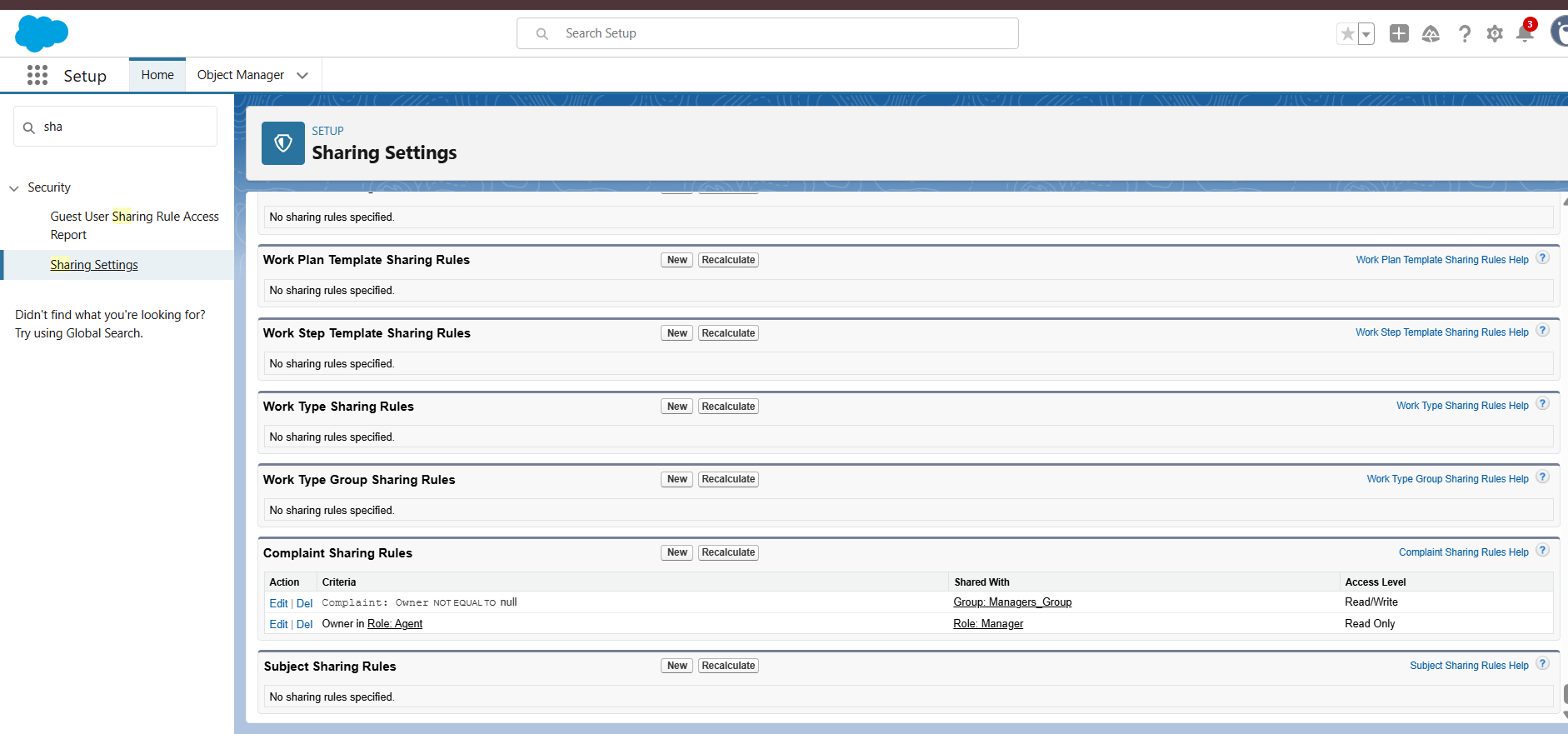
* Go to: Users → [Agent/Manager Test User]
* Click **Login As**

**✅ Step 2: Check Permissions**

| **Test User** | **What to Test** |
| --- | --- |
| Agent | Can Create, Read, Edit complaints Cannot see Resolution\_\_c Cannot delete |
| Manager | Can Read/Edit/Delete Can see Resolution\_\_c Can access agent-owned complaints |
| Admin | Full access to everything |



**Sharing rules and complaint sharing rule:**



**Step 8 — Reports & Dashboard**

**Reports**:

* Complaints by Status
* High Priority Complaints
* Complaints by Branch (Account)

**Dashboard**:

* KPI: Open Complaints
* Chart: Complaints by Status
* Table: High Priority Complaints

**Step 1: Prepare the Reports**

**Dashboards can only use Reports as a data source. So first, we’ll make 3 reports.**

**A. Report for KPI → “Number of Open Complaints”**

1. **App Launcher → Reports → New Report.**
2. **Report Type: Complaints (if not visible, create a Custom Report Type with Complaint as primary object).**
3. **Add Filters:**
   * **Field = Status\_\_c**
   * **Operator = equals**
   * **Value = New, In Progress  
     (Use “Add Filter” and select multiple values).**
4. **Columns: Complaint Number, Subject, Status, Assigned Agent.**
5. **Save Report as: Open Complaints.**

**B. Report for Chart → “Complaints by Branch”**

1. **Reports → New Report → Type = Complaints with Accounts (or custom report Complaint + Account).**
2. **Group rows by: Account\_\_c (Branch).**
3. **Add filter: Status\_\_c not equal to Closed.**
4. **Add chart (top-right “Add Chart” → choose Bar Chart or Pie Chart).**
5. **Save Report as: Complaints by Branch.**

**C. Report for Table → “High Priority Unresolved Complaints”**

1. **Reports → New Report → Type = Complaints.**
2. **Filter:**
   * **Priority\_\_c = High**
   * **Status\_\_c = New, In Progress, Escalated**
3. **Columns: Complaint Number, Subject, Contact, Status, Assigned Agent, SLA Due Date.**
4. **Sort by SLA Due Date ascending (so urgent ones appear first).**
5. **Save Report as: High Priority Complaints.**

**🔹 Step 2: Create the Dashboard**

1. **App Launcher → Dashboards → New Dashboard.**
   * **Name: Complaint Management Dashboard**
   * **Folder: Complaint Dashboards (create if needed).**
   * **Save.**

**🔹 Step 3: Add Components**

1. **Dashboard Designer opens. Click + Component.**

**👉 Component 1: KPI**

* **Source Report: Open Complaints**
* **Visualization: KPI (choose big number).**
* **Label: “Open Complaints.”**
* **Save.**

**👉 Component 2: Chart**

* **Source Report: Complaints by Branch**
* **Visualization: Bar Chart (x-axis = Branch, y-axis = Count).**
* **Label: “Complaints by Branch.”**
* **Save.**

**👉 Component 3: Table**

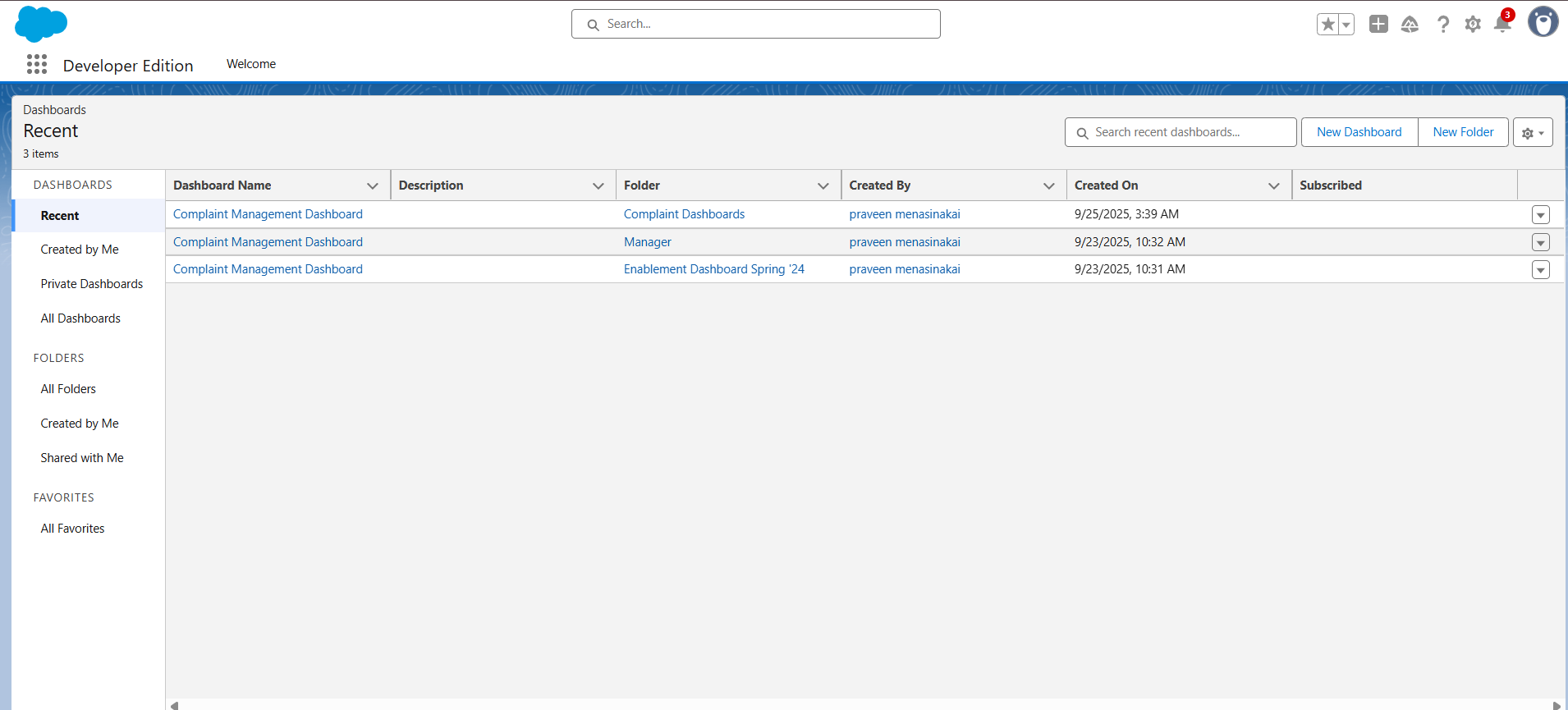
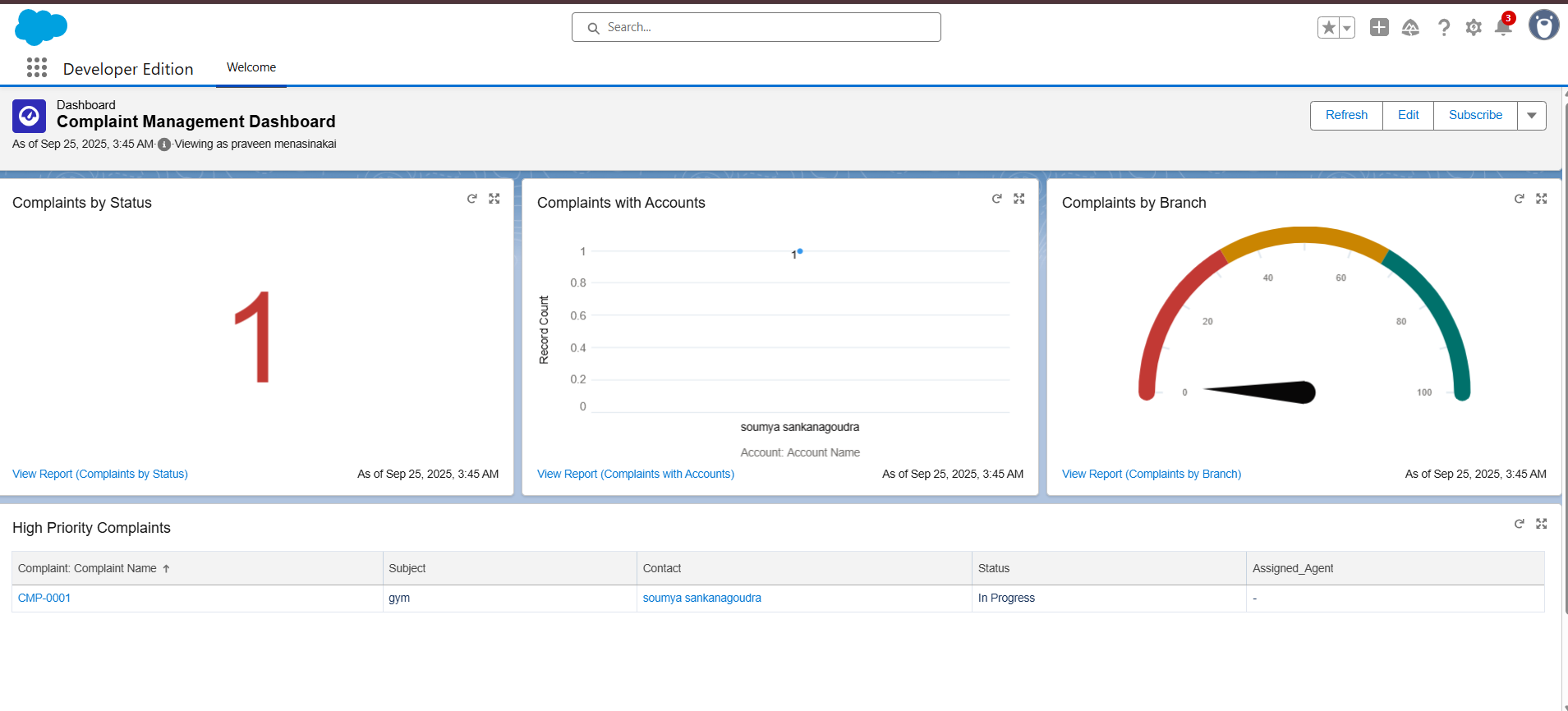
* **Source Report: High Priority Complaints**
* **Visualization: Table**
* **Columns: Complaint Number, Subject, Status, Assigned Agent, SLA Due Date.**
* **Save.**

**🔹 Step 4: Save & Share**

1. **Top-right → Save & Done.**
2. **In Dashboard Folder Sharing, grant View access to Manager users (or Public Group Managers).**

**🔹 Step 5: Test the Dashboard**

1. **Go to App Launcher → Complaints → create sample complaints:**
   * **Complaint 1: Status = New, Priority = Low.**
   * **Complaint 2: Status = In Progress, Priority = High, Account = Branch A.**
   * **Complaint 3: Status = Escalated, Priority = High, Account = Branch B.**
   * **Complaint 4: Status = Closed, Priority = Low.**
2. **Refresh the Dashboard:**
   * **KPI should show 2 open complaints (New + In Progress).**
   * **Chart should show counts per Branch (Branch A: 1, Branch B: 1).**
   * **Table should list only High Priority unresolved complaints (2 & 3).**

**** ****

**Step 9 — Data Import & Sample Data**

**Tools**: Data Import Wizard or Data Loader.  
**Sample CSV Header**:

Name,Contact\_\_c:Contact External ID,Account\_\_c:Account External ID,

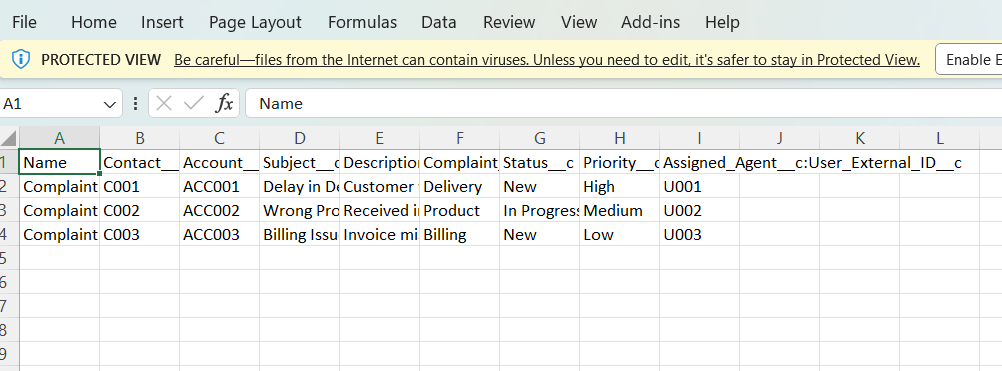
Subject\_\_c,Description\_\_c,Complaint\_Type\_\_c,Status\_\_c,

Priority\_\_c,Assigned\_Agent\_\_c:User External ID

1. Create sample Accounts (Branches) and Contacts manually or via Data Import Wizard:
   * Setup → Data → Data Import Wizard → Standard Objects → Contacts and Accounts CSV.
2. Create sample Complaints manually (App → Complaints → New) or via Data Loader for bulk import.

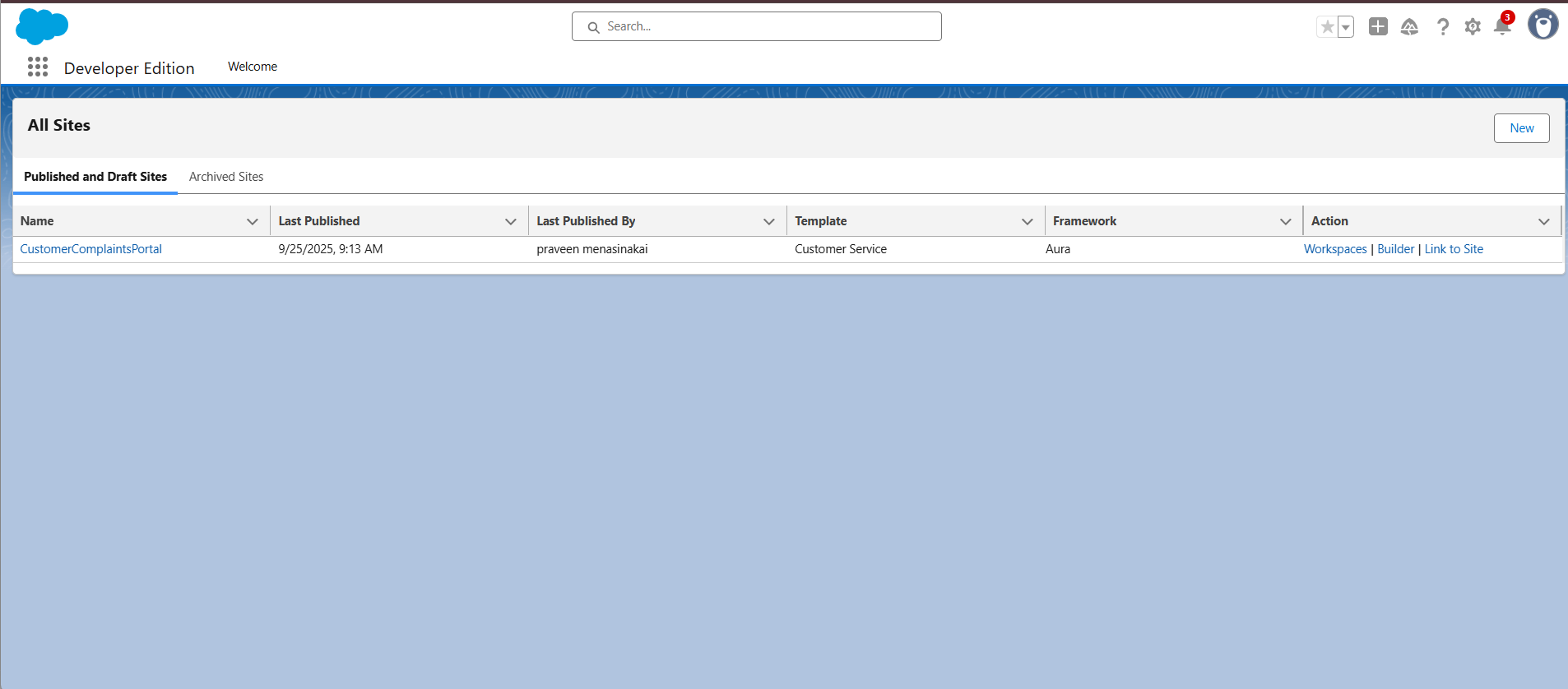
**Sample CSV header for complaints (for Data Loader):**

Name,Contact\_\_c:Contact External ID,Account\_\_c:Account External ID,Subject\_\_c,Description\_\_c,Complai



**Step 10 — Experience Cloud (Optional)**

**Enable Portal**: Setup → All Sites → New → Build Your Own.  
**Sharing Set**: User:ContactId → Complaint:Contact\_\_c (Read/Create).



**Step 11 — Testing Checklist**

* Complaint must have Type.
* Cannot close without Resolution.
* SLA auto-assign and escalation work.
* Duplicate complaints blocked.
* Reports & Dashboard reflect data.
* Permissions & FLS enforce correctly.
* (Optional) Portal users can submit/view only their complaints.

**Step 12 — Handoff & Documentation**

**Deliverables**:

* **User Guide**: Creating, assigning, resolving complaints.
* **Technical Document**: Objects, fields, flows, triggers, permissions.
* **Repository Export**: Metadata via SFDX for deployment.

