

Project Title: “Telecom Service Hub – Customer & SIM Request Management System”

Industry: Telecommunications

Project Type: B2C Salesforce CRM Implementation

Target Users: Customer Support Agents, Service Managers, and Telecom Customers

Problem Statement:

A telecom company is struggling to efficiently manage SIM activations, customer service requests, and cancellations. Customers face delays in activation, poor communication on ticket progress, and agents are overloaded due to manual assignment of service requests. Managers lack real-time visibility into churn rates, customer complaints, and service performance.

The company wants a Salesforce CRM to:

- Automate service request creation and assignment
- Track SIM activations, complaints, and cancellations
- Provide real-time SMS/email notifications to customers
- Offer self-service portals for customers to raise requests
- Deliver dashboards for churn analysis and agent productivity

Use cases:

Customer Management

- Maintain centralized customer profiles with contact details and SIM status.
- Link customers to their subscribed plans.

Service Request Management

- Customers raise activation, complaint, or cancellation requests.
- Requests automatically assigned to available agents.
- Agents update request progress (Open → In Progress → Closed).

SIM Activation & Cancellation

- On activation → CRM auto-sends SMS confirmation.
- On cancellation → SIM status is updated to Inactive automatically.

Plan Management

- Store plan details: Name, Data Limit, Price, Validity.
- Customers linked with their current active plan.

Reporting & Analytics

- Reports: Service Requests by Priority, Requests by Type.
- Dashboard: Active vs Cancelled Customers (Churn Analysis).
- Agent Performance: Requests resolved per agent.

Phase 2: Org Setup & Configuration

Goal of Phase 2

Prepare the Salesforce environment with company setup, users, roles, profiles, and security model so that development can begin smoothly.

1. Salesforce Edition

- **Developer Edition Org** (free) is used.
 - This org acts as both development and testing environment.
 - In real companies, Sandboxes are used, but for this project, Dev Org serves the purpose.
-

2. Company Profile Setup

- **Company Name:** Telecom Service Hub Pvt. Ltd.
- **Time Zone:** (GMT+05:30) Asia/Kolkata
- **Currency:** INR (₹)

- **Locale:** English (India)

“Company Profile setup with org information, timezone, and currency.”

The screenshot shows the Salesforce Setup interface with the 'Company Information' page open. The organization's name is 'Telecom Service Hub Pvt. Ltd.'. Key details shown include:

Organization Detail	Value
Organization Name	Telecom Service Hub Pvt. Ltd.
Primary Contact	OrgFarm EPIC
Division	
Address	Belagavi Ghataprabha 591306 India
Fiscal Year Starts In	January
Activate Multiple Currencies	<input type="checkbox"/>
Enable Data Translation	<input type="checkbox"/>
Newsletter	<input checked="" type="checkbox"/>
Admin Newsletter	<input checked="" type="checkbox"/>
Hide Notices About System Maintenance	<input type="checkbox"/>
Phone	(776) 048-5975
Fax	
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	342 KB (7%) [View]
Used File Space	17 KB (0%) [View]
API Requests, Last 24 Hours	0 (15,000 max)
Streaming API Events, Last 24 Hours	0 (10,000 max)

3. Business Hours & Holidays

- **Standard Business Hours:** Monday–Friday, 9 AM – 6 PM.
- **Holiday Added:** Independence Day (Aug 15).
- Purpose: Ensures service requests and escalations only occur during working hours.

4. Fiscal Year Settings

- Chosen: **Standard Fiscal Year (Jan–Dec).**
- Note: Can be changed to Apr–Mar if required for Indian telecom companies.

5. User Setup & Licenses

Created two sample users to simulate real-world roles:

1. Manager User

- Username: manager1@telecomhub.com
- Role: Service Manager
- License: Salesforce

2. Agent User

- Username: agent1@telecomhub.com

- Role: Customer Support Agent
- License: Salesforce

“Manager and Agent users created with Salesforce licenses.”



Roles	<input type="checkbox"/> Edit <u>EPIC_OrgFarm</u>	<u>OEPIC</u>	<u>epic.ec40c284a3a3@orgfarm.salesforce.com</u>	<input checked="" type="checkbox"/> <u>System Administrator</u>
User Management Settings	<input type="checkbox"/> Edit <u>koparde_savitri</u>	<u>say</u>	<u>savitrimkoparde116@agentforce.com</u>	<input checked="" type="checkbox"/> <u>System Administrator</u>
Users	<input type="checkbox"/> Edit <u>Telecom_Agent1</u>	<u>agent1</u>	<u>agent1@telecomhub.com</u>	<u>Customer support Agent</u> <input checked="" type="checkbox"/> <u>Telcom Agent Profile</u>
	<input type="checkbox"/> Edit <u>Telecom_Maneger</u>	<u>mtele</u>	<u>manager1@telecomhub.com</u>	<u>Service manager</u> <input checked="" type="checkbox"/> <u>Telcom Manager Profile</u>

6. Profiles

Profiles were created to define object-level permissions:

- **Telecom Agent Profile**
 - Access: Create/Edit Service Requests, Read-only on Customers & Plans.
- **Telecom Manager Profile**
 - Full Access to all objects (Customers, Plans, Service Requests).

7. Roles

Role hierarchy created to manage record-level visibility:

- **Service Manager** (Top role)
 - **Customer Support Agent** (Child role)

Assignments:

- Manager User → Service Manager
- Agent User → Customer Support Agent

Effect: Manager can see all records; Agent can only see their own.

“Role hierarchy created for Service Manager and Agents.”

The screenshot shows the Salesforce Setup Roles page. The left sidebar has a search bar for 'roles' and sections for 'Users' (selected), 'Feature Settings', 'Sales' (Contact Roles on Contracts, Contact Roles on Opportunities), 'Service' (Case Teams, Case Team Roles, Contact Roles on Cases), and a note about global search. The main content area is titled 'Creating the Role Hierarchy' and shows the 'Your Organization's Role Hierarchy'. It lists 'Telecom Service Hub Pvt. Ltd.' with an 'Add Role' option. Under it are 'CEO' (with Edit, Del, Assign buttons) and 'Service manager' (with Edit, Del, Assign buttons). 'Service manager' also has a 'Customer support Agent' child node with its own 'Edit', 'Del', and 'Assign' buttons. A 'Show in tree view' dropdown is at the top right.

8. Org-Wide Defaults (OWD)

Defined baseline data visibility:

- **Customer_c** → Public Read Only
- **Plan_c** → Public Read Only
- **ServiceRequest_c** → Private

This ensures sensitive service requests are only visible to the owner and their manager.

Phase 3: Data Modeling & Relationships

Goal of Phase 3

To design and configure the **data model** in Salesforce for managing **customers, telecom plans, and service requests**. This includes creating objects, fields, relationships, and layouts that support the real-time telecom use cases.

1. Standard & Custom Objects

Standard Objects

- **User** → For Agents and Managers who handle requests.

Custom Objects

1. **Customer_c** → Stores customer details and SIM information.

2. **Plan__c** → Stores telecom plan details (price, data, validity).
 3. **ServiceRequest__c** → Tracks service requests such as activation, complaints, or cancellations.
-

2. Fields

Customer__c

- Phone (Phone)
- Email (Email)
- SIM Number (Text, 20)
- Status (Picklist: Active, Inactive, Suspended)
- Region (Picklist: East, West, North, South)

SETUP > OBJECT MANAGER Customer					
Details	Fields & Relationships				
	Fields & Relationships				
	9 Items, Sorted by <i>Created</i>				
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
	Created By	CreatedById	Lookup(User)		
	Customer Name	Name	Text(80)	✓	▼
	Email	Email__c	Email	▼	▼
	Last Modified By	LastModifiedById	Lookup(User)	✓	▼
	Owner	OwnerId	Lookup(User,Group)	✓	▼
	Phone	Phone__c	Phone	▼	▼
Page Layouts					
Lightning Record Pages					
Buttons, Links, and Actions					
Compact Layouts					
Field Sets					
Object Limits					
Record Types					
Related Lookup Filters					
Search Layouts					
List View Button Layout					
Restriction Rules					
Scoping Rules					
Object Access					

Plan__c

- Price (Currency)
- Validity (Number, 3 digits)
- Data Limit (Number, 3 digits, GB)
- Plan Type (Picklist: Prepaid, Postpaid, Broadband)

SETUP > OBJECT MANAGER

Plan

Details	Fields & Relationships				
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Page Layouts	Created By	CreatedById	Lookup(User)		
Lightning Record Pages	Data Limit	Data_Limit__c	Number(3, 0)		
Buttons, Links, and Actions	Last Modified By	LastModifiedById	Lookup(User)		
Compact Layouts	Owner	OwnerId	Lookup(User,Group)	✓	
Field Sets	Plan Name	Name	Text(80)	✓	
Object Limits	Plan Type	Plan_Type__c	Picklist		
Record Types	Price	Price__c	Currency(18, 0)		
Related Lookup Filters	Validity	Validity__c	Number(3, 0)		
Restriction Rules					
Scoping Rules					
Object Access					
Triggers					
Flow Triggers					

ServiceRequest__c

- Request Type (Picklist: Activation, Cancellation, Network Issue, Billing)
- Status (Picklist: New, In Progress, Resolved, Cancelled)
- Priority (Picklist: High, Medium, Low)
- Description (Long Text Area)

Details	Fields & Relationships				
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Page Layouts	Created By	CreatedById	Lookup(User)		
Lightning Record Pages	Customer	Customer__c	Lookup(Customer)	✓	
Buttons, Links, and Actions	Customer Email.	Customer_Email__c	Formula (Text)		
Compact Layouts	Description	Description__c	Long Text Area(32768)		
Field Sets	Last Modified By	LastModifiedById	Lookup(User)		
Object Limits	Owner	OwnerId	Lookup(User,Group)	✓	
Record Types	Priority	Priority__c	Picklist		
Related Lookup Filters	Record Type	RecordTypeId	Record Type	✓	
Search Layouts	Request Number (Auto Number → Format SR-	Name	Text(80)	✓	
List View Button Layout	Request Type	Request_Type__c	Picklist		
Restriction Rules					
Scoping Rules					
Object Access					

3. Relationships

- **Customer → Plan → Lookup** (a customer subscribes to a plan).
 - **ServiceRequest → Customer → Lookup** (each request belongs to a customer).
-

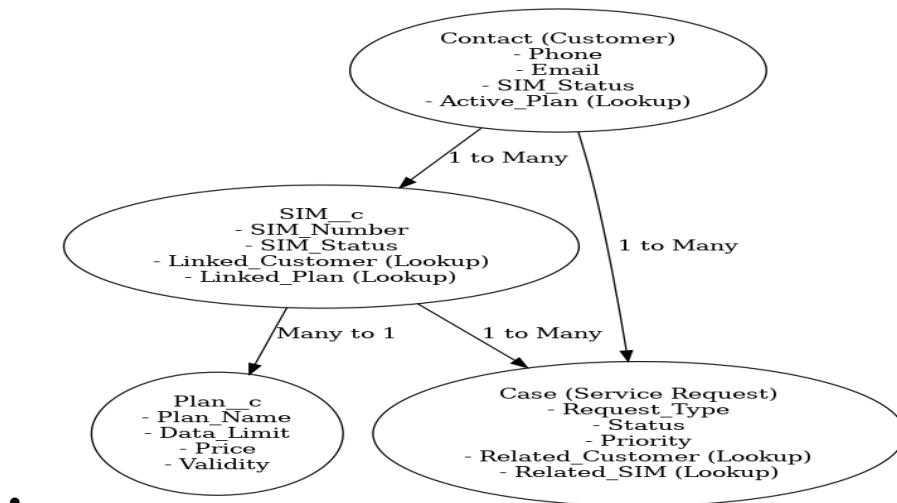
4. Page Layouts

- **Customer Layout** → Phone, Email, SIM Number, Status, Region.
 - **Plan Layout** → Price, Validity, Data Limit, Plan Type.
 - **Service Request Layout** → Request Type, Status, Priority, Description + related Customer.
-

5. Schema Builder

Schema Builder used to visualize objects and their relationships:

- Customer__c linked to Plan__c (Lookup).
- ServiceRequest__c linked to Customer__c (Lookup).



6. Record Types

Record Types (ServiceRequest__c)

Created **different record types** to simplify request handling:

- **Activation Request** → For SIM activations.
- **Complaint Request** → For issues like network/billing.

- **Cancellation Request** → For SIM termination.

This helps agents capture the right type of request quickly.

7. Compact Layouts

Optimized for Salesforce Mobile:

- **Customer__c Compact Layout** → Shows Name, Phone, SIM Number, Status.

The screenshot shows the 'Customer' object in the Object Manager. The 'Compact Layouts' tab is selected in the sidebar. A compact layout named 'Customer Compact Layout' is displayed, which includes fields for Name, Phone, SIM Number, and Status. The layout is assigned to the 'Customer' object.

Label	Customer Compact Layout	Object Name	Customer
API Name	Customer__Compact_Layout		
Included Fields	Name Phone SIM Number Status		
Created By	savitri.koparde, 9/21/2025, 11:48 PM	Modified By	savitri.koparde, 9/21/2025, 11:48 PM

- **Plan__c Compact Layout** → Shows Name, Price, Plan Type.

The screenshot shows the 'Plan' object in the Object Manager. The 'Compact Layouts' tab is selected in the sidebar. A compact layout named 'Plan Compact Layout' is displayed, which includes fields for Price, Plan Type, and Plan Name. The layout is assigned to the 'Plan' object.

Label	Plan Compact Layout	Object Name	Plan
API Name	Plan__Compact_Layout		
Included Fields	Price Plan Type Plan Name		
Created By	savitri.koparde, 9/22/2025, 12:21 AM	Modified By	savitri.koparde, 9/22/2025, 12:21 AM

- **ServiceRequest__c Compact Layout** → Shows Request Type, Status, Priority.

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

- Header:** Search Setup, Home, Object Manager
- Breadcrumb:** SETUP > OBJECT MANAGER
- Section:** Service Request
- Left Sidebar:** Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, **Compact Layouts** (selected), Field Sets, Object Limits, Record Types, Related Lookup Filters.
- Main Content:** Service Request Compact Layout (Service Request Compact Layout) with the URL [+ Back to Service Request](#).
 - Compact Layout Detail:** Label: Service Request Compact Layout, API Name: Service_Request_Compact_Layout, Object Name: Service Request.
 - Included Fields:** Request Type, Status, Priority, Customer.
 - Timestamps:** Created By: savitri koparde, 9/21/2025, 10:10 PM; Modified By: savitri koparde, 9/21/2025, 10:39 PM.
- Buttons:** Edit, Clone, Delete, Compact Layout Assignment.

Phase 4 Report: Must-Have Process Automation (Admin)

Goal

Automate critical telecom workflows in Salesforce to reduce manual work, improve service turnaround, and keep customers informed in real time.

1. Validation Rules

Purpose: Ensure data integrity and prevent incomplete/incorrect records.

- **Customer Email Validation**
 - Formula: NOT(CONTAINS>Email, "@")
 - Error: “Enter a valid Email Address.”

The screenshot shows the Salesforce Setup interface for the Customer object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, and Triggers. The main content area displays the 'Customer Validation Rule' detail page for a rule named 'Email_Format_Check'. The rule is active and has the formula `NOT(CONTAINS>Email__c, "@")`. The error message is 'Please enter a valid email address.' and it is located in the Email field.

Validation Rule Detail	
Rule Name	Email_Format_Check
Error Condition Formula	NOT(CONTAINS>Email__c, "@")
Error Message	'Please enter a valid email address.'
Description	
Created By	savitri koparde, 9/22/2025, 8:32 PM
Modified By	savitri koparde, 9/22/2025, 8:32 PM

- **Service Request Priority Validation**

- Formula: `AND(ISPICKVAL(Request_Type__c, "Network Issue"), ISBLANK(TEXT(Priority__c)))`
- Error: "Priority is required for Network Issues."

Impact: Prevents bad customer data and enforces business rules.

Setup Home Object Manager

SETUP > OBJECT MANAGER Service Request

Service Request Validation Rule

[Back to Service Request](#)

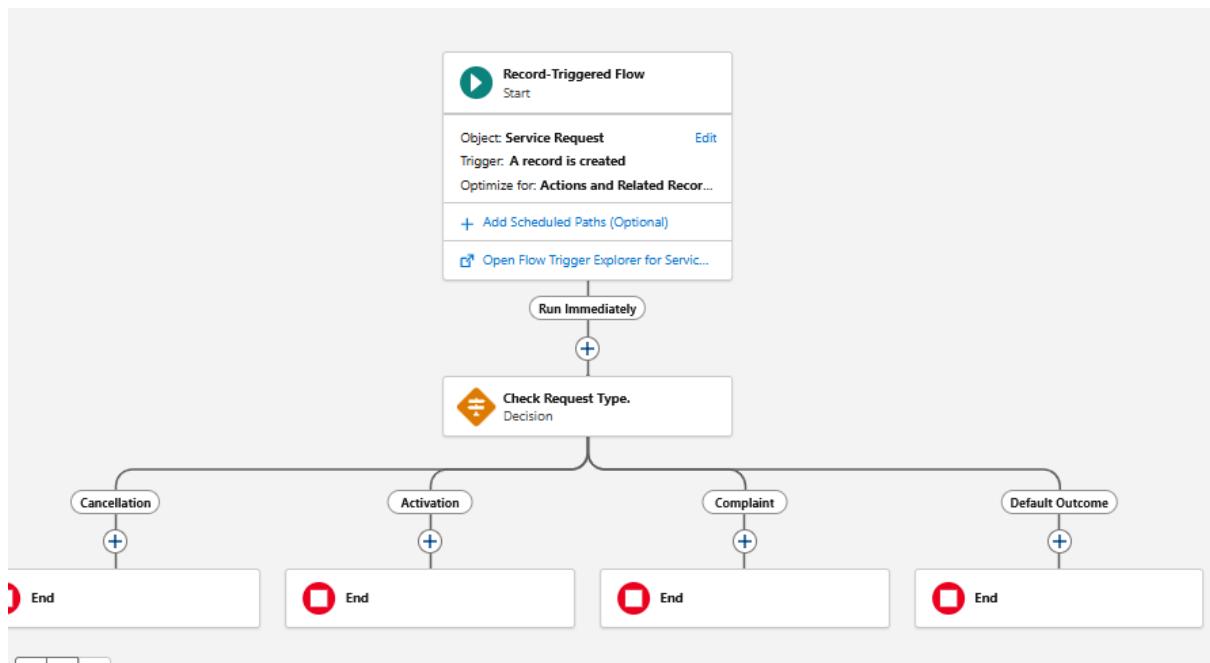
Validation Rule Detail		Active
Rule Name	Priority_Required_For_Network_Issues	<input checked="" type="checkbox"/>
Error Condition Formula	AND(ISPOKVAL(Request_Type__c, "Network Issue"), ISBLANK(TEXT(Priority__c)))	
Error Message	"Priority is required when Request Type = Network Issue."	Error Location Priority
Description		
Created By	savitri kogarde, 9/22/2025, 8:34 PM	Modified By savitri kogarde, 9/22/2025, 8:34 PM
	Edit Clone	

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers

2. Flow Builder (Record-Triggered)

Purpose: Automate request handling and SIM status updates.

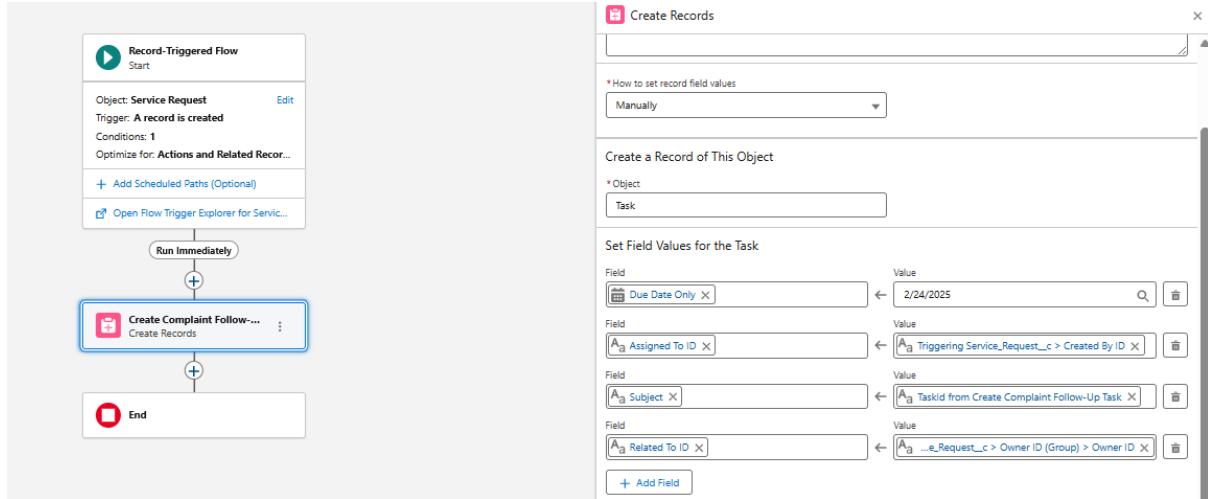
- **Round-Robin Assignment Flow**
 - Trigger: On ServiceRequest__c creation.
 - Logic: Distributes requests evenly among active Agents.
 - Outcome: No manual assignment needed, balanced workload.



- **SIM Cancellation Flow**

- Trigger: When Request_Type = Cancellation.
- Logic: Auto-updates Customer__c.Status = Inprogress.
- Outcome: SIM instantly deactivated, no agent action required.

Impact: Fast response, reduced manual errors, higher customer satisfaction.



3. Email Alerts

Purpose: Keep customers informed automatically.

- **Activation Confirmation** → “Your SIM {SIM Number} has been activated successfully.”
- **Cancellation Confirmation** → “Your SIM has been deactivated.”

Impact: Customers receive real-time updates, boosting trust.

SETUP **Classic Email Templates**

Custom Email Template
SIM_Cancellation_Notification

Preview your email template below.

Email Template Detail	
Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	SIM_Cancellation_Notification
Template Unique Name	SIM_Cancellation_Notification
Encoding	Unicode (UTF-8)
Author	savitri koparde [Change]
Description	
Created By	savitri koparde, 9/25/2025, 12:42 AM
Modified By	savitri koparde, 9/25/2025, 12:45 AM

[Edit Properties](#) [Edit HTML Version](#) [Edit Text Version](#) [Delete](#) [Clone](#)

Email Template

[Send Test and Verify Merge Fields](#)

Subject | Your SIM has been cancelled — Telecom Service Hub

HTML Preview

Hi {ServiceRequest__c.Customer__r.Name}, We confirm that your SIM (SIM Number: {ServiceRequest__c.Customer__r.SIM_Number__c}) has been cancelled on {TEXT(ServiceRequest__c.CreatedDate)}. If you need help, call 1800-XXX-XXXX. Regards, Telecom Service Hub Support Team

SETUP **Email Alerts**

Email Alert
Notify customer on SIM Cancellation

[Rules Using This Email Alert](#) | [Approval Processes Using This Email Alert](#) | [Enrollment Processes Using This Email Alert](#)

Email Alert Detail	
Description	Notify customer on SIM Cancellation
Unique Name	Notify_customer_on_SIM_Cancellation
From Email Address	Current User's email address
Recipients	User_Agent1_Telecom User_Agent2_Telecom
Additional Emails	savitrikoparde012@gmail.com
Created By	savitri koparde, 9/25/2025, 12:59 AM
Modified By	savitri koparde, 9/25/2025, 5:13 AM

[Edit](#) [Delete](#) [Clone](#)

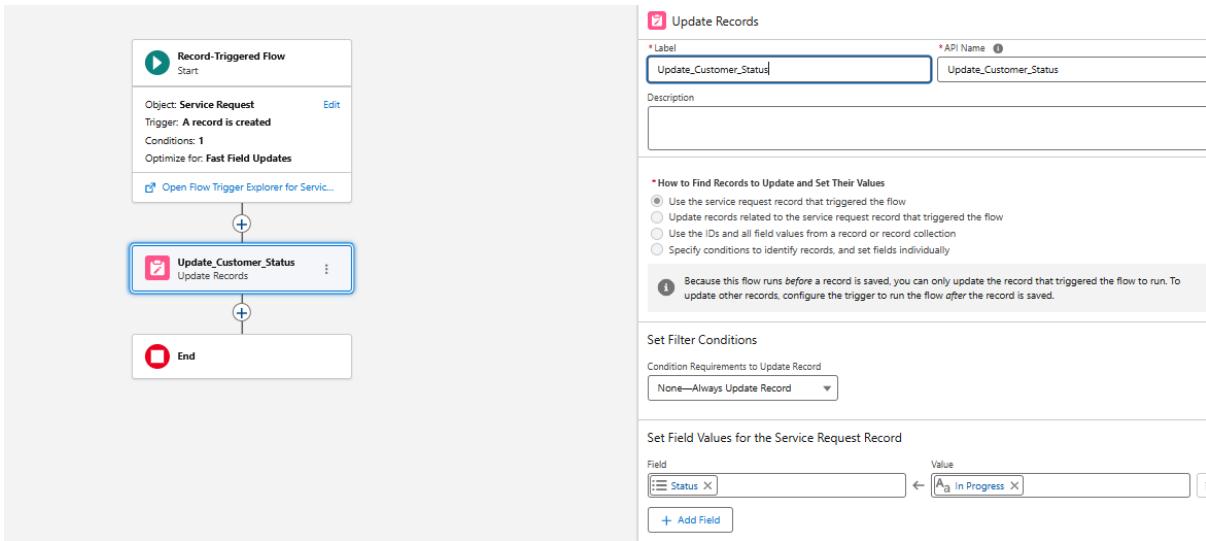
Email Template **Object** **Service Request**

4. Field Updates

Purpose: Maintain accurate record states automatically.

- **Resolved Request Auto-Close**
 - Condition: When ServiceRequest__c.Status = Resolved.
 - Action: Update Status = Closed.

Impact: Ensures consistent lifecycle tracking of service requests.



5. Tasks

Purpose: Drive timely follow-ups for customer complaints.

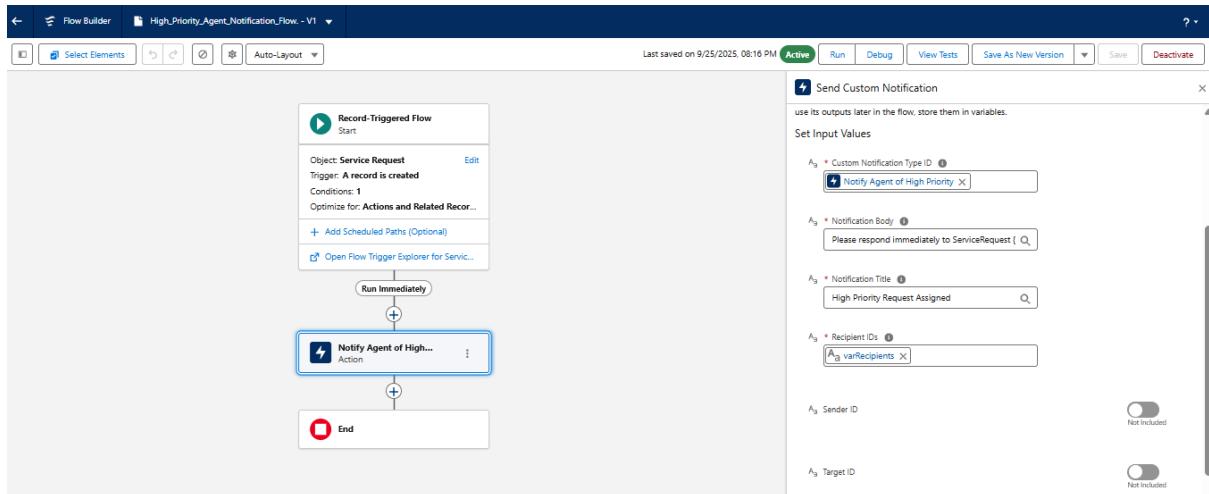
- **Complaint Follow-Up Task**
 - Trigger: When Request_Type = Complaint.
 - Auto-created Task:
 - Subject: “Follow up with Customer”
 - Due Date: 2 hours from request creation.

Impact: Agents act quickly on complaints, reducing churn risk.

6. Custom Notifications

Purpose: Escalate critical service requests to Agents/Managers.

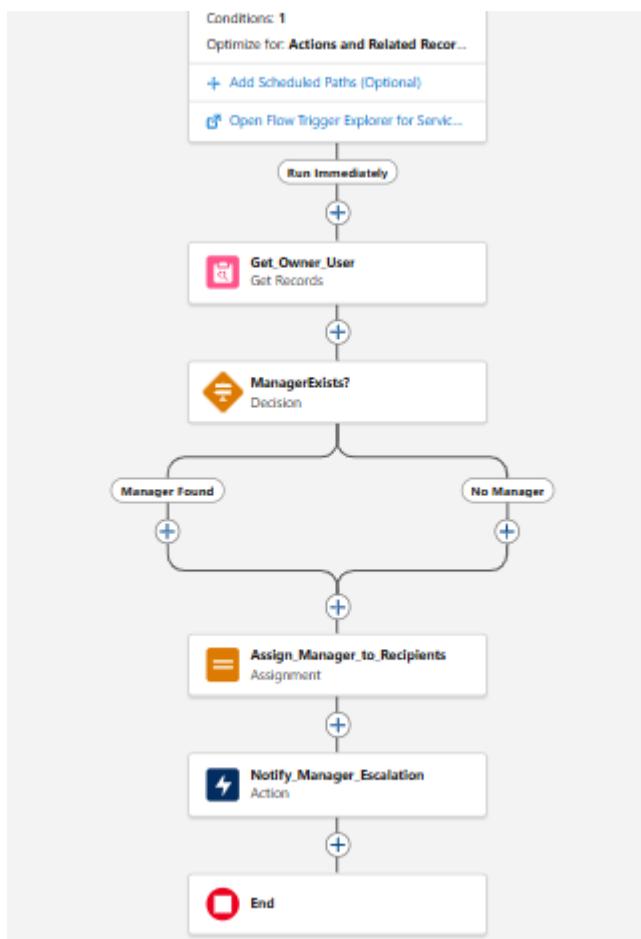
- **Agent Notification:**
 - Trigger: When High Priority Service Request assigned.
 - Message: “⚠️ High Priority Request Assigned: Please respond immediately.”



- **Manager Notification:**

- Trigger: When approval or escalation required.
- Message: “Approval required for ServiceRequest #{{!Id}}.”

Impact: Ensures immediate attention to high-risk cases.



Phase 5: Apex Programming (Developer)

Goal

Enable advanced customization in Salesforce using **Apex programming** to handle scenarios not possible with declarative automation. This includes writing triggers, classes, queries, and test code for maintaining data integrity and automating complex telecom workflows.

1. Apex Classes & Objects

Implementation:

- Created CustomerHelper.cls with reusable logic to update Customer status when a Service Request is processed.

```
public class CustomerHelper {  
    public static void deactivateCustomer(Id custId) {  
        try {  
            Customer__c cust = [SELECT Id, Status__c FROM Customer__c WHERE Id = :custId LIMIT 1];  
            cust.Status__c = 'Inactive';  
            update cust;  
        } catch (Exception e) {  
            System.debug('Error while deactivating customer: ' + e.getMessage());  
        }  
    }  
}
```

Purpose: Centralized business logic for reusability.

The screenshot shows the Salesforce Setup interface with the 'Apex Classes' section selected. The page displays the 'CustomerHelper' class. The code is as follows:

```

1 public class CustomerHelper {
2     public static void deactivateCustomers(Set<Id> custIds) {
3         if (custIds == null || custIds.isEmpty()) return;
4
5         List<Customer> customersToUpdate = [
6             SELECT Id, Status__c
7             FROM Customer
8             WHERE Id IN :custIds
9         ];
10
11        for (Customer c : customersToUpdate) {
12            c.Status__c = 'Inactive';
13        }
14
15        if (!customersToUpdate.isEmpty()) {
16            try {
17                update customersToUpdate;
18            } catch (DmlException e) {
19                System.debug('Error: ' + e.getMessage());
20            }
21        }
22    }
23
24    public static void deactivateCustomer(Id custId) {

```

2. Apex Trigger with Design Pattern

Implementation:

- Created trigger ServiceRequestTrigger on ServiceRequest__c.
- Delegated logic to ServiceRequestTriggerHandler.cls.

Trigger:

```

trigger ServiceRequestTrigger on ServiceRequest__c (after insert, after update) {

    if(Trigger.isAfter && Trigger.isInsert) {

        ServiceRequestTriggerHandler.handleAfterInsert(Trigger.new);

    }
}

```

Handler Class:

```

public class ServiceRequestTriggerHandler {

    public static void handleAfterInsert(List<ServiceRequest__c> newReqs) {

```

```

Set<Id> custIds = new Set<Id>();

for(ServiceRequest__c req : newReqs) {

    if(req.Request_Type__c == 'Cancellation' && req.Customer__c != null) {

        custIds.add(req.Customer__c);

    }

}

if(!custIds.isEmpty()) {

    for(Id cId : custIds) {

        CustomerHelper.deactivateCustomer(cId);

    }

}

}

```

Purpose: Updates related Customer to “Inactive” when a cancellation request is created.

The screenshot shows the Salesforce Setup Apex Triggers page. The left sidebar has sections for Email, Custom Code, Environments, and Jobs. Under Custom Code, Apex Classes is expanded, showing Apex Exception Email, Apex Settings, Apex Test Execution, and Apex Test History. Under Jobs, Apex Flex Queue and Apex Jobs are listed. A search bar at the top finds 'Ape'. The main content area is titled 'Apex Triggers' and displays a message about Apex usage. Below is a table of triggers:

Action	Name	Namespace Prefix	sObject Type	API Version	Status	Size Without Comments	Last Modified By
Edit Del	ServiceRequestTrigger	Service_Request	Service_Request	64.0	Active	72	savitri koparde, 9/25/2025, 10:05 PM
Edit Del	ServiceRequestTriggerHandler	Service_Request	Service_Request	64.0	Active	79	savitri koparde, 9/25/2025, 10:50 PM
Edit Del	SRTtrigger	Service_Request	Service_Request	64.0	Active	60	savitri koparde, 9/25/2025, 10:56 PM

3. SOQL Query

- Used in CustomerHelper.cls to fetch Customer details:

```
Customer__c cust = [SELECT Id, Status__c FROM Customer__c WHERE Id = :custId LIMIT 1];
```

Purpose: Retrieve customer data from Salesforce for processing.

4. Collections (List, Set, Map)

- **Set:** custIds ensures only unique Customers are updated.
- **List:** Trigger passes list of new ServiceRequests.
- **Map:** Could be used to map Customer Id → Customer record for bulk operations.

Example:

```
Map<Id, Customer__c> custMap = new Map<Id, Customer__c>(  
    [SELECT Id, Status__c FROM Customer__c WHERE Id IN :custIds]  
)
```

5. Control Statements

- **If condition** checks request type.
- **For loop** iterates over trigger records.

Example:

```
for(ServiceRequest__c req : newReqs) {  
    if(req.Request_Type__c == 'Cancellation') {  
        // logic  
    }  
}
```

6. Exception Handling

- Used try-catch in CustomerHelper.cls:

```
try {
```

```
update cust;  
} catch (Exception e) {  
    System.debug('Error: ' + e.getMessage());  
}
```

Purpose: Ensures errors don't break transaction flow.

7. Test Class

Implementation:

```
@isTest  
  
public class TestServiceRequestTrigger {  
  
    @isTest  
  
    static void testCancellationFlow() {  
  
        // Create test customer  
  
        Customer__c cust = new Customer__c(  
            Name = 'Test Customer',  
            Email__c = 'test@test.com',  
            SIM_Number__c = '12345',  
            Status__c = 'Active'  
        );  
  
        insert cust;  
  
        // Create cancellation request  
  
        ServiceRequest__c req = new ServiceRequest__c(  
            Request_Type__c = 'Cancellation',  
            Customer__c = cust.Id,  
            Status__c = 'New'  
        );  
  
        insert req;
```

```

// Fetch updated customer

Customer__c updatedCust = [SELECT Status__c FROM Customer__c WHERE Id = :cust.Id];

System.assertEquals('Inactive', updatedCust.Status__c);

}

}

```

Purpose: Validates that trigger and helper class logic works correctly. Achieves code coverage required for deployment.

Phase 6: User Interface Development

Goal:

Provide a functional Salesforce Lightning UI for Agents and Managers to manage Customers, SIM activations/cancellations, and Service Requests efficiently, with only essential features.

1. Lightning App

Purpose: Central workspace for Agents and Managers.

Steps:

1. Setup → **App Builder** → **Apps** → **New Lightning App**
2. Enter Details:
 - Name: **Telecom Service Hub**
 - Description: Telecom CRM for Customer & SIM Request Management
 - Logo & Theme (optional)
3. Navigation: **Standard Navigation**
4. Add Tabs:
 - **Customers (Customer__c)**
 - **Service Requests (Service_Request__c)**
 - **Plans (Plan__c)**
5. Assign App to **Agent** and **Manager** profiles
6. Click **Finish** → **Save**

Impact: Users have a single workspace with quick access to main objects.

The image shows two screenshots of a Salesforce interface. The top screenshot is the 'App Launcher' window, which has a search bar at the top and a list of apps below. The apps listed are 'Salesforce Chatter', 'Content', and 'Telecom Service Hub'. The bottom screenshot shows a custom object page for 'Telecom Service Hub'. It features a logo for 'Telecom services', a search bar, and a navigation bar with tabs for 'Plans', 'Customers', and 'Service Requests'. The page has a blue decorative banner at the bottom.

2. Tabs

Purpose: Quick access to main objects.

Steps:

- Ensure **Custom Object Tabs** exist for Customer__c and Service_Request__c
- Use existing tab for Plan__c

Impact: Objects are accessible from the app navigation.

The image shows the 'Custom Object Tabs' configuration screen. It lists four custom object tabs: 'Assignment Counters', 'Customers', 'Plans', and 'Service Requests'. Each tab is associated with an 'Edit | Del' link and a 'Label'. To the right of the tabs, there is a 'Tab Style' column showing four colored bars with corresponding icons: a green bar for 'Assignment Counters' (labeled 'Triangle'), an orange bar for 'Customers' (labeled 'People'), a dark blue bar for 'Plans' (labeled 'Compass'), and a purple bar for 'Service Requests' (labeled 'Computer').

3. Record Pages

Purpose: Customize layout for clarity and efficiency.

Steps:

1. Setup → Object Manager → Customer / Service_Request__c → Lightning Record Page
2. Add Components:
 - **Highlights Panel:** Key fields like Status, Plan, Priority
 - **Related Lists:** Service Requests per Customer
3. Assign Lightning Page to **Agent** and **Manager** profiles
4. Save & Activate

Impact: Shows important information clearly for workflow efficiency.

LABEL	ORG DEFAULT	APP DEFAULT	OTHER ASSIGNMENTS	MODIFIED BY
Service Record Page		Desktop (1), Phone (1)		savitri koparde, 9/26/2025, 4:29 AM

Structure:

Phase 7: Integration & External Access – Must-Have Documentation

Goal: Enable Salesforce to securely communicate with external systems, exchange data, and automate processes for the Telecom Service Hub project.

1. Named Credentials

Purpose: Store external API authentication securely for callouts. **Steps:**

1. Setup → Named Credentials → New Named Credential

Fill in: Label: TelecomAPI

URL: <https://external-service.example.com>

Identity Type: Named Principal

Authentication: Password or OAuth

Impact: Makes external API calls secure and easy to manage.

Named Credentials:

The screenshot shows the 'SETUP > NAMED CREDENTIALS' section. A 'TelecomAPI' credential is selected. The 'Edit' and 'Delete' buttons are visible. The 'TelecomAPI' row contains fields for Label (TelecomAPI), Name (TelecomAPI), URL (<https://external-service.example.com>), and Enabled for Callouts (checked). The 'Authentication' section lists 'External Credential' (Telecom_Auth) and 'Client Certificate'. The 'Callout Options' section includes 'Generate Authorization Header' (unchecked), 'Allow Formulas in HTTP Header' (checked), and 'Allow Formulas in HTTP Body' (unchecked).

Label	Name
TelecomAPI	TelecomAPI

Authentication

External Credential: [Telecom_Auth](#)

Client Certificate

Callout Options

Generate Authorization Header

Allow Formulas in HTTP Header

Allow Formulas in HTTP Body

SETUP > NAMED CREDENTIALS

Telecom_Auth

Edit **Delete**

Label	Name
Telecom_Auth	Telecom_Auth
Authentication Protocol	
Custom	
Managed Package Access	
Created By Namespace ?	

Related Named Credentials

Label	Name	URL
TelecomAPI	TelecomAPI	https://external-service.example.com

Principals

Se...	Parameter Name	A...	Authentication Status	Actions
1	NamedPrincipal	2	Unknown	New

Developer Console (Apex code)

Developer Console - Google Chrome

orgfarm-11cac0a6a9-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

Log executeAnonymous @9/26/2025, 8:32:03 PM

Execution Log

Timestamp	Event	Details
20:32:03:002	USER_INFO	[EXTERNAL] 005gL0000087PrW savitrikoparde116@agentforce.com (GMT-07:00) Pacific Daylight Time (America/Los_Angeles)
20:32:03:002	EXECUTION_ST...	
20:32:03:002	CODE_UNIT_ST...	[EXTERNAL] execute_anonymous_apex
20:32:03:002	VARIABLE_SCO...	[6] http System.Http true false
20:32:03:002	VARIABLE_SCO...	[1] req System.HttpRequest true false
20:32:03:002	VARIABLE_SCO...	[7] res System.HttpResponse true false
20:32:03:002	HEAP_ALLOCATE	[95] Bytes:3
20:32:03:002	HEAP_ALLOCATE	[100] Bytes:152
20:32:03:002	HEAP_ALLOCATE	[417] Bytes:408
20:32:03:002	HEAP_ALLOCATE	[430] Bytes:408
20:32:03:002	HEAP_ALLOCATE	[317] Bytes:6
20:32:03:002	HEAP_ALLOCATE	[EXTERNAL] Bytes:15
20:32:03:002	STATEMENT_EX...	[1]
20:32:03:002	STATEMENT_EX...	[1]

This Frame Executable Debug Only Filter [Click here to filter the log](#)

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
savitri koparde	Unknown	/services/data/v64...	9/26/2025, 8:32:0...	Success		8.5 KB

2. Remote Site Settings

Purpose

If using legacy authentication (without Named Credentials), Salesforce requires whitelisting the external API domain.

Steps

1. Go to **Setup** → **Remote Site Settings** → **New Remote Site**.
2. Enter:
 - **Remote Site Name:** TelecomAPI
 - **Remote Site URL:** <https://external-service.example.com>
3. Save

Impact

Ensures Salesforce allows outbound calls to the external API.

The screenshot shows the 'Remote Site Settings' page in Salesforce. At the top, there's a 'SETUP' button and a 'Remote Site Settings' section. Below that, a 'Remote Site Details' section is displayed. The 'Remote Site Detail' table contains the following data:

Remote Site Detail		Edit	Delete	Clone
Remote Site Name	TelecomAPI	Modified By savitri koparde , 9/26/2025, 8:52 AM		
Remote Site URL	https://external-service.example.com			
Disable Protocol Security	<input type="checkbox"/>			
Description	Telecom Service Hub API Endpoint.			
Active	<input checked="" type="checkbox"/>			
Created By	savitri koparde , 9/26/2025, 8:52 AM	Edit	Delete	Clone

3. Sample REST Callout (Apex Integration)

Purpose

Demonstrates how Salesforce can fetch data from an external system.

✖ Code Example

```
public class TelecomAPIService {  
  
    public static void fetchCustomerData() {  
  
        Http http = new Http();  
  
        HttpRequest request = new HttpRequest();  
  
        // Use Named Credential  
  
        request.setEndpoint('callout:TelecomAPI/customers');  
  
        request.setMethod('GET');  
  
  
        HttpResponse response = http.send(request);  
  
  
        if (response.getStatusCode() == 200) {  
  
            System.debug('Response: ' + response.getBody());  
  
        } else {  
  
            System.debug('Error: ' + response.getStatus());  
  
        }  
    }  
}
```

} Impact

- Shows Salesforce making a secure outbound call.
- This can later be extended to fetch Service Requests, Plans, or SIM status.

The screenshot shows the Salesforce Developer Console interface. The top bar displays "Developer Console - Google Chrome" and the URL "orgfarm-11cac0a6a9-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage". Below the header, there are tabs for "elecomAPIService.apxc" and "Log executeAnonymous @9/26/2025, 9:29:20 PM". The main area contains the following Apex code:

```

1 public class TelecomAPIService {
2     public static void fetchCustomerData() {
3         // Step 1: Create HTTP instance
4         Http http = new Http();
5
6         // Step 2: Prepare the request
7         HttpRequest request = new HttpRequest();
8
9         // IMPORTANT: "TelecomAPI" must match your Named Credential Label/N
10        request.setEndpoint('callout:TelecomAPI/customers');
11        request.setMethod('GET');
12
13        // Step 3: Send the request
14        HttpResponse response = http.send(request);
15

```

Below the code, a log entry is shown:

Logs	Tests	Checkpoints	Query Editor	View State	Progress	Problems
ver	Application	Operation	Time	Status	Read	Size
vitri koparde	Unknown	/services/data/v64...	9/26/2025, 9:29:2...	Success		4.69 KB

By completing this phase, our **Telecom Service Hub** can:

- Securely connect to external APIs.
- Demonstrate real-time data exchange.
- Keep credentials secure with **Named Credentials**.

Phase 8: Data Management & Deployment –

Goal: Manage data and deploy project changes effectively with only the essential steps.

1. Data Import Wizard

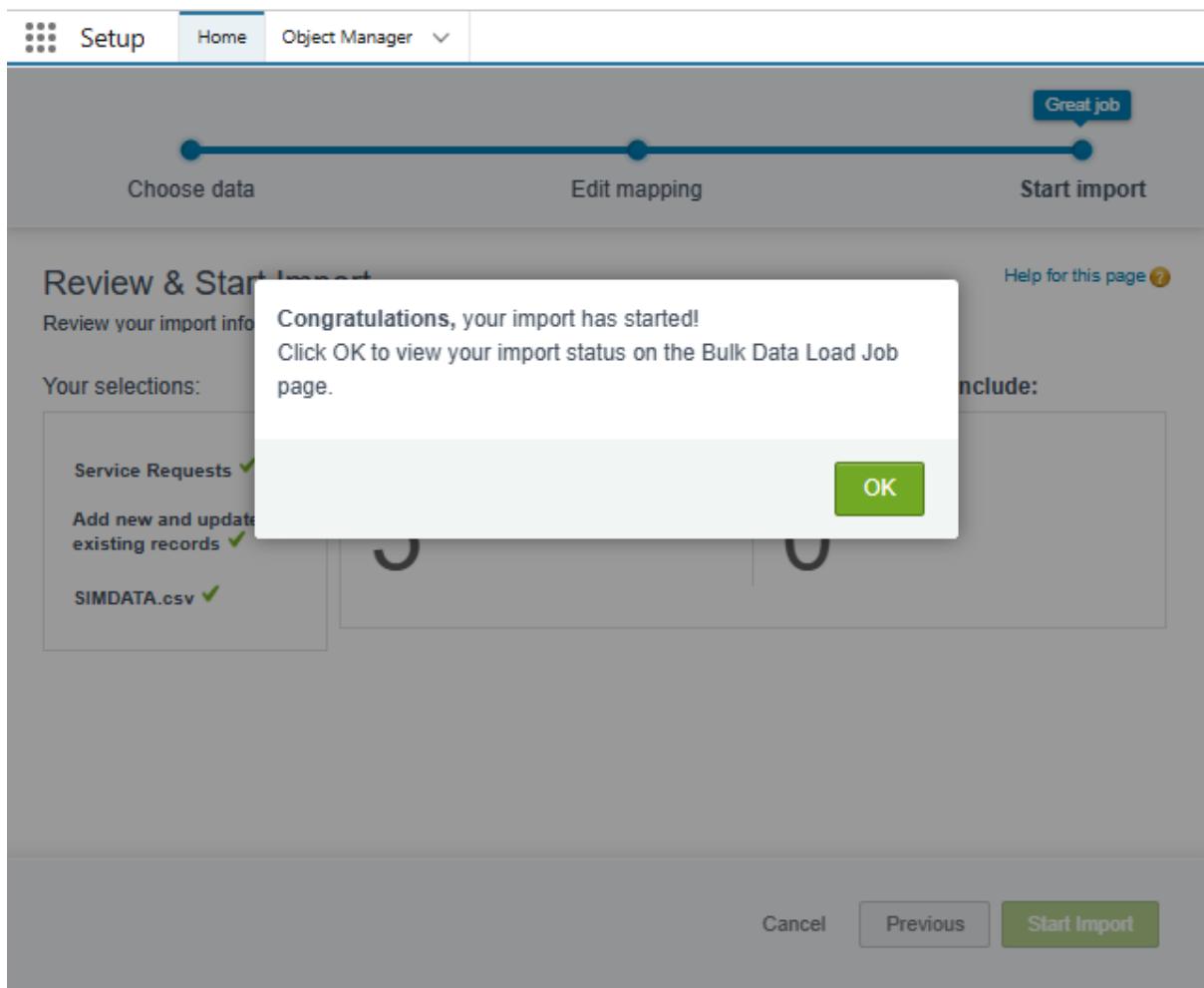
Purpose: Quickly import records into Salesforce.

Steps:

1. **Setup → Data → Data Import Wizard**
2. Choose object: **Customer__c, Service_Request__c, Plan__c**
3. Upload CSV → Map fields → Start Import

4. Review import results

Impact: Populate initial data for testing or demonstration.



2. Duplicate Rules

Purpose: Prevent duplicate records (important for Customer data).

Steps:

1. **Setup → Duplicate Management → Duplicate Rules → New Rule**
2. Select object: **Customer_c**
3. Matching Rule: **Email_c + SIM_Number_c**
4. Action: **Alert / Block duplicates → Activate**

Impact: Ensures data integrity and avoids duplicate entries.

Matching rules:

Matching Rule Detail

Object	Customer
Rule Name	Customer Email and SIM
Unique Name	Customer_Email_and_SIM
Description	(Customer: Email EXACT MatchBlank = TRUE) AND (Customer: Phone EXACT MatchBlank = FALSE) AND (Customer: Status EXACT MatchBlank = FALSE) AND (Customer: SIM_Number EXACT MatchBlank = FALSE) AND (Customer: Region EXACT MatchBlank = FALSE)
Status	Inactive
Created By	savitri koparde, 9/26/2025, 11:37 AM
Modified By	savitri koparde, 9/26/2025, 11:37 AM

Duplicate rules:

Duplicate Rule Edit

Rule Details	Required Information
Rule Name	Customer Email and SIM M
Description	Blocks new records when Customer Email and SIM Number match.
Object	Customer
Record-Level Security	<input checked="" type="radio"/> Enforce sharing rules i <input type="radio"/> Bypass sharing rules i

Actions

Specify what happens when a user tries to save a duplicate record.

Action On Create	Block ▼
Action On Edit	Allow ▼ <input checked="" type="checkbox"/> Alert <input type="checkbox"/> Report
Alert Text	A Customer with the same Email and SIM Number already exists. i

Matching Rules

Define how duplicate records are identified.

Compare Customers With	Customers ▼
Matching Rule	Customer Email and SIM Match matching rule ▼
Matching Criteria	(Customer: Email EXACT MatchBlank = TRUE) AND (Customer: SIM Number EXACT MatchBlank = FALSE)

3. Change Sets

Purpose: Deploy metadata (objects, flows, triggers) from Sandbox to Production.

Steps:

1. **Setup → Outbound Change Sets → New Change Set**
2. Add components: Custom Objects, Flows, Apex Classes
3. Upload to target org → Validate & Deploy

Impact: Moves Salesforce configurations safely between environments.

4. VS Code & Salesforce CLI (SFDX)

Purpose: Optional but helpful for deployment and version control.

Steps:

1. Install VS Code + Salesforce Extension Pack
2. Authenticate org:

```
sfdx force:auth:web:login -a DevOrg
```

3. Deploy metadata:

```
sfdx force:source:deploy -p force-app
```

Impact: Provides developer-friendly deployment workflow.

Phase 9: Reporting, Dashboards & Security Review – Must-Have Documentation

Goal: Enable Managers and Agents to track performance, monitor customer requests, and maintain data security in Salesforce.

1. Reports

Purpose: Summarize and analyze data for decision-making.

Steps (must-have reports):

1. **Service Requests by Type**

- o Object: Service_Request_c
- o Type: Summary Report
- o Group by: Request_Type_c
- o Show: Count of records, Status, Priority

The screenshot shows the Salesforce report builder interface. At the top, there's a navigation bar with 'Telecom SERVICES' logo, search bar, and various icons. Below it, the main menu includes 'Telecom Service Hub', 'Plans', 'Customers', 'Service Requests', and a dropdown for 'Reports'. The current report is titled 'New Service Requests Report' under the 'Service Requests' category.

The report configuration area has two sections: 'Fields' on the left and the report preview on the right.

Fields Section:

- Outline:** Shows grouped fields: 'Groups' (with 'GROUP ROWS'), 'Service Request: ID', 'Status', and 'Priority'.
- Filters:** Shows filters: 'Service Request: Request Number' and 'Request Type'.
- Columns:** Shows columns: 'Service Request: Request Number' and 'Request Type'.

Report Preview Section:

The preview shows a table with the following data:

Service Request: ID	Status	Priority	Service Request: Request Number (Auto Number --> Format SR-)	Request Type
a01gL00000PHOHJ (1)	In Progress (1)	Medium (1)	D02	Cancellation
Subtotal				
Subtotal				
Total (1)				

At the bottom of the preview, there are checkboxes for 'Row Counts', 'Detail Rows', 'Subtotals', and 'Grand Total'.

2. Active vs Cancelled Customers (Churn Analysis)

- o Object: Customer_c
- o Type: Summary Report
- o Group by: Status_c
- o Metrics: Count

← → ⌂ orgfarm-11cac0a6a9-dev-ed.develop.lightning.force.com/one/on... 🔍 ☆ ⌂ | ⌂ | ⌂ | ⌂ | ⌂ | ⌂

Console Home | Co... 100 Most Repeated... Get Ready for Agen... sales All Bookmarks

Telecom Services Search... REPORT New Customer Report Customer Add Chart Save & Run Save Close Run

REPORT ▾

New Customer Report Customer

Fields

Outline Filters 2

Groups GROUP ROWS

Status X

Plan: Plan Name X

Customer Status Type X

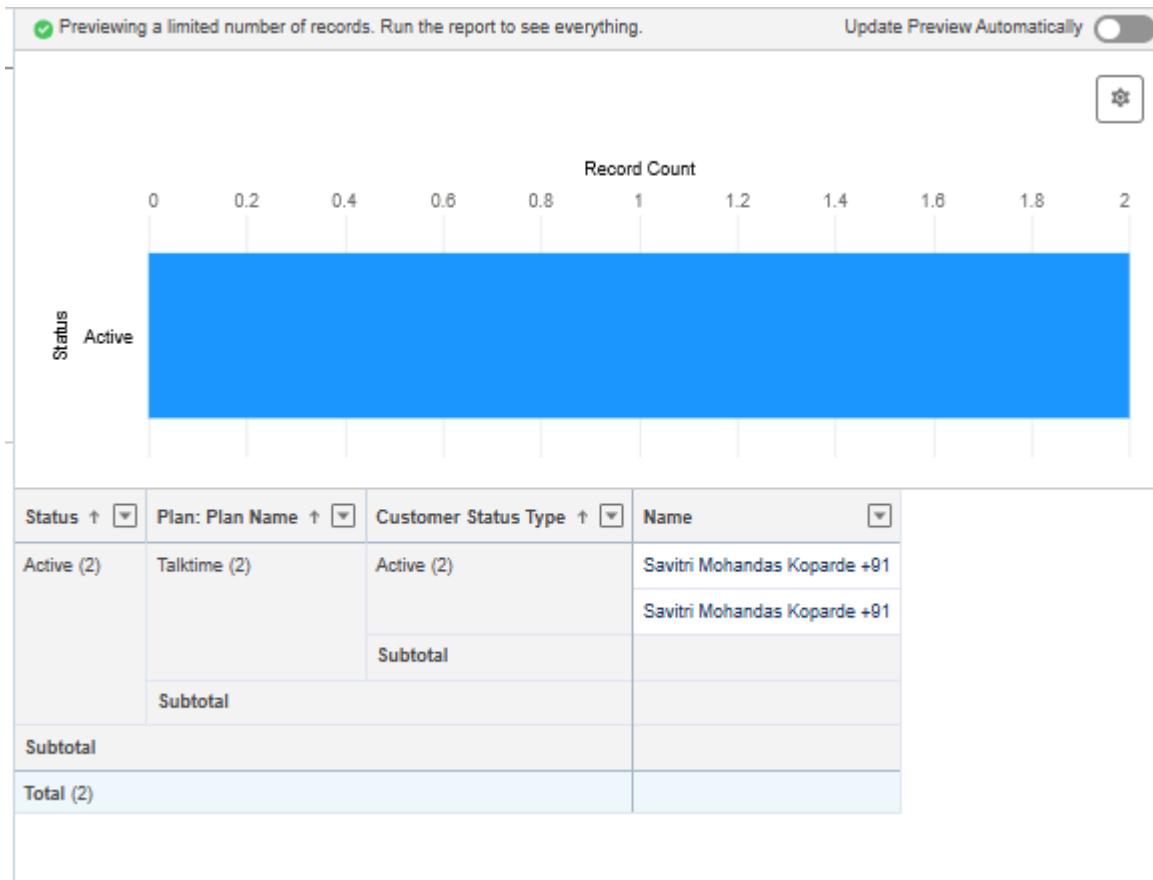
GROUP COLUMNS

Columns Add column... Name X

Previews a limited number of records. Run the report to see everything. Update Preview Automatically

Status	Plan: Plan Name	Customer Status Type	Name
Active (2)	Talktime (2)	Active (2)	Savitri Mohandas Koparde +91
			Savitri Mohandas Koparde +91
		Subtotal	
		Subtotal	
		Total (2)	

Anlysis of report using bar chart:



3. Agent Performance Report

- Object: Service_Request_c
- Type: Summary Report
- Group by: Owner (Agent)
- Show: Count of resolved requests

Impact: Real-time insights into requests, customer churn, and agent efficiency.

2. Report Types

Purpose: Define which objects and fields are available for reporting.

Steps:

1. Setup → Report Types → New Custom Report Type

2. Primary Object: Service_Request_c or Customer_c
3. Related Object: Optional (Customer_c for Service Requests)
4. Deploy → Make it available for reporting

Impact: Ensures all necessary fields and relationships are reportable.

Customer

Below is the information for this custom report type. You can click the buttons on this to preview or update information for the custom report type.

Preview Layout Edit Layout Clone
Delete Close

Details

Disp... Customer
API ... Customer
Des... details
Crea... savitri koparde,
9/27/25, 9:33 AM
Stor... other
Dep... Deployed
Mo... savitri koparde,
9/27/25, 9:33 AM

Object Relationship

Customers (A)

A
↓
B

Service Request Custom Report:

New Custom Report Type

2 Define Report Records Set

Select related objects to define which records are included in reports using this report type.

A Service Requests

Primary Object

A
↓
B

A B

↓

A B

A to B Relationship:

Each "A" record must have at least one related "B" record.
 "A" records may or may not have related "B" records.

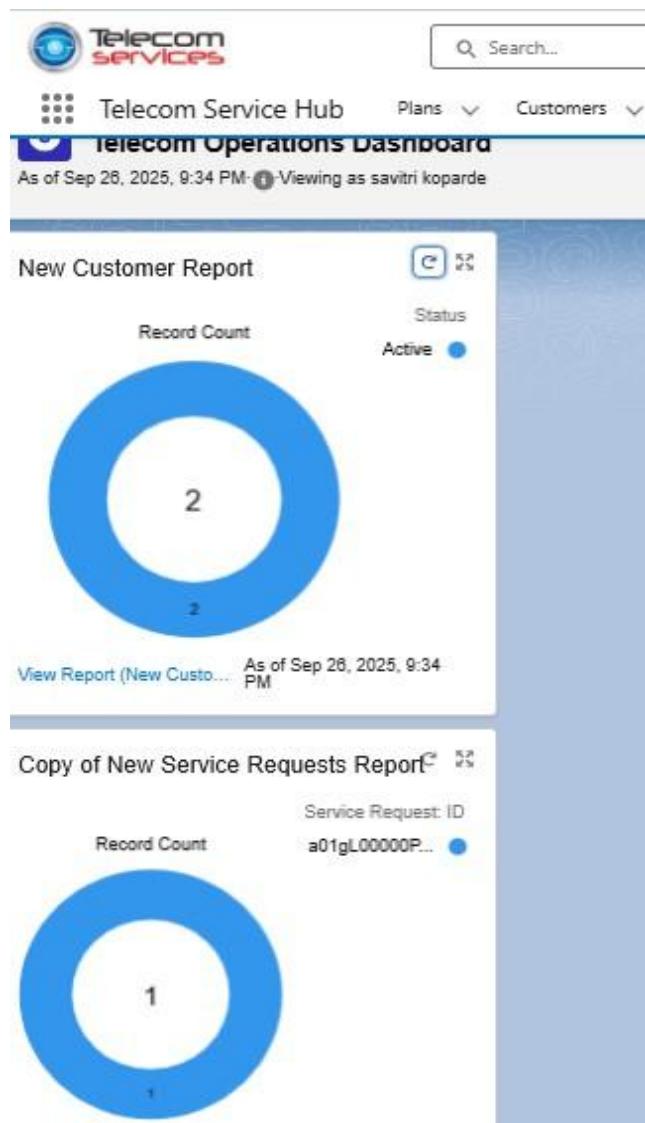
3. Dashboards

Purpose: Visualize report data for quick insights.

Steps:

1. Setup → Dashboards → New Dashboard
2. Add Components:
 - Pie chart → Service Requests by Type
 - Bar chart → Active vs Cancelled Customers
 - Table → Top Agents by Resolved Requests
3. Assign running user: Manager (so dashboard shows data according to manager access)
4. Save & Activate

Impact: Provides Managers and Agents a quick snapshot of operations.



Creating the Role Hierarchy

You can build on the existing role hierarchy shown on this page. To insert a new role, click Add Role.

Our Organization's Role Hierarchy

- Collapse All Expand All
- Telecom Service Hub Pvt. Ltd.
 - Add Role
 - CEO** Edit | Del | Assign
 - Add Role
 - Service manager** Edit | Del | Assign
 - Add Role

Show in tree view

4. Sharing Settings

Purpose: Control who sees which records in Salesforce.

Steps (must-have):

1. Setup → Sharing Settings
2. Ensure Customer_c and Plan_c → Public Read Only
3. Service_Request_c → Private
4. Role hierarchy ensures Managers see all requests, Agents see only their own

Impact: Protects sensitive customer data while enabling operational visibility. Sharing setting:

Plan	Public Read Only	Public Read Only	<input checked="" type="checkbox"/>
Service Request	Private	Private	<input checked="" type="checkbox"/>

Customer	Public Read Only	Private	<input type="checkbox"/>
	Public Read Only	Public Read Only	<input checked="" type="checkbox"/>

Role hierarchy:

5. Field Level Security (FLS)

Purpose: Restrict access to specific fields on objects.

Steps (must-have):

1. Setup → Object Manager → Object → Fields & Relationships → Set Field-Level Security
2. Example: Hide Email from Agents if required
3. Ensure Managers have full visibility

Impact: Maintains confidentiality and enforces data access policies.

System Administrator	<input type="checkbox"/>	<input type="checkbox"/>
Telcom Agent Profile	<input type="checkbox"/>	<input type="checkbox"/>
Telcom Agent Profile3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Telcom Manager Profile	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Telecom Agent	<input type="checkbox"/>	<input type="checkbox"/>
Telcom Agent Profile2	<input type="checkbox"/>	<input type="checkbox"/>
