Government Arts & Science College

Anthiyur − 638 501

Department of Computer Science

Sales Force Developer with Agent Blazer Champion

Garage Management System

Team Leader : Thiruganasambanthar M

Team Members : Uthayakumar T
Hariprakash K
Vivekananthan D

PROJECT OVERVIEW

The Garage Management System (GMS) project on Salesforce is a CRM solution aimed at transforming the way automotive garages manage customer relationships, vehicle service records, and day-to-day operations. With centralized customer profiles, service histories, and appointment scheduling, the system empowers garages to deliver more personalized and efficient customer service. Automated reminders and service notifications help enhance customer engagement, ensuring regular maintenance and timely service updates to boost customer satisfaction.

In addition to customer management, the GMS CRM includes comprehensive vehicle tracking and work order management, allowing technicians and advisors to access detailed service histories, track service progress, and assign tasks seamlessly. An integrated parts inventory module ensures that essential components are in stock, with automated reordering for low-stock items to prevent service delays. Technicians can use mobile access to view work orders and inventory on the go, optimizing workflow and reducing downtime.

Finally, the GMS CRM supports billing and invoicing, providing a streamlined system for generating service estimates, digital invoices, and online payment options. With built-in analytics and reporting, managers gain insights into customer trends, revenue, and operational efficiency, enabling data-driven decisions for business growth. The system is designed to scale with the garage, offering robust data security and compliance, making it a future-proof solution for modern garage management needs.

OBJECTIVES

- **Enhance Customer Retention and Loyalty:** By providing personalized service reminders, streamlined appointment scheduling, and post-service follow-ups, the GMS CRM aims to build long-term customer relationships and increase repeat visits.

- **Optimize Resource Allocation & Technician Efficiency:** Efficient scheduling, real-time work order tracking, and inventory management help maximize technician productivity, reduce wait times, and ensure that resources are allocated effectively for each service job.

- **Increase Revenue through Data-Driven Upselling:** With insights into customer preferences and service history, the CRM enables garages to offer tailored service recommendations and upsell additional maintenance packages, driving revenue growth.

- **Reduce Operational Costs through Automation:** By automating routine tasks like appointment reminders, parts reordering, and invoicing, the GMS CRM minimizes manual workload, reduces errors, and cuts down operational costs.

- **Improve Business Insights and Strategic Planning:** Advanced reporting and analytics provide managers with valuable insights into service trends, technician performance, and financial metrics, supporting informed decision-making and strategic business growth.

KEY FEATURES

- **Customer details:**
  - Store and manage customer details including contact information, service history, and preferences.
  - Access comprehensive service records for customer insights and tailored service.
- **Appointment:**
  - Automated reminders and notifications to reduce no-shows.
  - Integrated calendar to optimize working time.
- **Service records:**
  - Track detailed service information including service type, parts used, and associated costs.
  - Link service records to specific customer appointments for easy tracking and reference.
- **Billing details:**
  - Generate and manage billing details, linking them to service records.
  - Track payment status and send timely payment reminders.
  - Automated invoice generation and email notifications for a streamlined payment process.
- **Feedback:**
  - Collect and manage customer feedback post-service.
  - Track ratings and comments to enhance service quality.
  - Analyze feedback trends for continuous improvement.

DETAILED PROCESS

Activity – 1

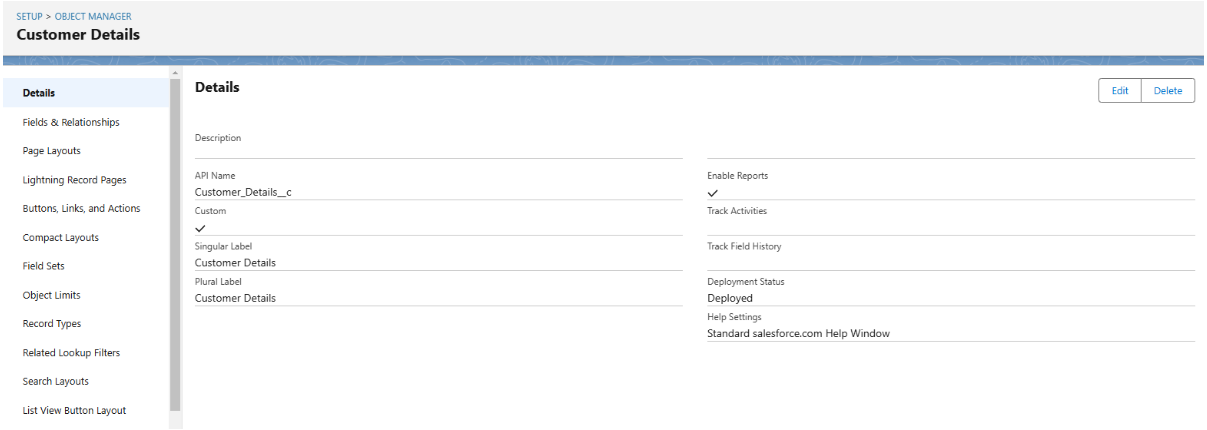
Salesforce account Creation & Activation

- Create a Developer Account by providing all the required details and Activate the account.
- Go to https://developer.salesforce.com/signup.

Activity – 2

Object Creation

- Create all the required objects with the given label name and format.
- **Customer details Object**
- Enter the label name >> Customer Details
- Plural label name >> Customer Details
- Enter Record Name Label and Format
- Record Name >> Customer Name
- Data Type >> Text
-



Customer Details

Details

Description

API Name
Customer_Details__c

Custom
✓

Singular Label
Customer Details

Plural Label
Customer Details

Enable Reports
✓

Track Activities

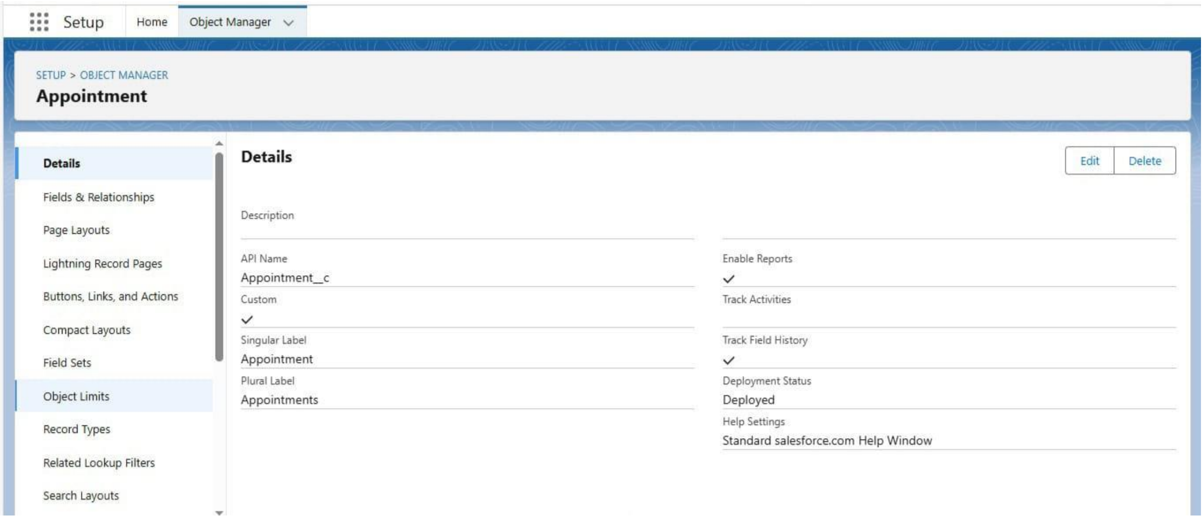
Track Field History

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

- **Appointment Object**
- Enter the label name >> Appointment
- Plural label name >> Appointments

- Enter Record Name Label and Format
- Record Name >> Appointment Name
- Data Type >> Auto Number
- Display Format >> app-{000}
- Starting number >> 1



Appointment

Details

Description

API Name
Appointment__c

Custom
✓

Singular Label
Appointment

Plural Label
Appointments

Enable Reports
✓

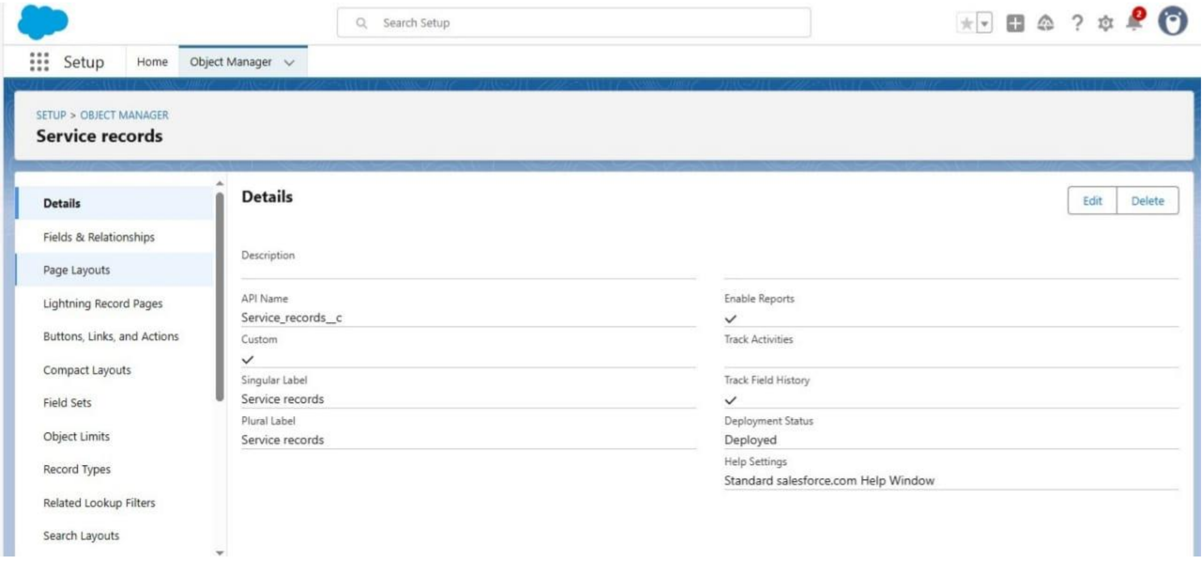
Track Activities

Track Field History
✓

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

- **Service records Object**
- Enter the label name >> Service records
- Plural label name >> Service records
- Enter Record Name Label and Format
- Record Name >>Service records Name
- Data Type >> Auto Number
- Display Format >> ser-{000}
- Starting number >> 1



Service records

Details

Description

API Name
Service_records__c

Custom
✓

Singular Label
Service records

Plural Label
Service records

Enable Reports
✓

Track Activities

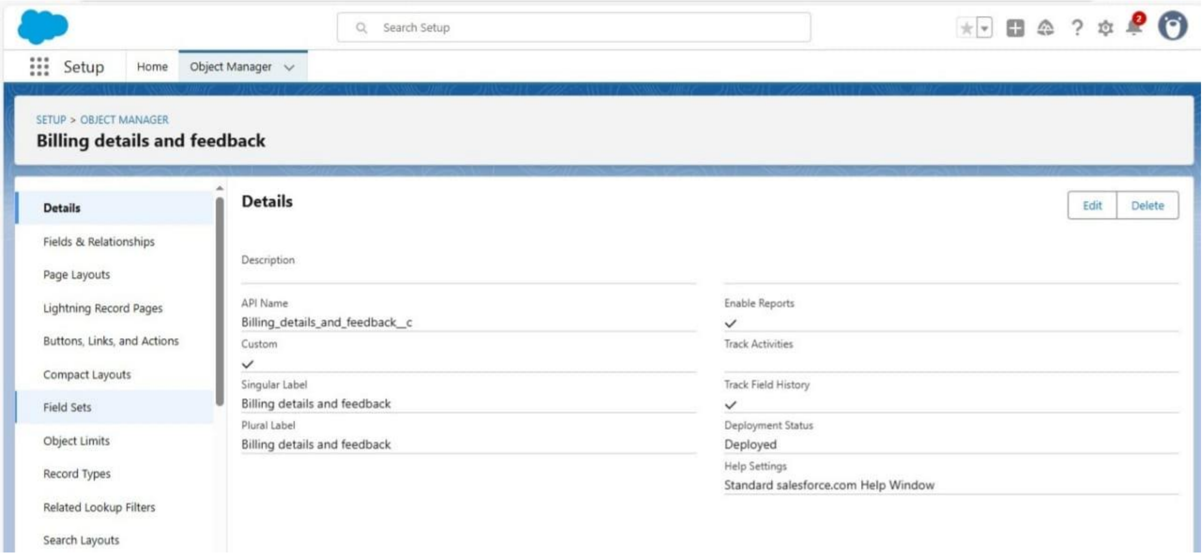
Track Field History
✓

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

- **Billing details & Feedback Object**
- Enter the label name >> Billing details and feedback

- Plural label name >> Billing details and feedback
- Enter Record Name Label and Format
- Record Name >> Billing details and feedback Name
- Data Type >> Auto Number
- Display Format >> bill-{000}
- Starting number >> 1



Billing details and feedback

Details

Description

API Name
Billing_details_and_feedback__c

Custom
✓

Singular Label
Billing details and feedback

Plural Label
Billing details and feedback

Enable Reports
✓

Track Activities

Track Field History
✓

Deployment Status
Deployed

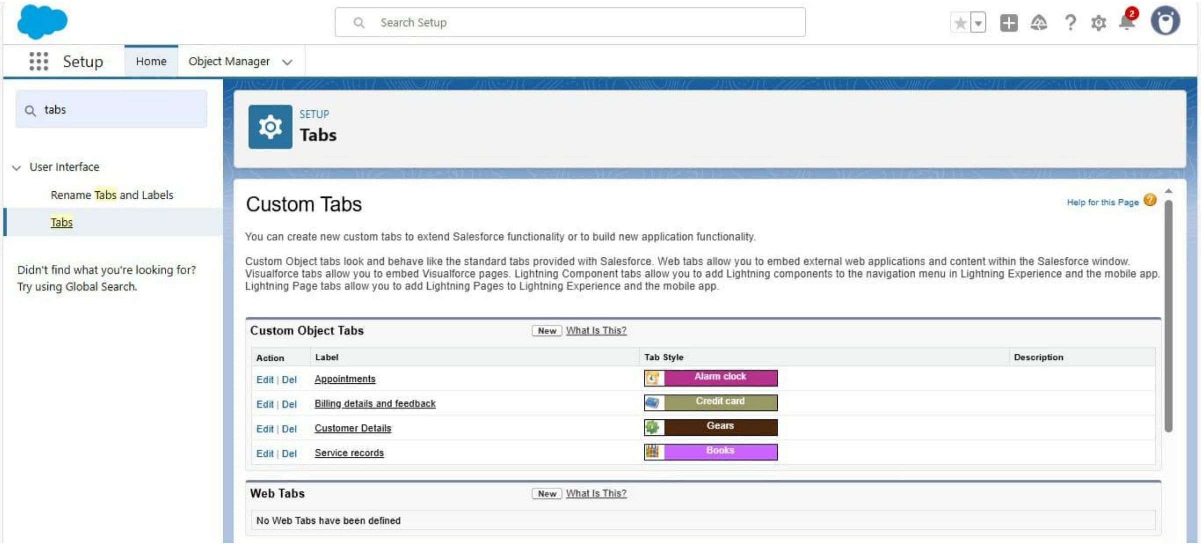
Help Settings
Standard salesforce.com Help Window

Activity – 3

Custom Tabs Creation

- A tab is like a user interface that is used to build records for objects and to view the records in the objects.
- Create these Custom tabs for every Object by following below steps:
- Select Object(Customer Details) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .

- Make sure that the Append tab to users' existing personal customizations is checked.



Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs — New — What Is This?

Action	Label	Tab Style	Description
Edit | Del	Appointments	Alarm clock	
Edit | Del	Billing details and feedback	Credit card	
Edit | Del	Customer Details	Gears	
Edit | Del	Service records	Books	

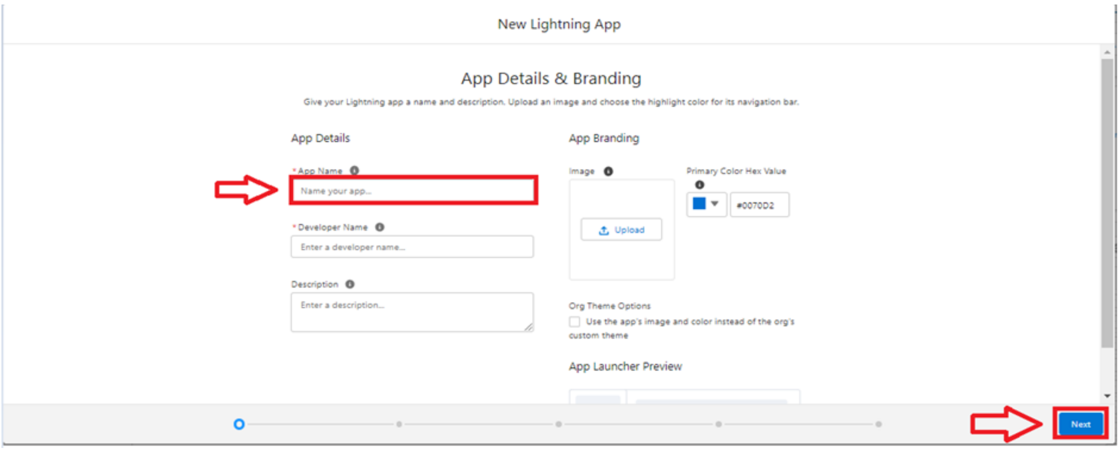
Web Tabs — New — What Is This?

No Web Tabs have been defined

Activity – 4

Create a Lighting App

1.Fill the app name in app details as Garage Management Application >> Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.



Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

* App Name: Name your app...

* Developer Name: Enter a developer name...

Description: Enter a description...

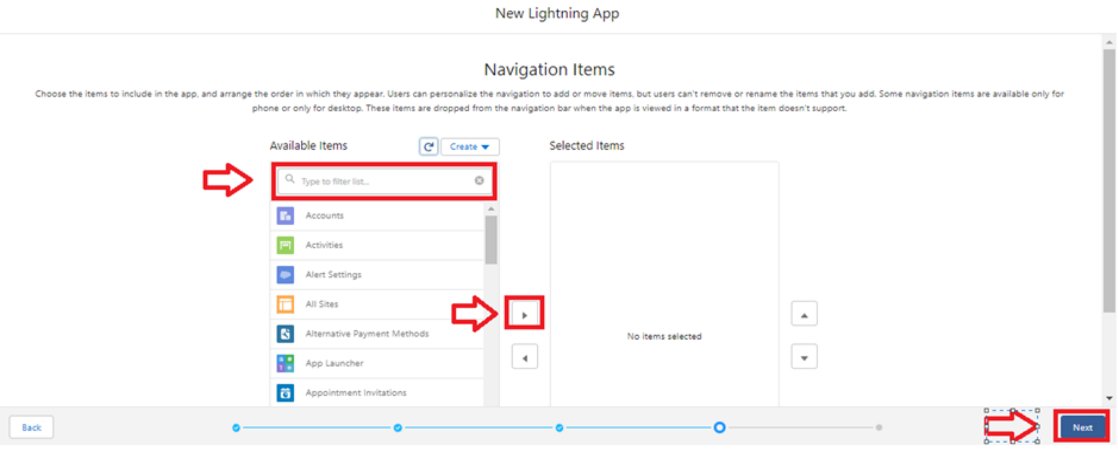
Image

Primary Color Hex Value: #0070D2

Org Theme Options

☐ Use the app's image and color instead of the org's custom theme

2. To Add Navigation Items:

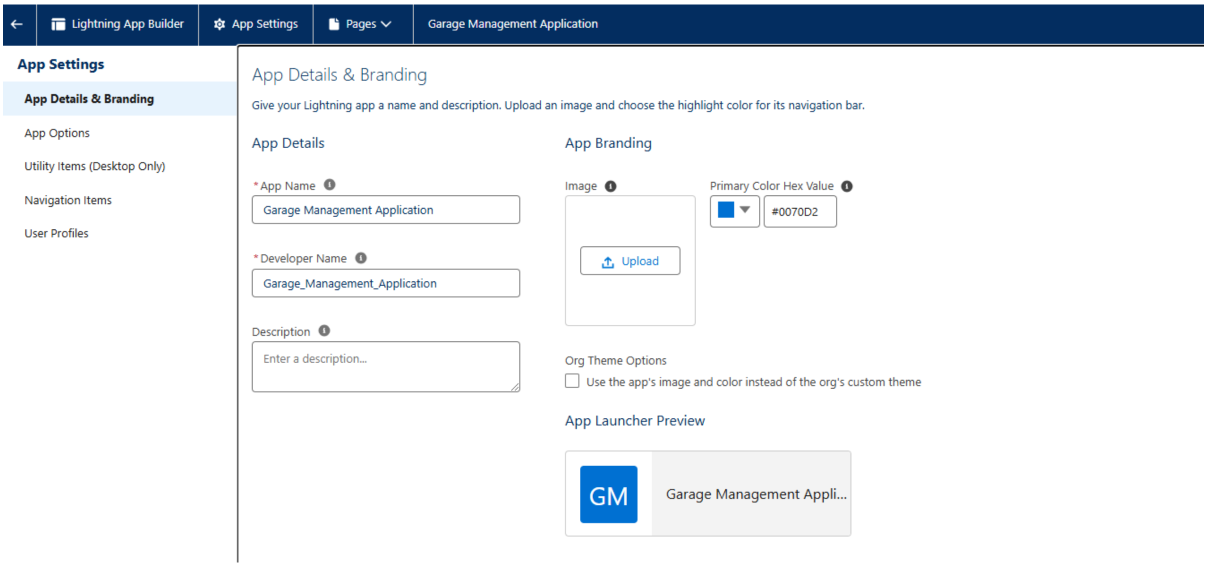


New Lightning App

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

3. Select the items (Customer Details,Appointments, Service records, Billing details and feedback, Reports and Dashboards) from the search bar and move it using the arrow button >> Next.



App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name: Garage Management Application

*Developer Name: Garage_Management_Application

Description: Enter a description...

App Branding

Primary Color Hex Value: #0070D2

Org Theme Options

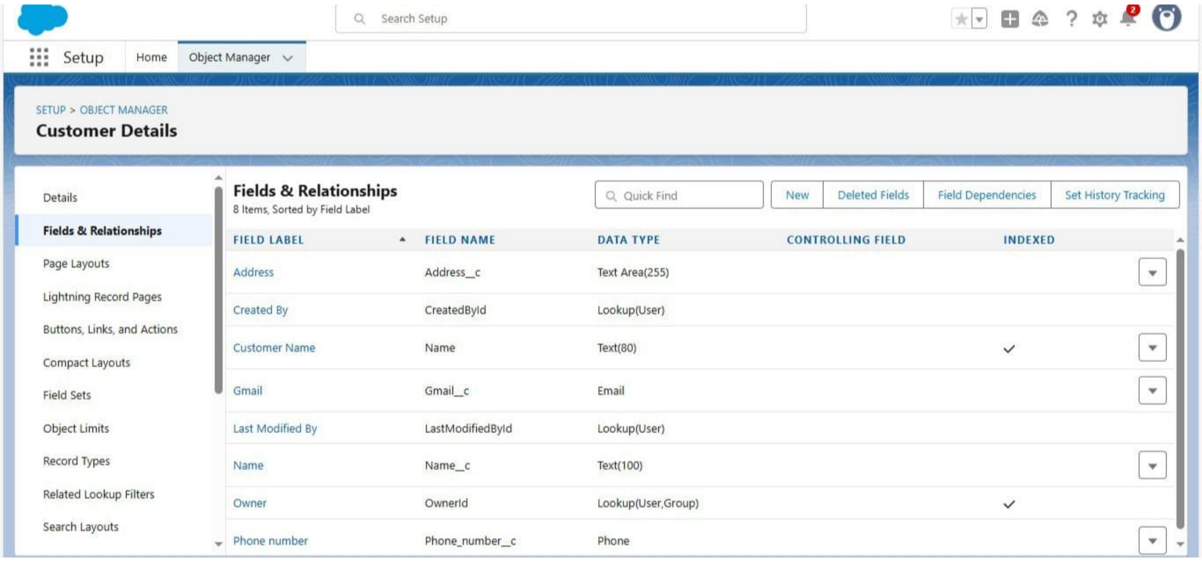
☐ Use the app's image and color instead of the org's custom theme

Activity – 5

Field Creation for every object:

Create fields for each and every object as per shown in below images.

- Customer details object



SETUP > OBJECT MANAGER

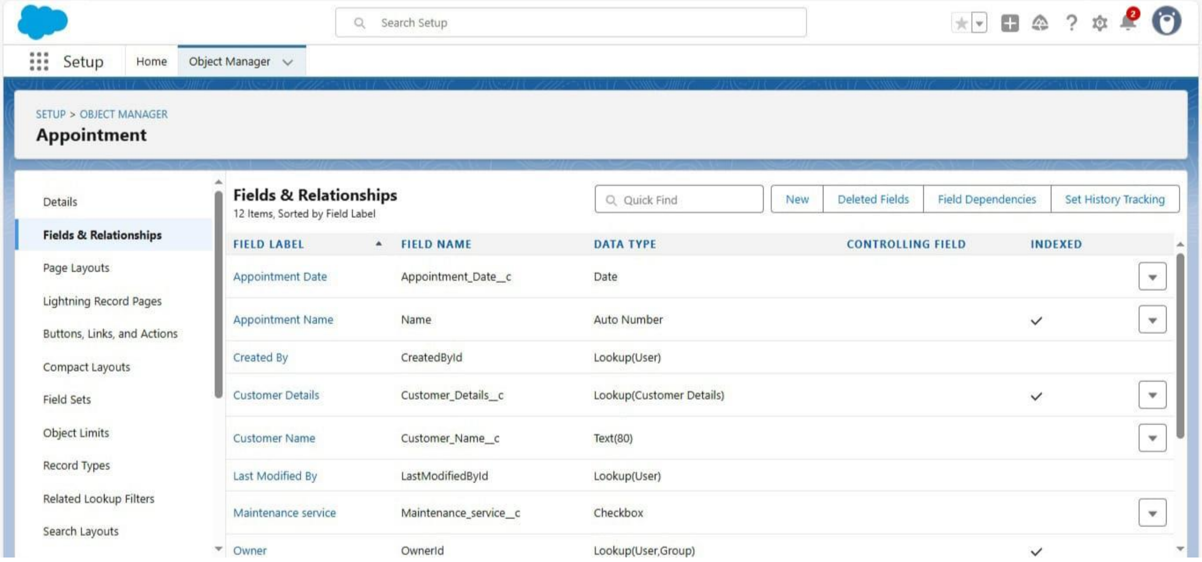
Customer Details

Fields & Relationships

8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text Area(255)		
Created By	CreatedById	Lookup(User)		
Customer Name	Name	Text(80)		✓
Gmail	Gmail__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(100)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone number	Phone_number__c	Phone		

- Appointment object



SETUP > OBJECT MANAGER

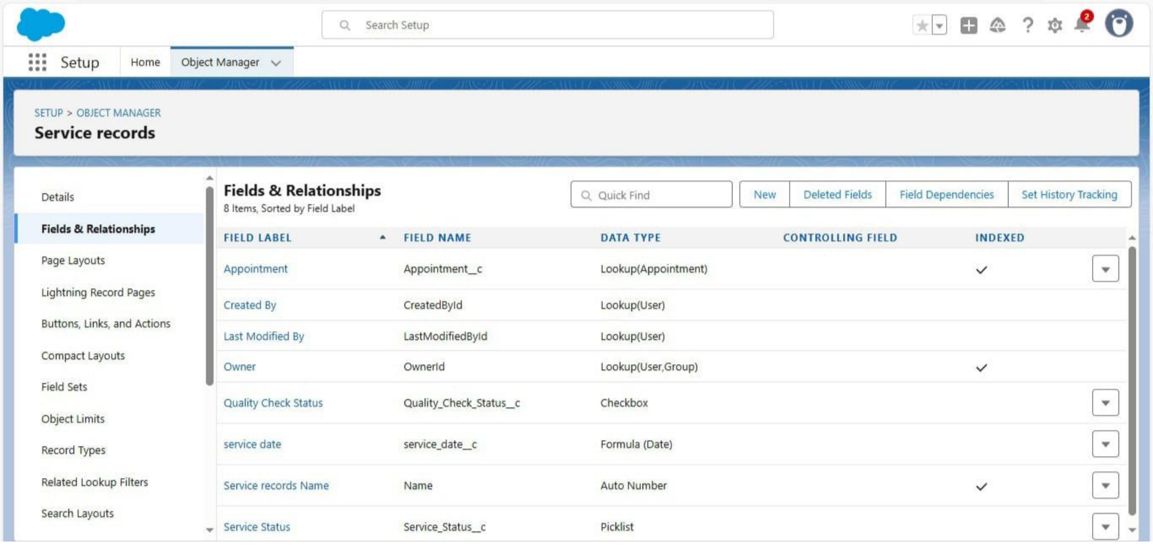
Appointment

Fields & Relationships

12 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment Date	Appointment_Date__c	Date		
Appointment Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Customer Details	Customer_Details__c	Lookup(Customer Details)		✓
Customer Name	Customer_Name__c	Text(80)		
Last Modified By	LastModifiedById	Lookup(User)		
Maintenance service	Maintenance_service__c	Checkbox		
Owner	OwnerId	Lookup(User,Group)		✓

- **Service records object**



SETUP > OBJECT MANAGER

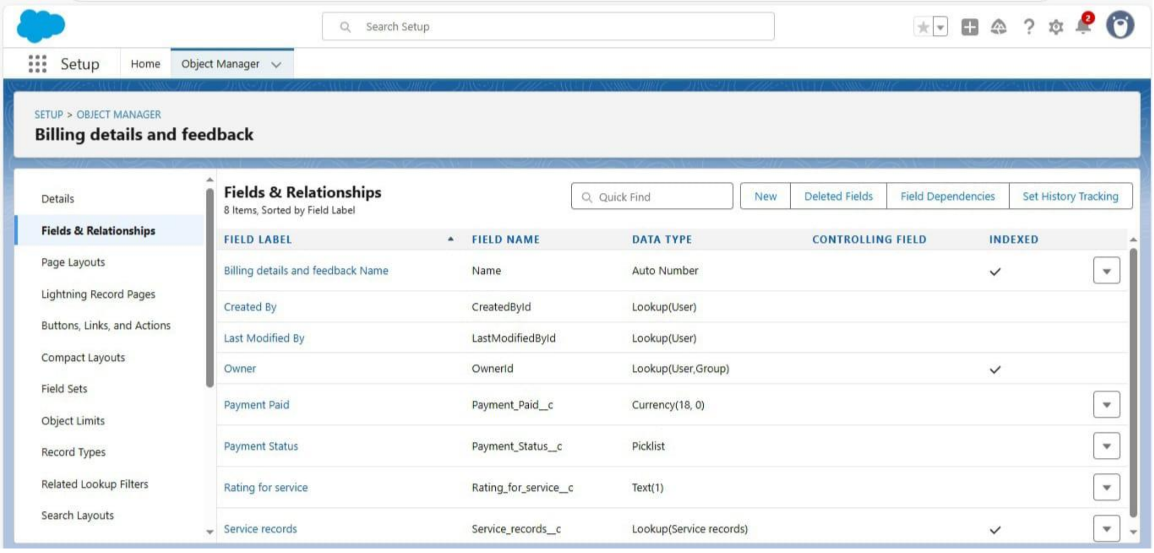
Service records

Fields & Relationships

8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment	Appointment__c	Lookup(Appointment)		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Quality Check Status	Quality_Check_Status__c	Checkbox		
service date	service_date__c	Formula (Date)		
Service records Name	Name	Auto Number		✓
Service Status	Service_Status__c	Picklist		

- **Billing details & Feedback object**



SETUP > OBJECT MANAGER

Billing details and feedback

Fields & Relationships

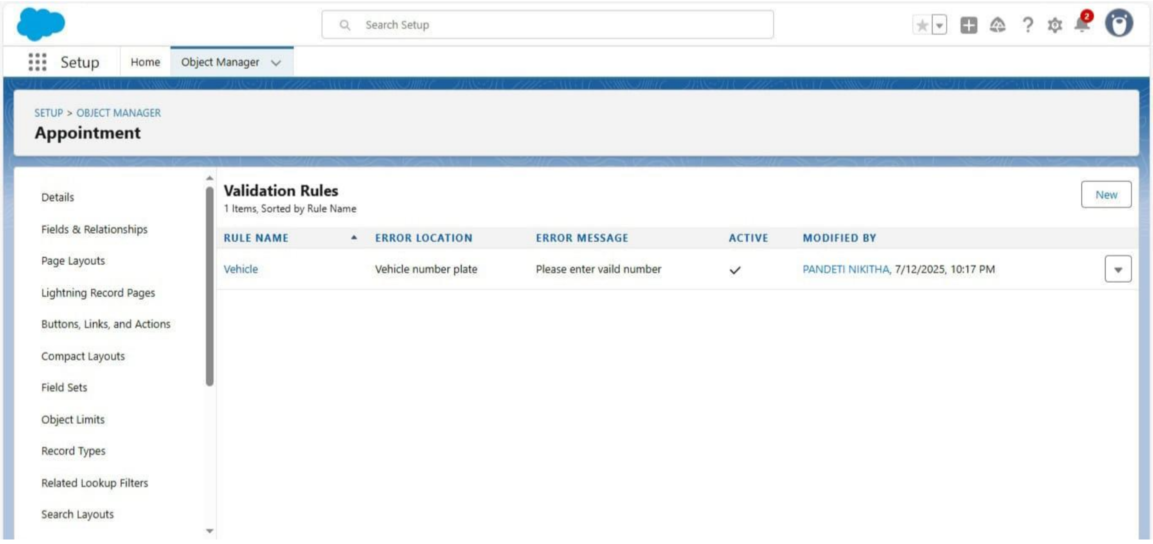
8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Billing details and feedback Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment Paid	Payment_Paid__c	Currency(18, 0)		
Payment Status	Payment_Status__c	Picklist		
Rating for service	Rating_for_service__c	Text(1)		
Service records	Service_records__c	Lookup(Service records)		✓

Activity – 6

Validation Rules

- **For Appointment**
- Enter the Rule name as " Vehicle ".
- Insert the Error Condition Formula as : -
- NOT(REGEX(Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))



SETUP > OBJECT MANAGER

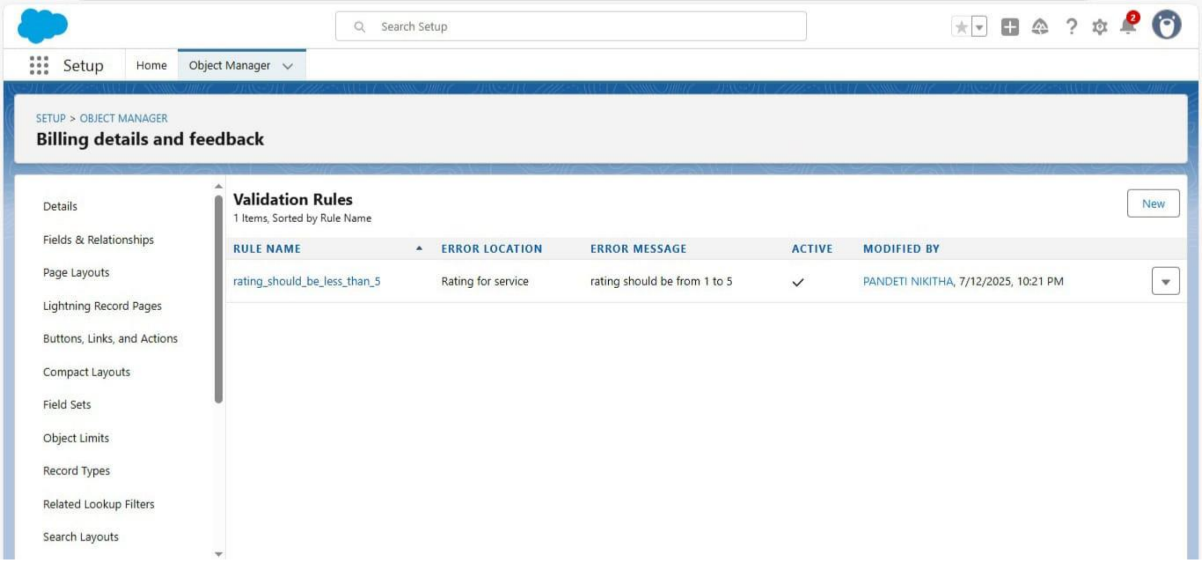
Appointment

Validation Rules

1 Items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Vehicle	Vehicle number plate	Please enter vaild number	✓	PANDETI NIKITHA, 7/12/2025, 10:17 PM

- ## For Billing details & Feedback
- Enter the Rule name as “ rating_should_be_less_than_5”.
- Insert the Error Condition Formula as : -
- NOT(REGEX(Rating_for_service__c , "[1-5]{1}"))
- Enter the Error Message as “rating should be from 1 to 5”, select the Error location as Field and select the field as “Rating for Service”.



SETUP > OBJECT MANAGER

Billing details and feedback

Validation Rules

1 Items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
rating_should_be_less_than_5	Rating for service	rating should be from 1 to 5	✓	PANDETI NIKITHA, 7/12/2025, 10:21 PM

Activity – 7

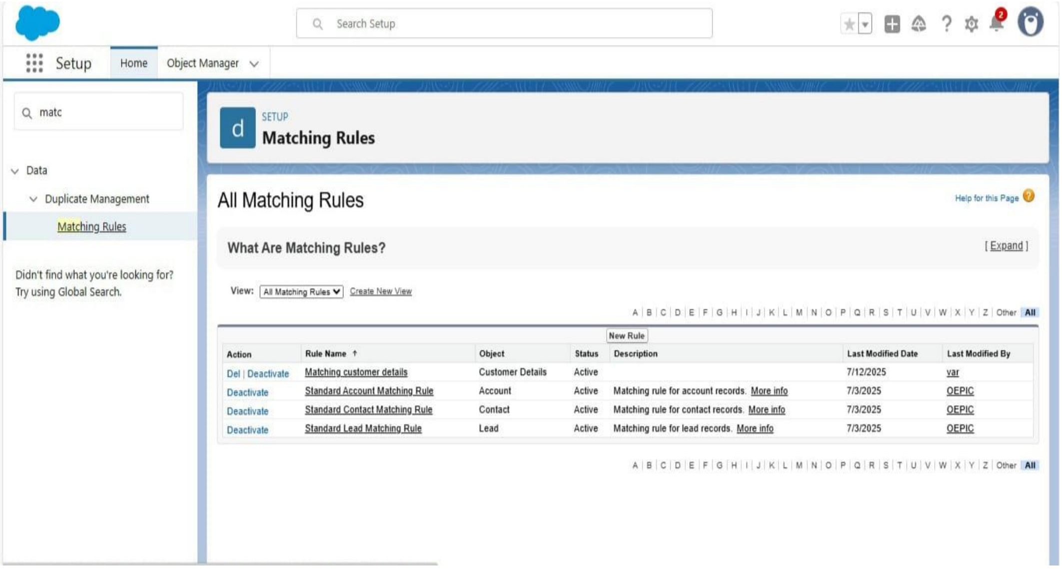
Duplication Rules

- ### Matching Rule:
- Give the Rule name : Matching customer details

- Unique name : is auto populated
- Define the matching criteria as:

Field	Matching Method
1. Gmail	Exact
2. Phone Number	Exact

- Create a matching rule for the Customer details Object with the given criteria.

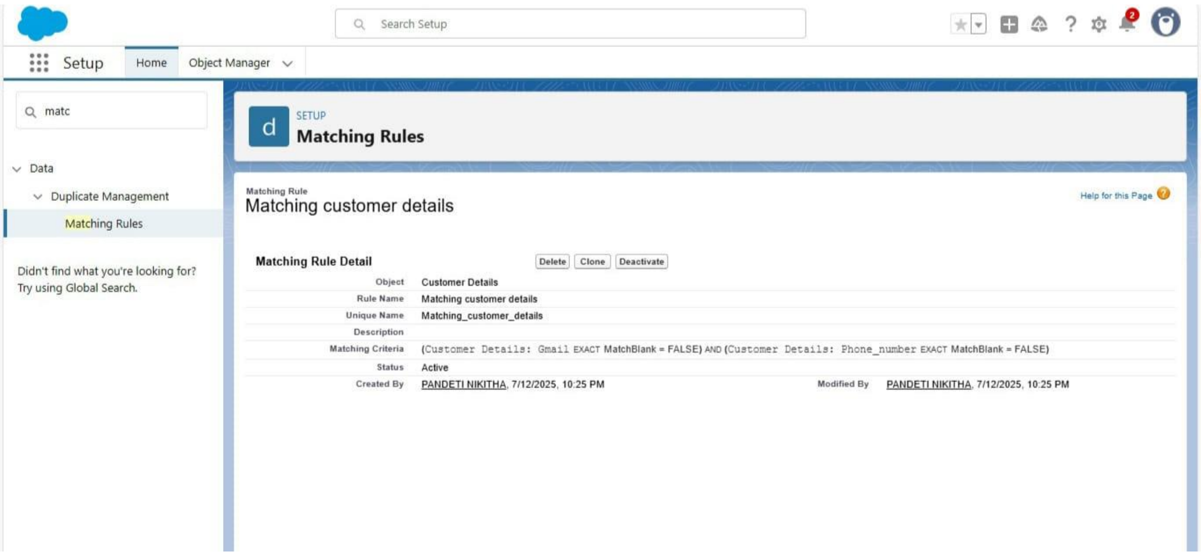


All Matching Rules

What Are Matching Rules?

View: All Matching Rules Create New View

Action	Rule Name ↑	Object	Status	Description	Last Modified Date	Last Modified By
Del | Deactivate	Matching customer details	Customer Details	Active		7/12/2025	var
Deactivate	Standard Account Matching Rule	Account	Active	Matching rule for account records. More info	7/3/2025	OEPIC
Deactivate	Standard Contact Matching Rule	Contact	Active	Matching rule for contact records. More info	7/3/2025	OEPIC
Deactivate	Standard Lead Matching Rule	Lead	Active	Matching rule for lead records. More info	7/3/2025	OEPIC



Matching Rule
Matching customer details

Matching Rule Detail

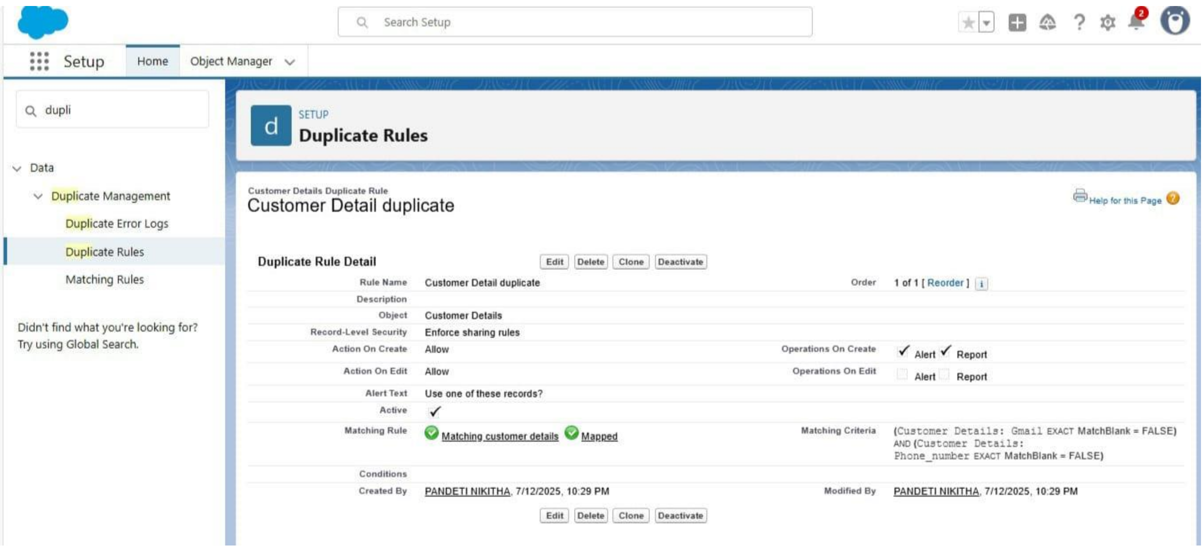
Object: Customer Details
Rule Name: Matching customer details
Unique Name: Matching_customer_details
Description:
Matching Criteria: (Customer Details: Gmail EXACT MatchBlank = FALSE) AND (Customer Details: Phone_number EXACT MatchBlank = FALSE)
Status: Active
Created By: PANDETI NIKITHA, 7/12/2025, 10:25 PM
Modified By: PANDETI NIKITHA, 7/12/2025, 10:25 PM

-Duplicate Rules:

Create a duplicate rule for the same customer details object which allows the duplication in input fields.

Steps:

- Give the Rule name as : Customer Detail duplicate.
- Select the matching rule : Matching customer details.

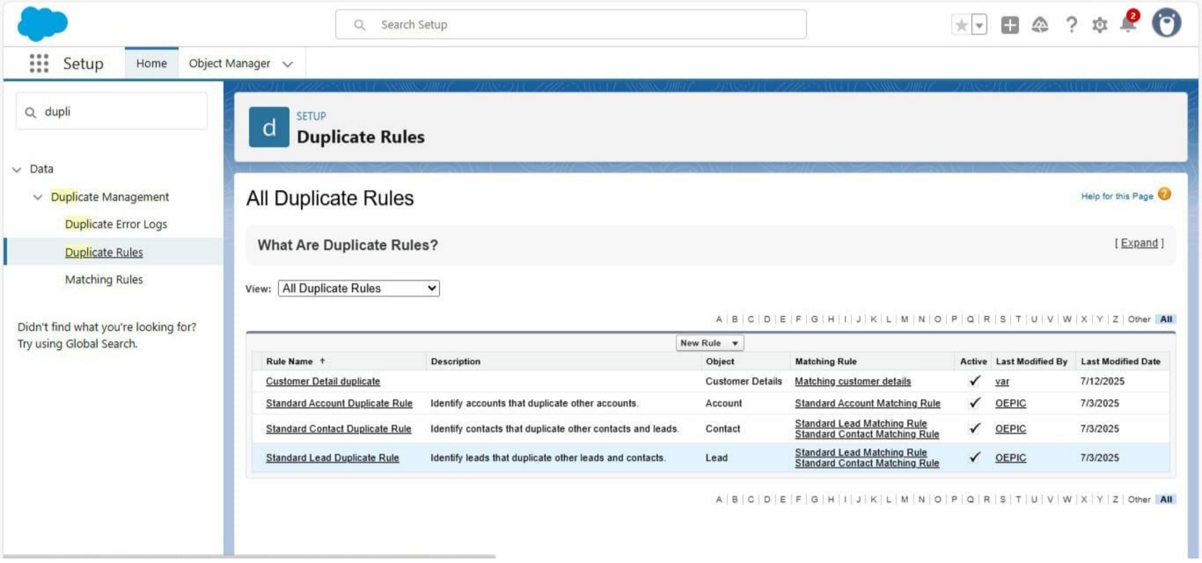


Customer Details Duplicate Rule

Customer Detail duplicate

Duplicate Rule Detail

Rule Name	Customer Detail duplicate	Order	1 of 1 [Reorder]
Description			
Object	Customer Details		
Record-Level Security	Enforce sharing rules		
Action On Create	Allow	Operations On Create	✓ Alert ✓ Report
Action On Edit	Allow	Operations On Edit	☐ Alert ☐ Report
Alert Text	Use one of these records?		
Active	✓		
Matching Rule	Matching customer details Mapped	Matching Criteria	(Customer Details: Gmail EXACT MatchBlank = FALSE) AND (Customer Details: Phone_number EXACT MatchBlank = FALSE)
Conditions			
Created By	PANDETI NIKITHA, 7/12/2025, 10:29 PM	Modified By	PANDETI NIKITHA, 7/12/2025, 10:29 PM



All Duplicate Rules

What Are Duplicate Rules?

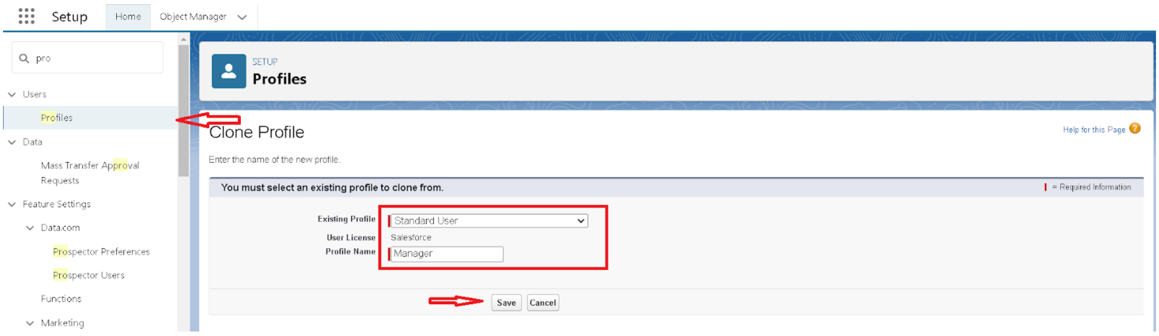
View: All Duplicate Rules

Rule Name ↑	Description	Object	Matching Rule	Active	Last Modified By	Last Modified Date
Customer Detail duplicate		Customer Details	Matching customer details	✓	var	7/12/2025
Standard Account Duplicate Rule	Identify accounts that duplicate other accounts.	Account	Standard Account Matching Rule	✓	OEPIC	7/3/2025
Standard Contact Duplicate Rule	Identify contacts that duplicate other contacts and leads.	Contact	Standard Lead Matching Rule Standard Contact Matching Rule	✓	OEPIC	7/3/2025
Standard Lead Duplicate Rule	Identify leads that duplicate other leads and contacts.	Lead	Standard Lead Matching Rule Standard Contact Matching Rule	✓	OEPIC	7/3/2025

Activity – 8

Creating Profiles

- ### Manager Profile



Clone Profile

Enter the name of the new profile.

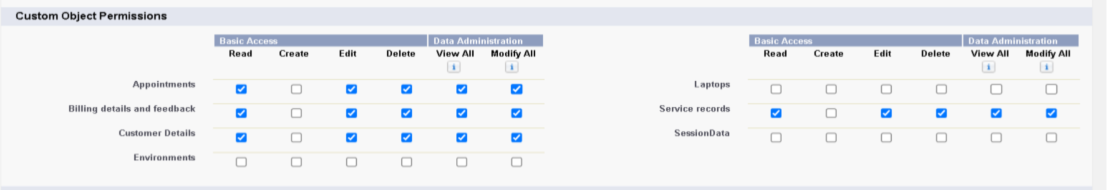
You must select an existing profile to clone from.

Existing Profile: Standard User
User License: Salesforce
Profile Name: Manager

- Select the Custom App settings as default for the Garage management.

- 

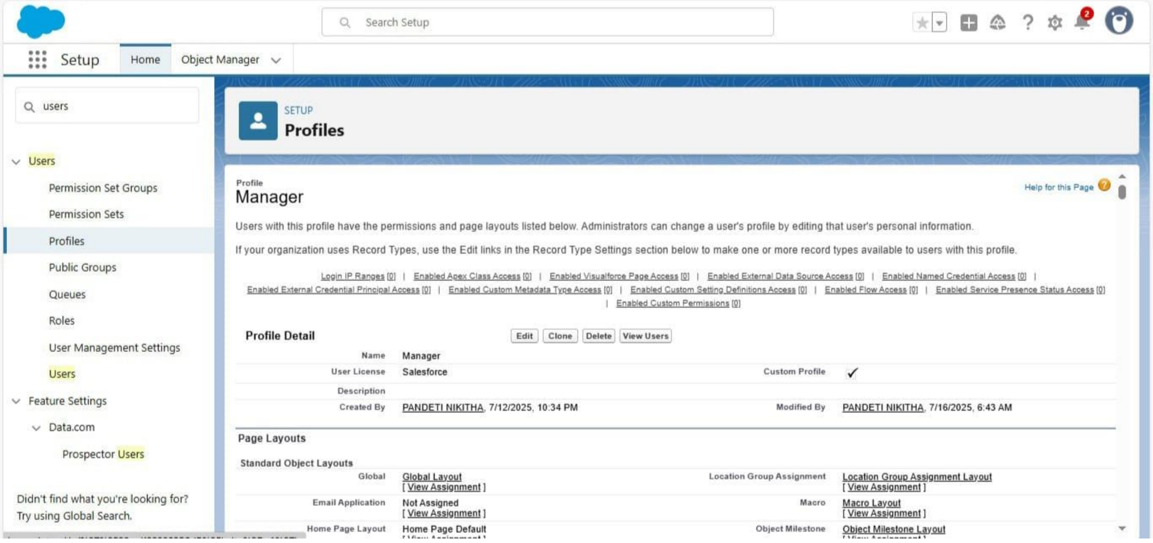
	☑	○	(standard__ServiceConsole)	☐	○
Data Manager (standard__DataManager)	☑	○	Service (standard__Service)	☑	○
Digital Experiences (standard__SalesforceCMS)	☑	○	Service Console (standard__LightningService)	☑	○
Garage Management Application (Garage_Management_Application)	☑	●	Site.com (standard__Sites)	☑	○
Laptop Hub (Laptop_Hub)	☐	○	Subscription Management (standard__RevenueCloudConsole)	☑	○

- Scroll down to Custom Object Permissions and Give access permissions for Appointments,Billing details and feedback , service records and customer details objects as mentioned in the below diagram.
- 

Custom Object Permissions

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Appointments	☑	☐	☑	☑	☑	☑
Billing details and feedback	☑	☐	☑	☑	☑	☑
Customer Details	☑	☐	☑	☑	☑	☑
Environments	☐	☐	☐	☐	☐	☐

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Laptops	☐	☐	☐	☐	☐	☐
Service records	☑	☐	☑	☑	☑	☑
SessionData	☐	☐	☐	☐	☐	☐

- Changing the session times out after should be “ 8 hours of inactivity”.
- Change the password policies as mentioned :
- User passwords expire in should be “ never expires ”.
- Minimum password length should be “ 8 ”.
- 

Profile
Manager

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Profile Detail

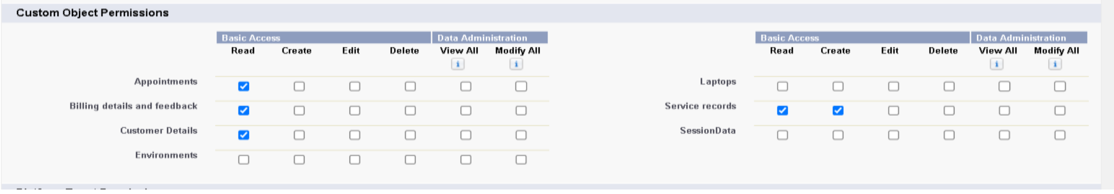
Name	Manager		
User License	Salesforce	Custom Profile	✓
Description			
Created By	PANDETI NIKITHA, 7/12/2025, 10:34 PM	Modified By	PANDETI NIKITHA, 7/16/2025, 6:43 AM

Page Layouts

Standard Object Layouts

Global	Global Layout [View Assignment]	Location Group Assignment	Location Group Assignment Layout [View Assignment]
Email Application	Not Assigned [View Assignment]	Macro	Macro Layout [View Assignment]
Home Page Layout	Home Page Default	Object Milestone	Object Milestone Layout

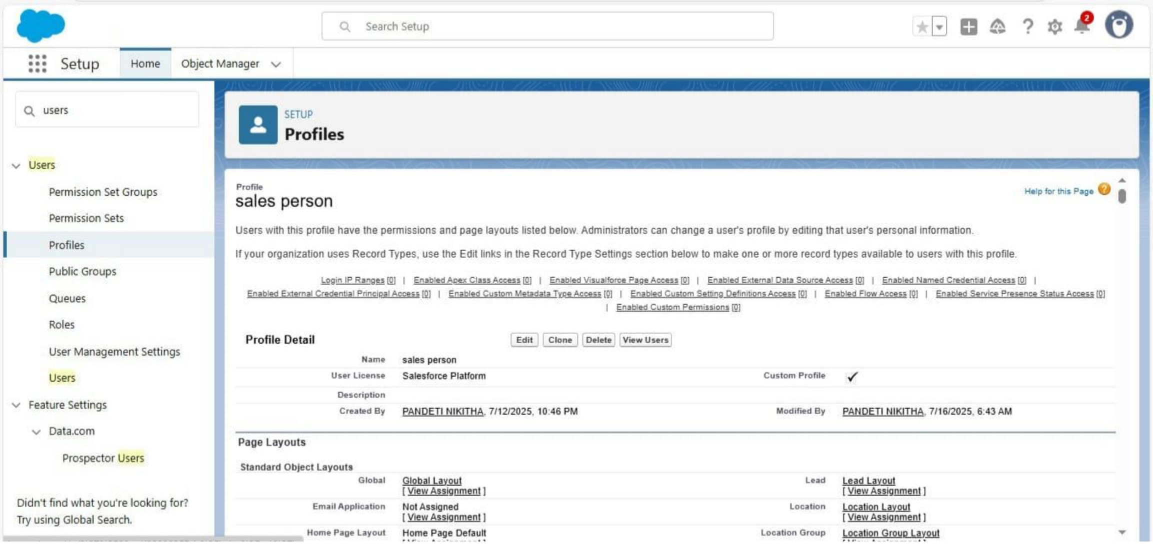
Sales Person Profile

- Select the Custom App settings as default for the GArage management.
- Scroll down to Custom Object Permissions and Give access permissions for Appointments,Billing details and feedback , service records and customer details objects as mentioned in the below diagram.
- 

Custom Object Permissions

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Appointments	☑	☐	☐	☐	☐	☐
Billing details and feedback	☑	☐	☐	☐	☐	☐
Customer Details	☑	☐	☐	☐	☐	☐
Environments	☐	☐	☐	☐	☐	☐

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Laptops	☐	☐	☐	☐	☐	☐
Service records	☑	☑	☐	☐	☐	☐
SessionData	☐	☐	☐	☐	☐	☐



SETUP

Profiles

Profile

sales person

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Login IP Ranges [0] | Enabled Apex Class Access [0] | Enabled Visualforce Page Access [0] | Enabled External Data Source Access [0] | Enabled Named Credential Access [0] | Enabled External Credential Principal Access [0] | Enabled Custom Metadata Type Access [0] | Enabled Custom Setting Definitions Access [0] | Enabled Flow Access [0] | Enabled Service Presence Status Access [0] | Enabled Custom Permissions [0]

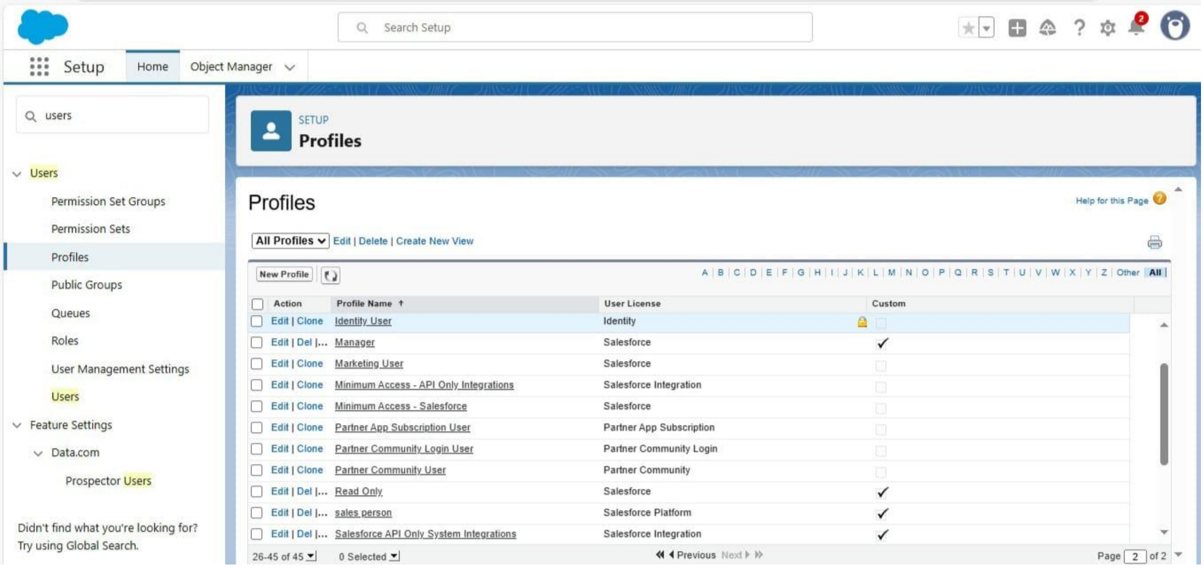
Profile Detail

Name	sales person		
User License	Salesforce Platform	Custom Profile	✓
Description			
Created By	PANDETI NIKITHA, 7/12/2025, 10:46 PM	Modified By	PANDETI NIKITHA, 7/16/2025, 6:43 AM

Page Layouts

Standard Object Layouts

Global	Global Layout [View Assignment]	Lead	Lead Layout [View Assignment]
Email Application	Not Assigned [View Assignment]	Location	Location Layout [View Assignment]
Home Page Layout	Home Page Default	Location Group	Location Group Layout



SETUP

Profiles

Profiles

All Profiles Edit | Delete | Create New View

☐ Action	Profile Name ↑	User License	Custom
☐ Edit | Clone	Identity User	Identity	☐
☐ Edit | Del |...	Manager	Salesforce	✓
☐ Edit | Clone	Marketing User	Salesforce	☐
☐ Edit | Clone	Minimum Access - API Only Integrations	Salesforce Integration	☐
☐ Edit | Clone	Minimum Access - Salesforce	Salesforce	☐
☐ Edit | Clone	Partner App Subscription User	Partner App Subscription	☐
☐ Edit | Clone	Partner Community Login User	Partner Community Login	☐
☐ Edit | Clone	Partner Community User	Partner Community	☐
☐ Edit | Del |...	Read Only	Salesforce	✓
☐ Edit | Del |...	sales person	Salesforce Platform	✓
☐ Edit | Del |...	Salesforce API Only System Integrations	Salesforce Integration	✓

26-45 of 45 0 Selected ◀◀ ◀ Previous Next ▶ ▶▶ Page 2 of 2

Activity – 9

Roles & Role Hierarchy

Manager Role:

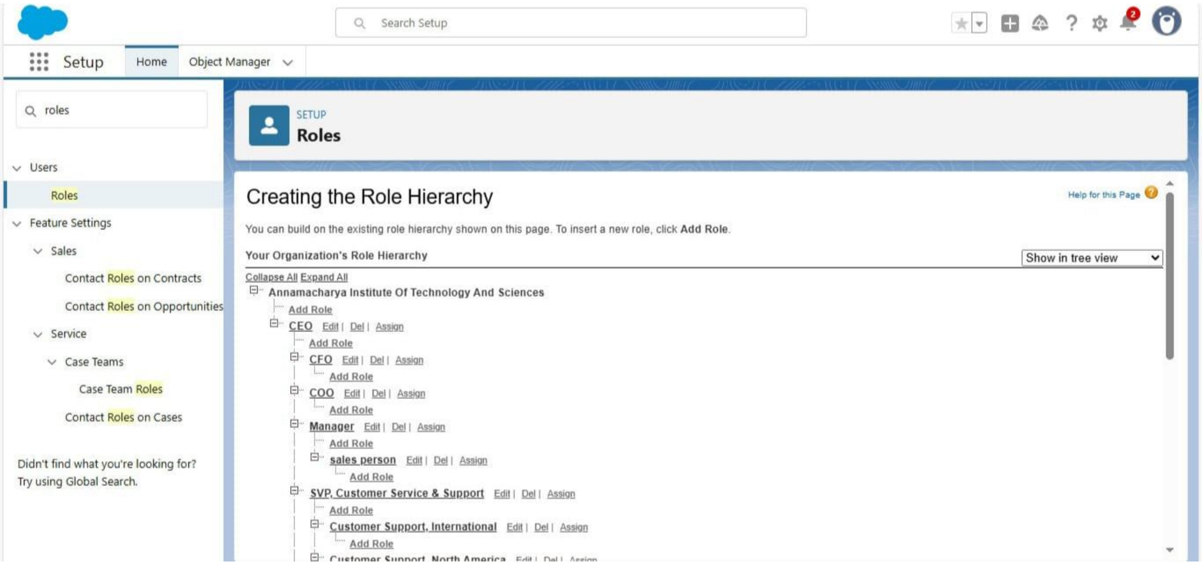
-Click on Expand All and click on add role under whom this role works.

-Give Label as “Manager” and Role name gets auto populated.

Another Role:

-Click plus on CEO role, and click add role under manager.

-Give Label as “sales person” and Role name gets auto populated.



Creating the Role Hierarchy

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

Your Organization's Role Hierarchy

- Annamacharya Institute Of Technology And Sciences
  - Add Role
  - CEO Edit | Del | Assign
    - Add Role
    - CFO Edit | Del | Assign
      - Add Role
    - COO Edit | Del | Assign
      - Add Role
    - Manager Edit | Del | Assign
      - Add Role
      - sales person Edit | Del | Assign
        - Add Role
    - SVP, Customer Service & Support Edit | Del | Assign
      - Add Role
      - Customer Support, International Edit | Del | Assign
        - Add Role
      - Customer Support, North America Edit | Del | Assign

Activity – 10

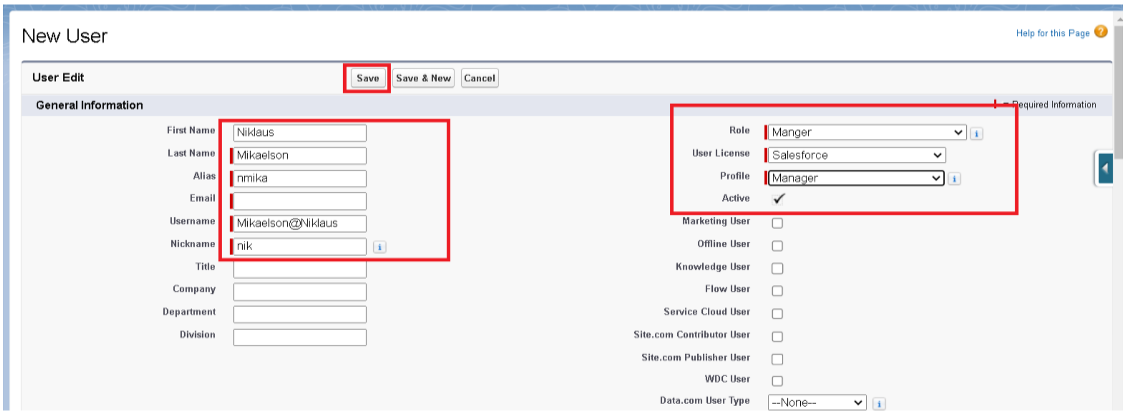
Users

anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records.

Create Users:

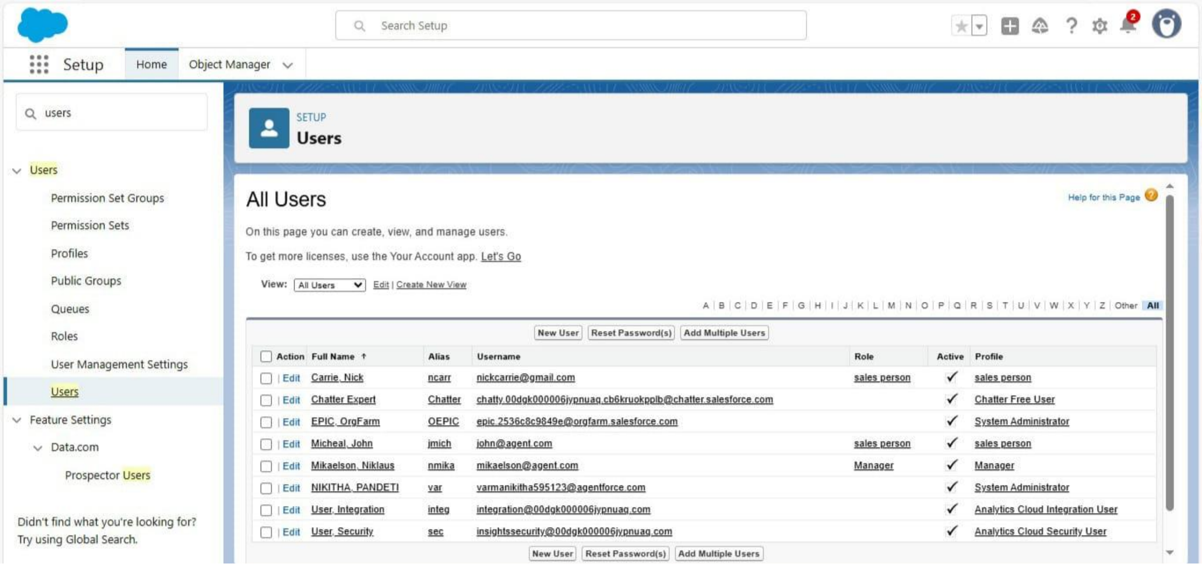
Fill in the fields :

1. First Name : Niklaus
2. Last Name : Mikaelson
3. Alias : Give a Alias Name
4. Email id : Give your Personal Email id
5. Username : Username should be in this form: text@text.text
6. Nick Name : Give a Nickname
7. Role : Manager
8. User licence : Salesforce
9. Profiles : Manager



First Name	Niklaus
Last Name	Mikaelson
Alias	nmika
Email	
Username	Mikaelson@Niklaus
Nickname	nik
Title	
Company	
Department	
Division	
Role	Manger
User License	Salesforce
Profile	Manager
Active	✓
Marketing User	☐
Offline User	☐
Knowledge User	☐
Flow User	☐
Service Cloud User	☐
Site.com Contributor User	☐
Site.com Publisher User	☐
WDC User	☐
Data.com User Type	--None--

Create Another Users:



All Users

On this page you can create, view, and manage users.

To get more licenses, use the Your Account app. Let's Go

☐ Action	Full Name ↑	Alias	Username	Role	Active	Profile
☐ | Edit	Carrie, Nick	ncarr	nickcarrie@gmail.com	sales person	✓	sales person
☐ | Edit	Chatter Expert	Chatter	chatty.00dgk000006jypnuaq.cb6kruokpplb@chatter.salesforce.com		✓	Chatter Free User
☐ | Edit	EPIC, OrgFarm	OEPIC	epic.2536c8c9849e@orgfarm.salesforce.com		✓	System Administrator
☐ | Edit	Micheal, John	jmich	john@agent.com	sales person	✓	sales person
☐ | Edit	Mikaelson, Niklaus	nmika	mikaelson@agent.com	Manager	✓	Manager
☐ | Edit	NIKITHA, PANDETI	var	varmanikitha595123@agentforce.com		✓	System Administrator
☐ | Edit	User, Integration	integ	integration@00dgk000006jypnuaq.com		✓	Analytics Cloud Integration User
☐ | Edit	User, Security	sec	insightssecurity@00dgk000006jypnuaq.com		✓	Analytics Cloud Security User

Activity – 11

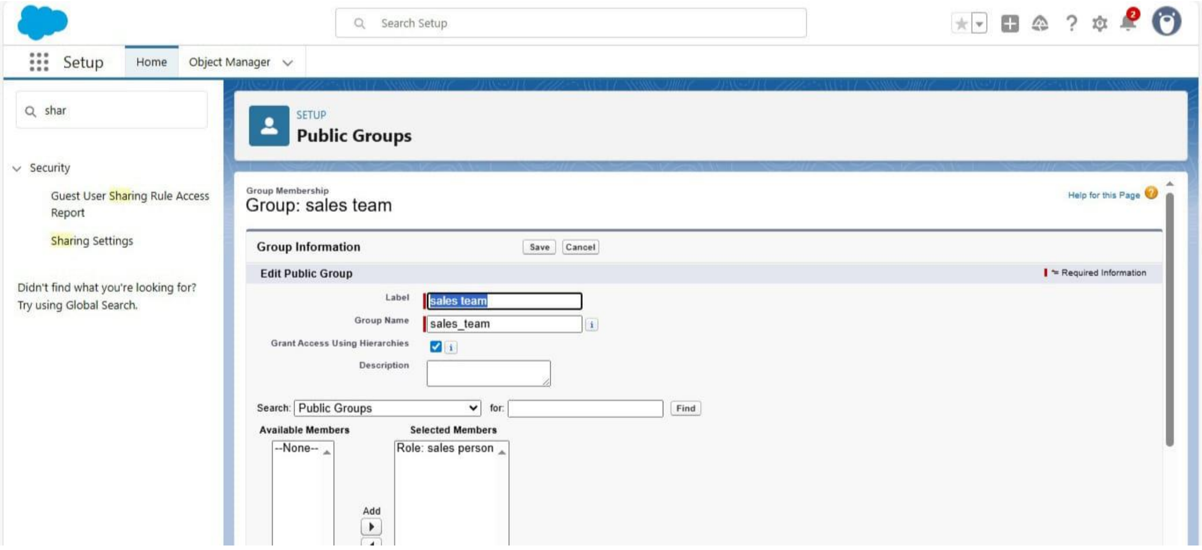
Public Groups

- **Creating New Public Groups**

  a valuable tool for Salesforce administrators and developers to streamline user management, data access, and security settings.

Steps:

1. Give the Label as "sales team".
2. Group name is autopopulated.
3. Search for Roles.
4. In Available Members select Sales person and click on add it will be moved to selected member.



Group Membership

Group: sales team

Group Information

Edit Public Group

Label: sales team

Group Name: sales_team

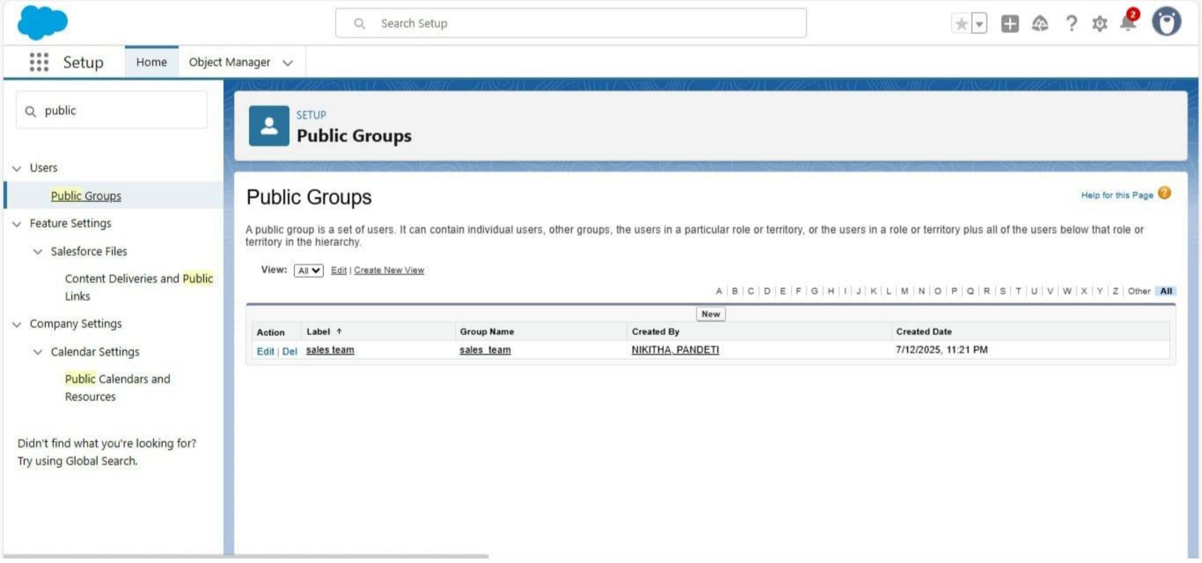
Grant Access Using Hierarchies: ☑

Description:

Search: Public Groups for:

Available Members: --None--

Selected Members: Role: sales person



Public Groups

A public group is a set of users. It can contain individual users, other groups, the users in a particular role or territory, or the users in a role or territory plus all of the users below that role or territory in the hierarchy.

View: All

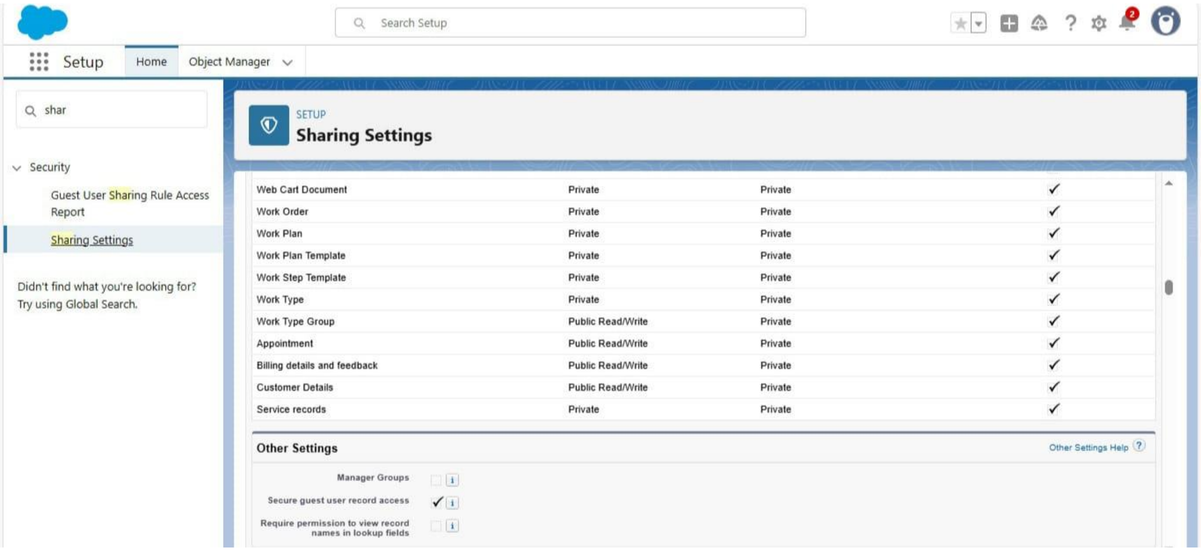
Action	Label ↑	Group Name	Created By	Created Date
Edit | Del	sales team	sales_team	NIKITHA, PANDETI	7/12/2025, 11:21 PM

Activity – 12

Sharing Settings: control how records are accessed and shared within your organization. These settings are crucial for maintaining data security and privacy.

Steps:

-Change the OWD setting of the Service records Object to private as shown in fig.



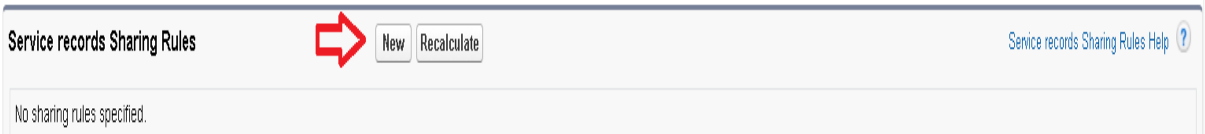
SETUP
Sharing Settings

Web Cart Document	Private	Private	✓
Work Order	Private	Private	✓
Work Plan	Private	Private	✓
Work Plan Template	Private	Private	✓
Work Step Template	Private	Private	✓
Work Type	Private	Private	✓
Work Type Group	Public Read/Write	Private	✓
Appointment	Public Read/Write	Private	✓
Billing details and feedback	Public Read/Write	Private	✓
Customer Details	Public Read/Write	Private	✓
Service records	Private	Private	✓

Other Settings

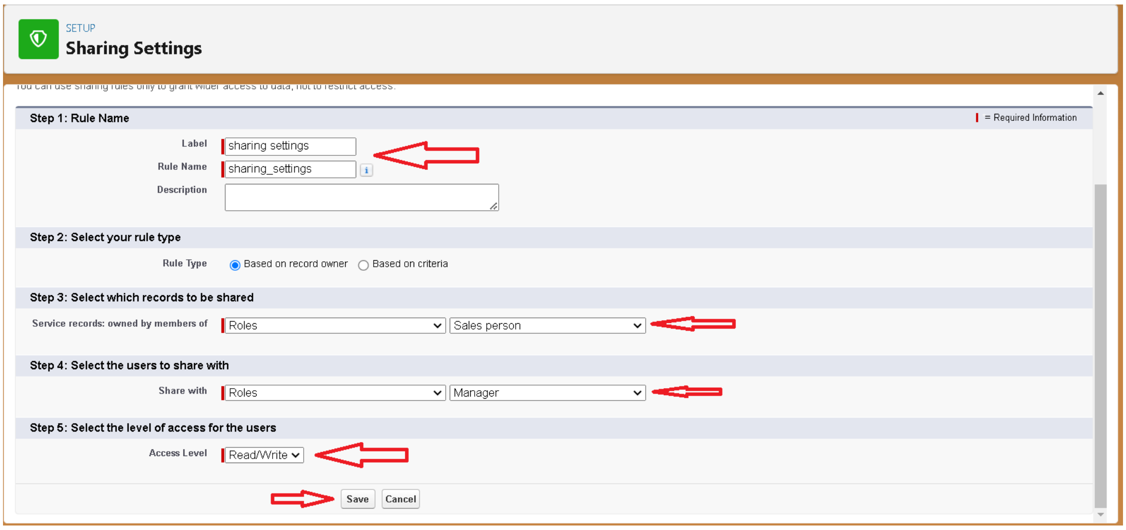
Manager Groups ☐
Secure guest user record access ☑
Require permission to view record names in lookup fields ☐

- Scroll down a bit, Click new on Service records sharing Rules



No sharing rules specified.

1. Give the Label name as “ Sharing setting”
2. Rule name is auto populated.
3. In step 3 : Select which records to be shared, members of “ Roles ” >> “ Sales person”
4. In step 4: share with, select “ Roles ” >> “ Manager ”
5. In step 5 : Change the access level to “ Read / write ”.
6. Click on save.


SETUP
Sharing Settings

You can use sharing rules only to grant wider access to data, not to restrict access.

Step 1: Rule Name | = Required Information

Label: sharing settings
Rule Name: sharing_settings
Description:

Step 2: Select your rule type

Rule Type: ● Based on record owner ○ Based on criteria

Step 3: Select which records to be shared

Service records: owned by members of: Roles | Sales person

Step 4: Select the users to share with

Share with: Roles | Manager

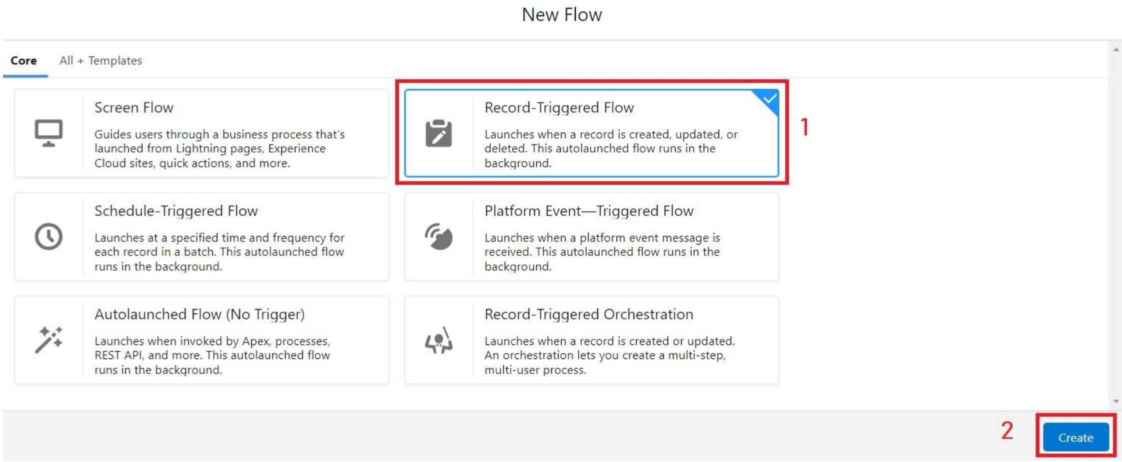
Step 5: Select the level of access for the users

Access Level: Read/Write

Activity – 13

Flows: a powerful tool that allows you to automate business processes, collect and update data, and guide users through a series of screens or steps.

- **Record-Trigger Flow**Select the Record-triggered flow and Click on Create.



New Flow

Screen Flow
Guides users through a business process that's launched from Lightning pages, Experience Cloud sites, quick actions, and more.

Record-Triggered Flow
Launches when a record is created, updated, or deleted. This autolaunched flow runs in the background.

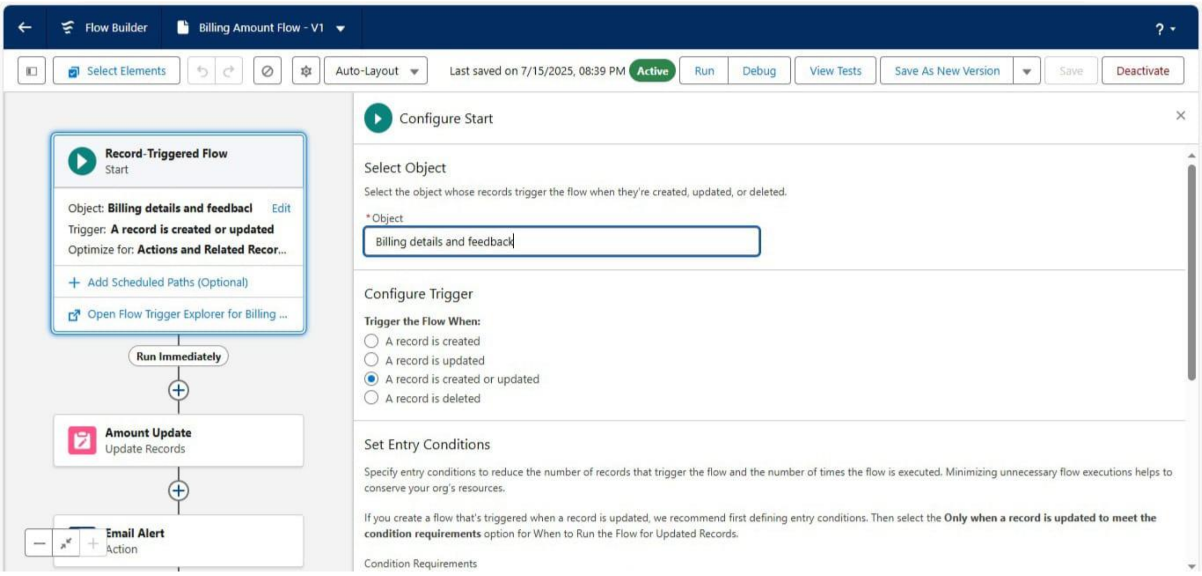
Schedule-Triggered Flow
Launches at a specified time and frequency for each record in a batch. This autolaunched flow runs in the background.

Platform Event—Triggered Flow
Launches when a platform event message is received. This autolaunched flow runs in the background.

Autolaunched Flow (No Trigger)
Launches when invoked by Apex, processes, REST API, and more. This autolaunched flow runs in the background.

Record-Triggered Orchestration
Launches when a record is created or updated. An orchestration lets you create a multi-step, multi-user process.

-
- Select the Object as “Billing details and feedback”in the Drop down list.
- Select the Trigger Flow when: “A record is Created or Updated”.
- Select the Optimize the flow for: “Actions and Related Records” and Click on Done.



Last saved on 7/15/2025, 08:39 PM

Object: **Billing details and feedbacl**
Trigger: **A record is created or updated**
Optimize for: **Actions and Related Recor...**

Configure Start

Select Object

Select the object whose records trigger the flow when they're created, updated, or deleted.

*Object: Billing details and feedback

Configure Trigger

Trigger the Flow When:
○ A record is created
○ A record is updated
● A record is created or updated
○ A record is deleted

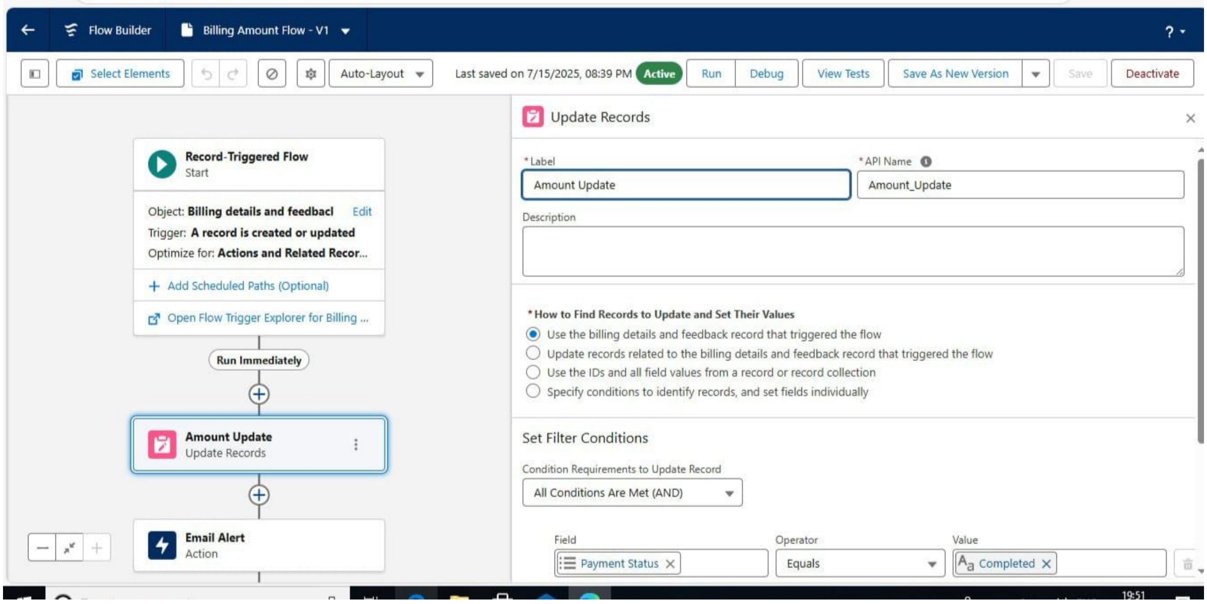
Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements

- **Amount Update Flow**
  - Field : Payment_Paid__c
  - Value : {!$Record.Service_records__r.Appointment__r.Service_Amount__c}



Update Records

*Label: Amount Update

*API Name: Amount_Update

Description

*How to Find Records to Update and Set Their Values

● Use the billing details and feedback record that triggered the flow

○ Update records related to the billing details and feedback record that triggered the flow

○ Use the IDs and all field values from a record or record collection

○ Specify conditions to identify records, and set fields individually

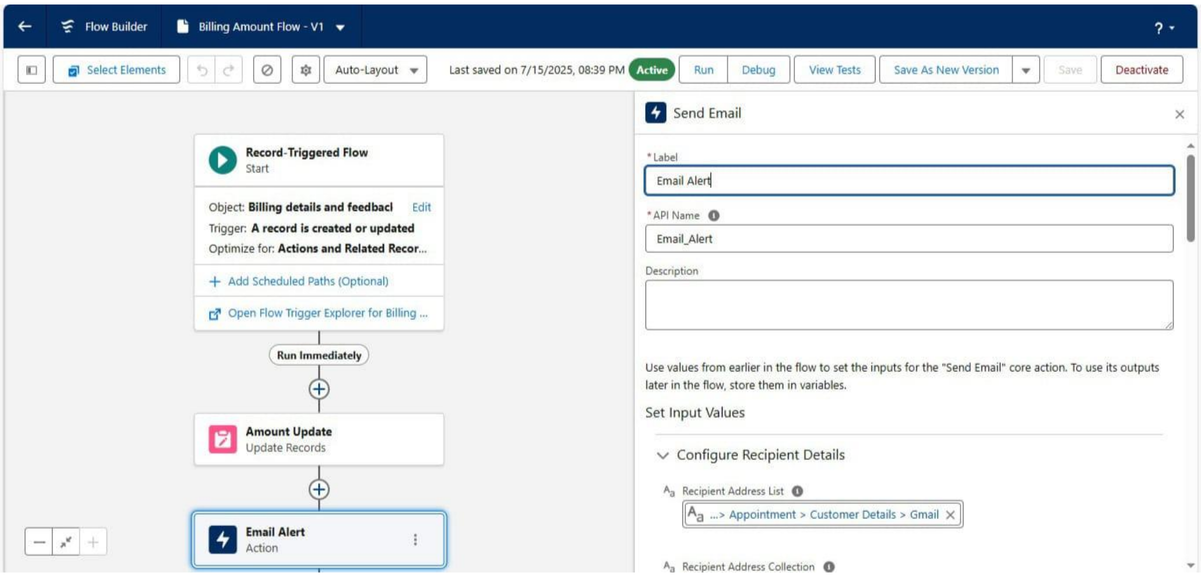
Set Filter Conditions

Condition Requirements to Update Record: All Conditions Are Met (AND)

Field	Operator	Value
Payment Status	Equals	Completed

- Email Alert Flow

- RecipientAddressList: {!$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail__c}



Send Email

*Label: Email Alert

*API Name: Email_Alert

Description

Use values from earlier in the flow to set the inputs for the "Send Email" core action. To use its outputs later in the flow, store them in variables.

Set Input Values

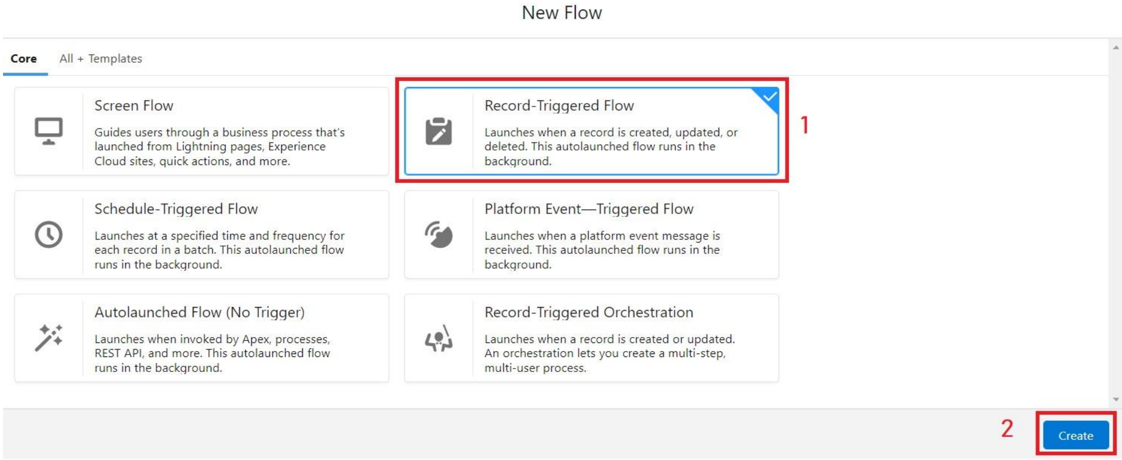
Configure Recipient Details

Recipient Address List: ...> Appointment > Customer Details > Gmail

Recipient Address Collection

Create Another Flow:

1. Select the Record-triggered flow and Click on Create.



New Flow

Screen Flow
Guides users through a business process that's launched from Lightning pages, Experience Cloud sites, quick actions, and more.

Record-Triggered Flow
Launches when a record is created, updated, or deleted. This autolaunched flow runs in the background.

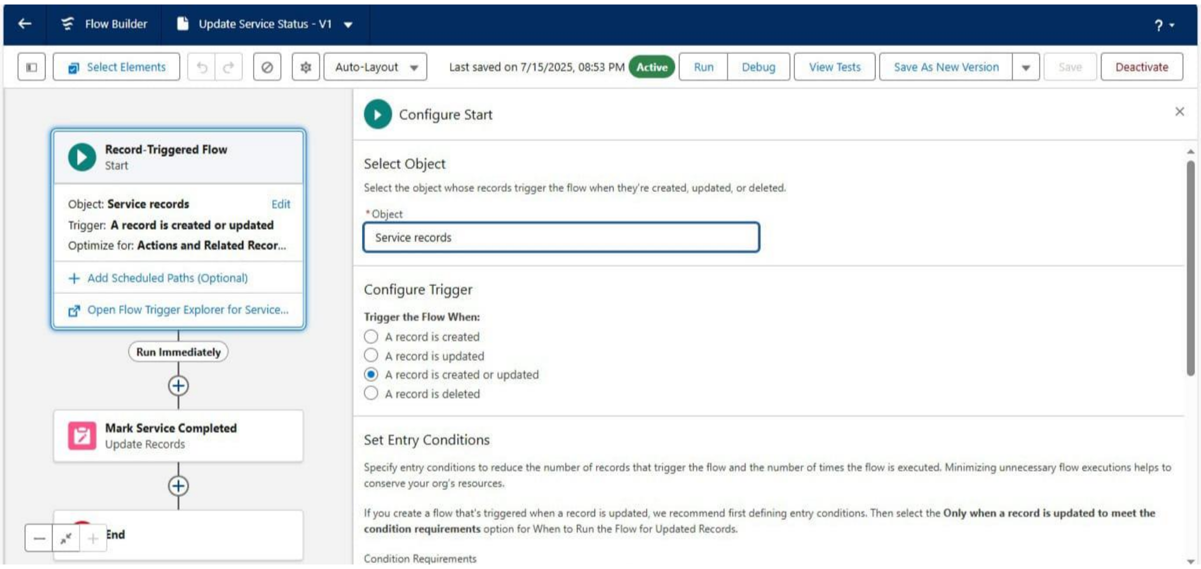
Schedule-Triggered Flow
Launches at a specified time and frequency for each record in a batch. This autolaunched flow runs in the background.

Platform Event—Triggered Flow
Launches when a platform event message is received. This autolaunched flow runs in the background.

Autolaunched Flow (No Trigger)
Launches when invoked by Apex, processes, REST API, and more. This autolaunched flow runs in the background.

Record-Triggered Orchestration
Launches when a record is created or updated. An orchestration lets you create a multi-step, multi-user process.

2.
3. Select the Object as “ Service records”in the Drop down list.
4. Select the Trigger Flow when: “A record is Created or Updated”.
5. Select the Optimise the flow for: “Actions and Related Records”



Last saved on 7/15/2025, 08:53 PM

Record-Triggered Flow
Start
Object: **Service records** Edit
Trigger: **A record is created or updated**
Optimize for: **Actions and Related Recor...**
+ Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Service...

Mark Service Completed
Update Records

Configure Start

Select Object
Select the object whose records trigger the flow when they're created, updated, or deleted.
*Object
Service records

Configure Trigger
Trigger the Flow When:
○ A record is created
○ A record is updated
● A record is created or updated
○ A record is deleted

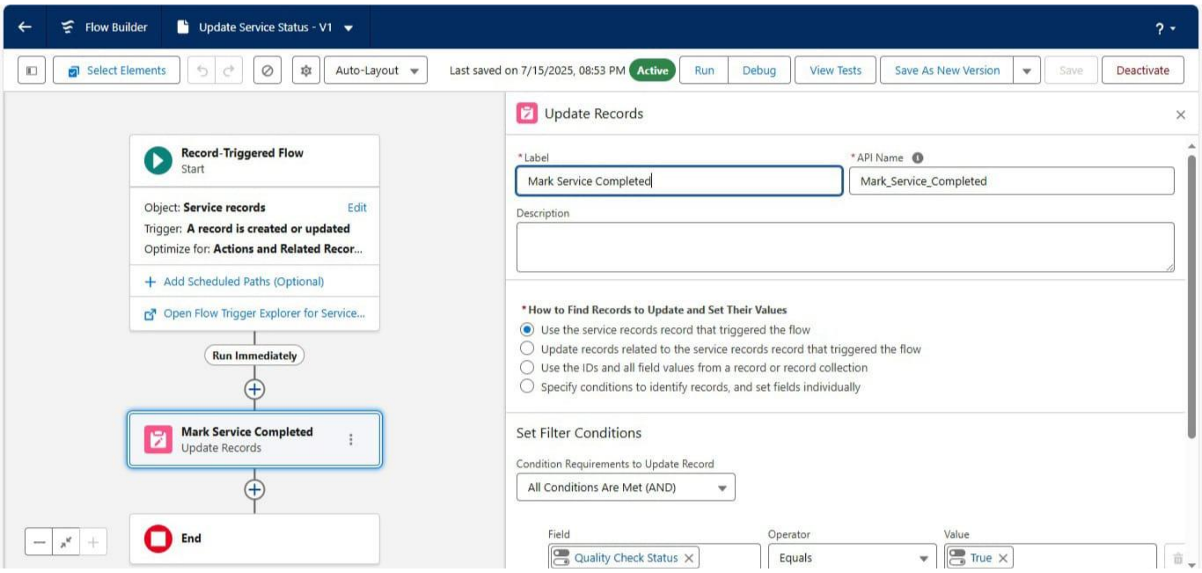
Set Entry Conditions
Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements

Set a filter condition : All Conditions are met(AND)

1. Field : Quality_Check_Status__c
2. Operator : Equals
3. Value : True
4. And Set Field Values for the Billing details and feedback Record
5. Field : Service_Status__c
6. Value : Completed



Record-Triggered Flow
Start

Object: **Service records**
Trigger: **A record is created or updated**
Optimize for: **Actions and Related Recor...**

Mark Service Completed
Update Records

Update Records

*Label: Mark Service Completed
*API Name: Mark_Service_Completed
Description

***How to Find Records to Update and Set Their Values**
- ● Use the service records record that triggered the flow
- ○ Update records related to the service records record that triggered the flow
- ○ Use the IDs and all field values from a record or record collection
- ○ Specify conditions to identify records, and set fields individually

Set Filter Conditions

Condition Requirements to Update Record: All Conditions Are Met (AND)

Field	Operator	Value
Quality Check Status	Equals	True

Activity – 14

Triggers

1. Name the class as "AmountDistributionHandler ".

Code:

```
public class AmountDistributionHandler {

    public static void amountDist(list<Appointment__c> listApp){
        list<Service_records__c> serList = new list <Service_records__c>();

        for(Appointment__c app : listApp){
            if(app.Maintenance_service__c == true && app.Repairs__c == true &&
app.Replacement_Parts__c == true){
                app.Service_Amount__c = 10000;
            }
            else if(app.Maintenance_service__c == true && app.Repairs__c == true){
                app.Service_Amount__c = 5000;
            }
            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c ==
true){
                app.Service_Amount__c = 8000;
            }
            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 7000;
            }
            else if(app.Maintenance_service__c == true){
                app.Service_Amount__c = 2000;
```

```
            }
            else if(app.Repairs__c == true){
                app.Service_Amount__c = 3000;
            }
            else if(app.Replacement_Parts__c == true){
                app.Service_Amount__c = 5000;
            }

    }
    }
}
```

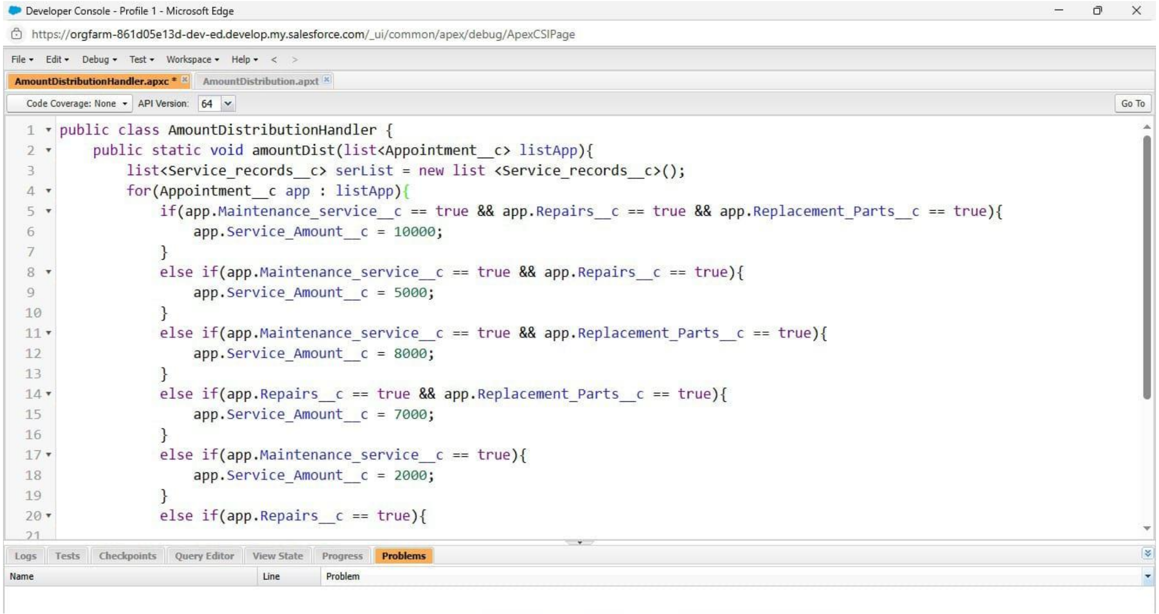
Trigger Handler :

1. Name : AmountDistribution
2. sObject : Appointment__c

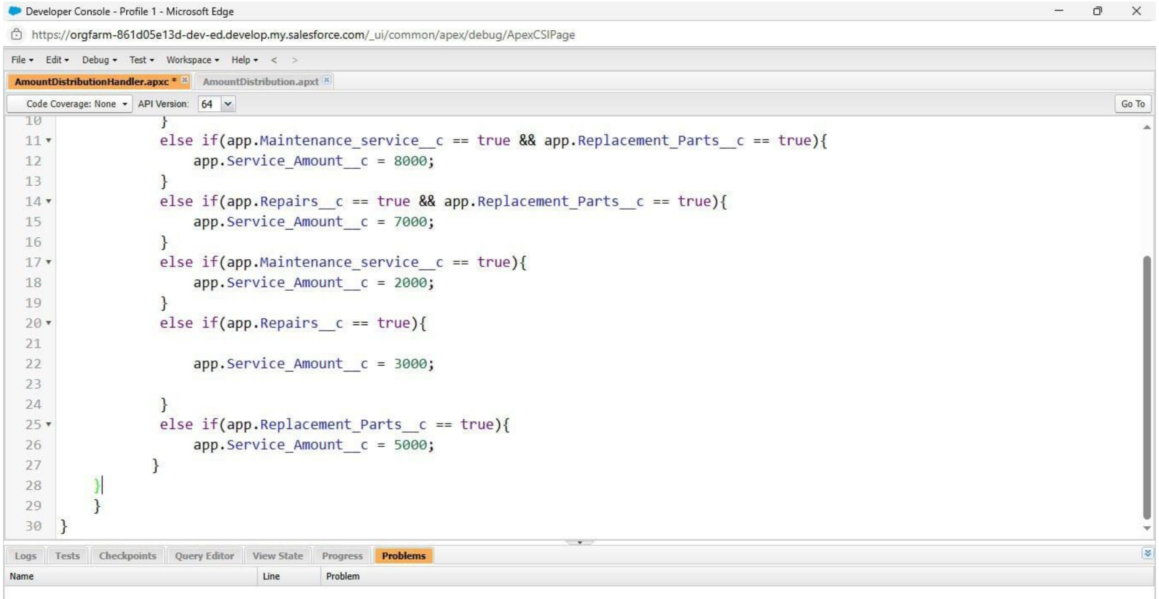
Code:

```
trigger AmountDistribution on Appointment__c (before insert, before update) {

    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
        AmountDistributionHandler.amountDist(trigger.new);

    }


}
```

- ## AmountDistributionHandler.apxc (Class)

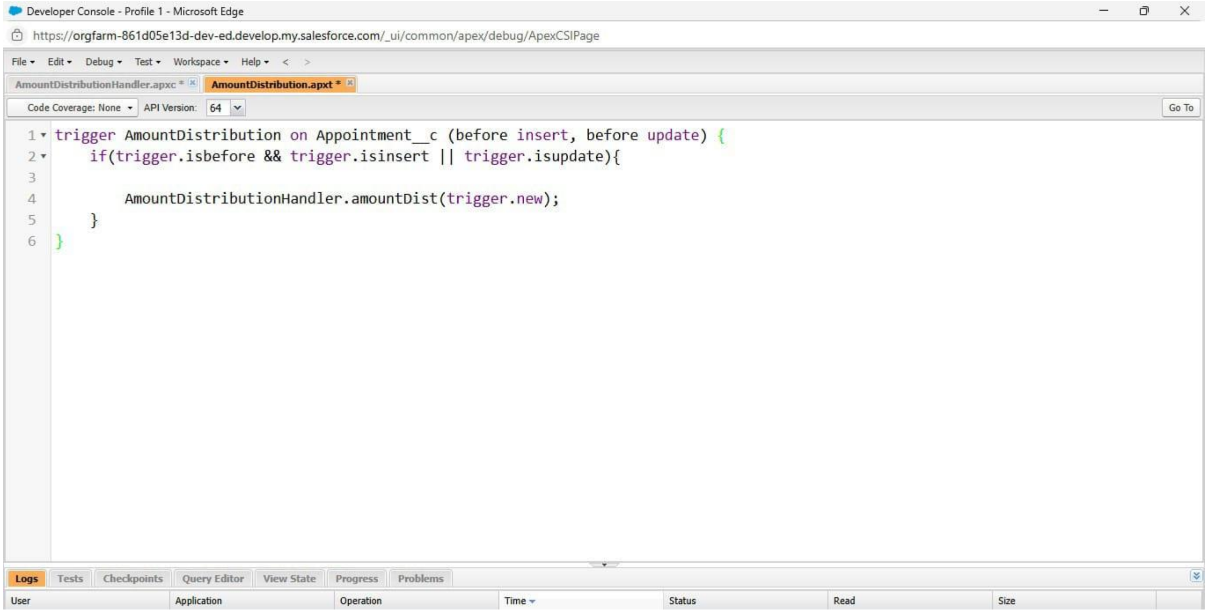


```
public class AmountDistributionHandler {
    public static void amountDist(list<Appointment__c> listApp){
        list<Service_records__c> serList = new list <Service_records__c>();
        for(Appointment__c app : listApp){
            if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 10000;
            }
            else if(app.Maintenance_service__c == true && app.Repairs__c == true){
                app.Service_Amount__c = 5000;
            }
            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 8000;
            }
            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 7000;
            }
            else if(app.Maintenance_service__c == true){
                app.Service_Amount__c = 2000;
            }
            else if(app.Repairs__c == true){
```



```
            }
            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 8000;
            }
            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 7000;
            }
            else if(app.Maintenance_service__c == true){
                app.Service_Amount__c = 2000;
            }
            else if(app.Repairs__c == true){

                app.Service_Amount__c = 3000;

            }
            else if(app.Replacement_Parts__c == true){
                app.Service_Amount__c = 5000;
           }
    }
    }
}
```

- ## AmountDistribution.apxt (Trigger)

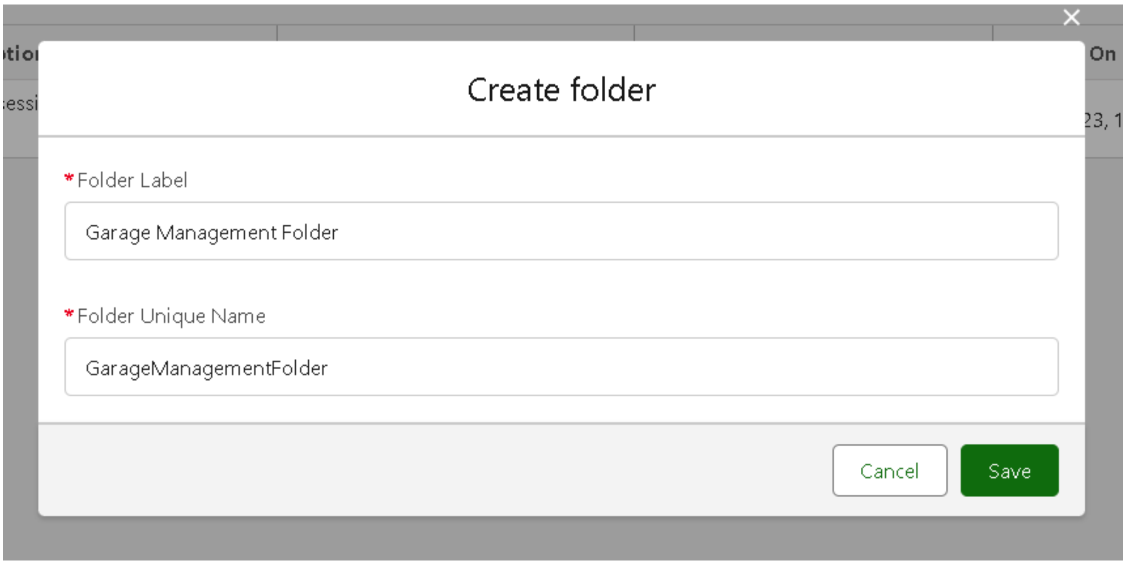


```
trigger AmountDistribution on Appointment__c (before insert, before update) {
    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){

        AmountDistributionHandler.amountDist(trigger.new);
    }
}
```

Activity – 15

Reports

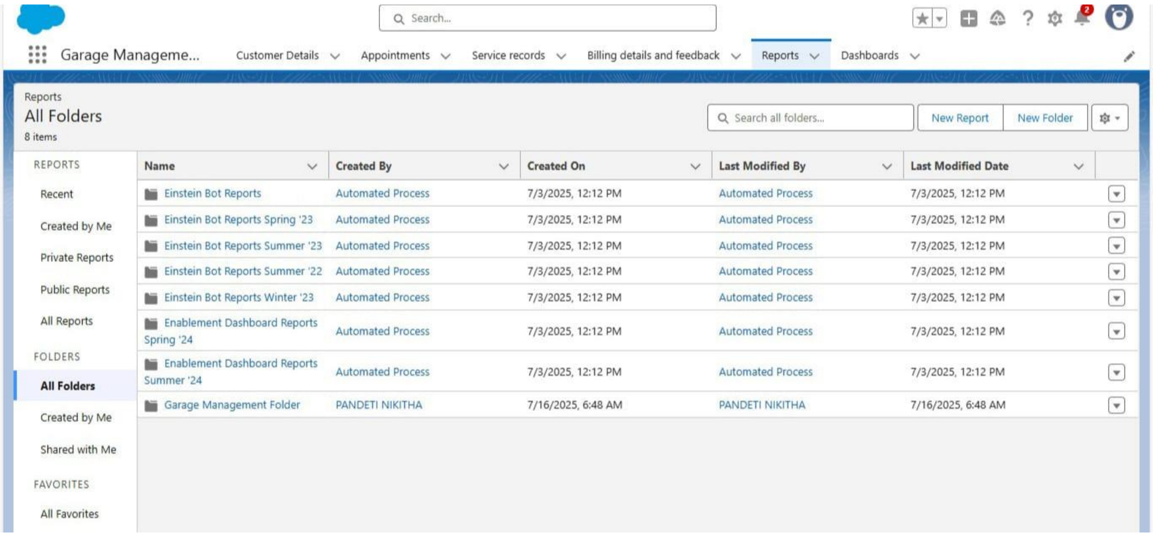
- ### Report Folder Creation
- Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.



* Folder Label: Garage Management Folder

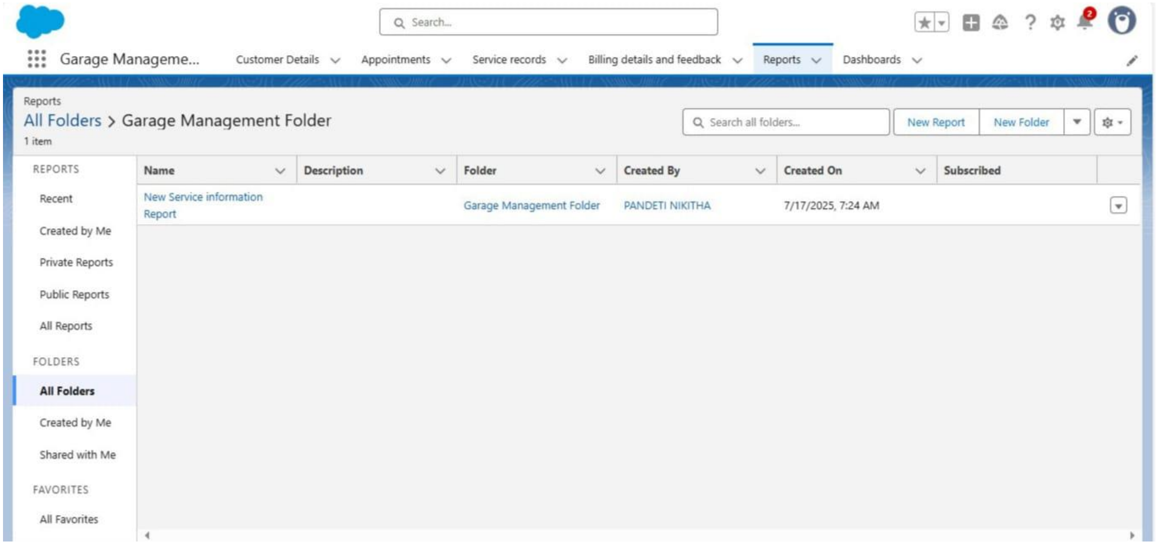
* Folder Unique Name: GarageManagementFolder

-



Reports
All Folders
8 items

Name	Created By	Created On	Last Modified By	Last Modified Date
Einstein Bot Reports	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
Einstein Bot Reports Spring '23	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
Einstein Bot Reports Summer '23	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
Einstein Bot Reports Summer '22	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
Einstein Bot Reports Winter '23	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
Enablement Dashboard Reports Spring '24	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
Enablement Dashboard Reports Summer '24	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
Garage Management Folder	PANDETI NIKITHA	7/16/2025, 6:48 AM	PANDETI NIKITHA	7/16/2025, 6:48 AM



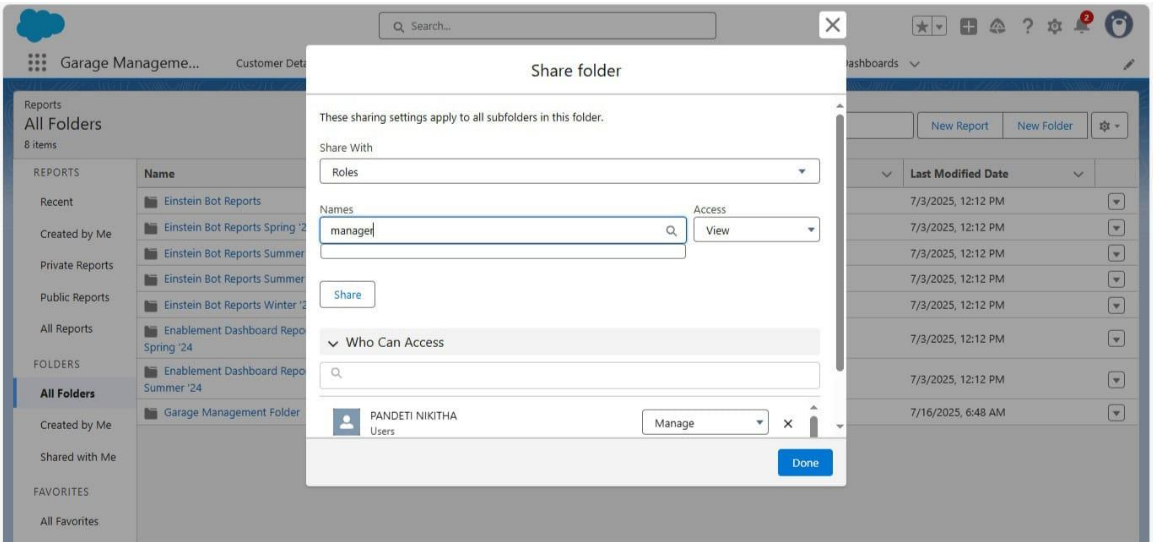
Reports
All Folders > Garage Management Folder
1 item

Name	Description	Folder	Created By	Created On	Subscribed
New Service information Report		Garage Management Folder	PANDETI NIKITHA	7/17/2025, 7:24 AM	

-

- Sharing a folder:

- Select the share with as “roles”, in name field search for “manager”, give “view” as access for that role.



Share folder

These sharing settings apply to all subfolders in this folder.

Share With: Roles

Names: manager

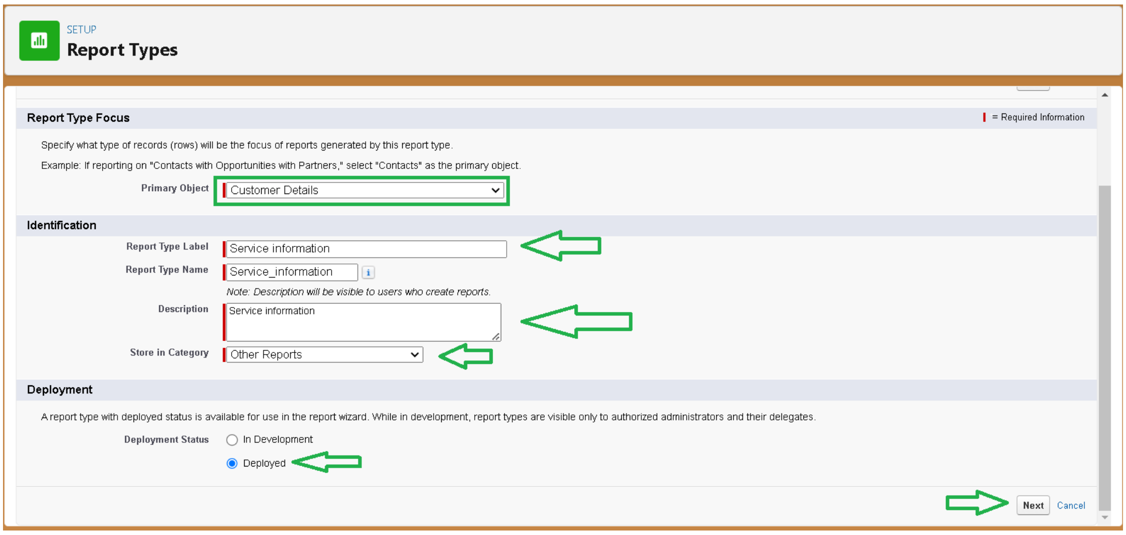
Access: View

Who Can Access

PANDETI NIKITHA — Users — Manage

-

- Report Types:

1. Select the Primary object as “ Customer details” .
2. Give the Report type Label as “ Service information ”
3. Report type Name is autopopulated.
4. Keep the Description as same.
5. Select Store in Category as “ other Reports ”
6. Select the deployment status as “ Depolyed ”, click on Next.
7. 

SETUP
Report Types

Report Type Focus

= Required Information

Specify what type of records (rows) will be the focus of reports generated by this report type.

Example: If reporting on "Contacts with Opportunities with Partners," select "Contacts" as the primary object.

Primary Object: Customer Details

Identification

Report Type Label: Service information

Report Type Name: Service_information

Note: Description will be visible to users who create reports.

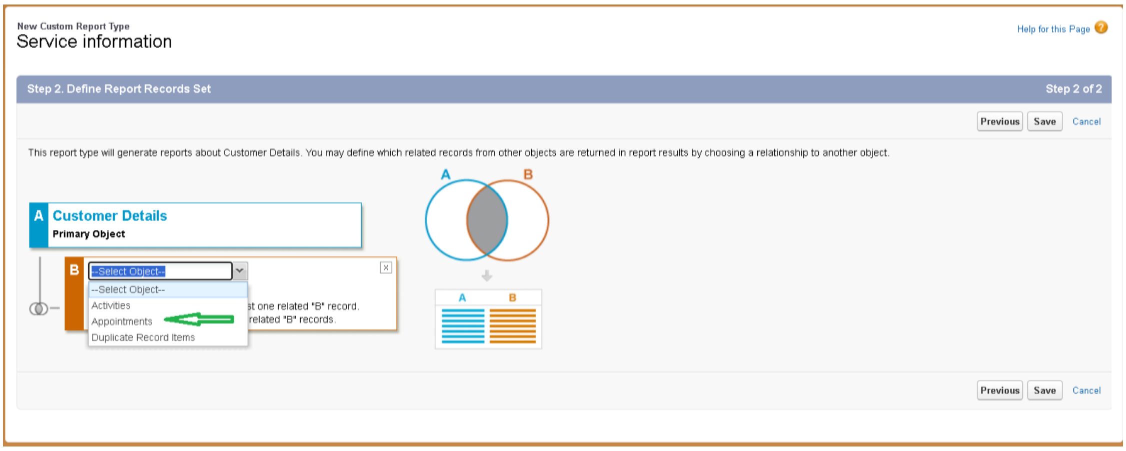
Description: Service information

Store in Category: Other Reports

Deployment

A report type with deployed status is available for use in the report wizard. While in development, report types are visible only to authorized administrators and their delegates.

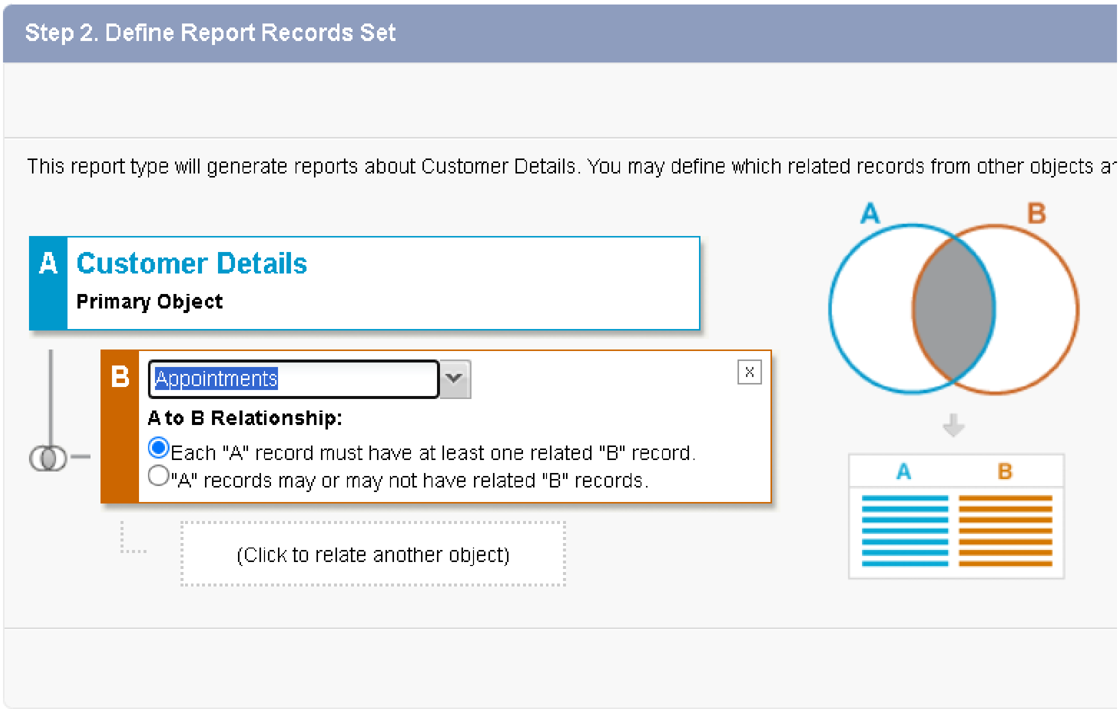
Deployment Status: ○ In Development ● Deployed

8. now , Click on Related object box.
9. Click on Select Object, choose Appointment Object as shown in fig.
10. 

New Custom Report Type
Service information

Step 2. Define Report Records Set Step 2 of 2

This report type will generate reports about Customer Details. You may define which related records from other objects are returned in report results by choosing a relationship to another object.



Step 2. Define Report Records Set

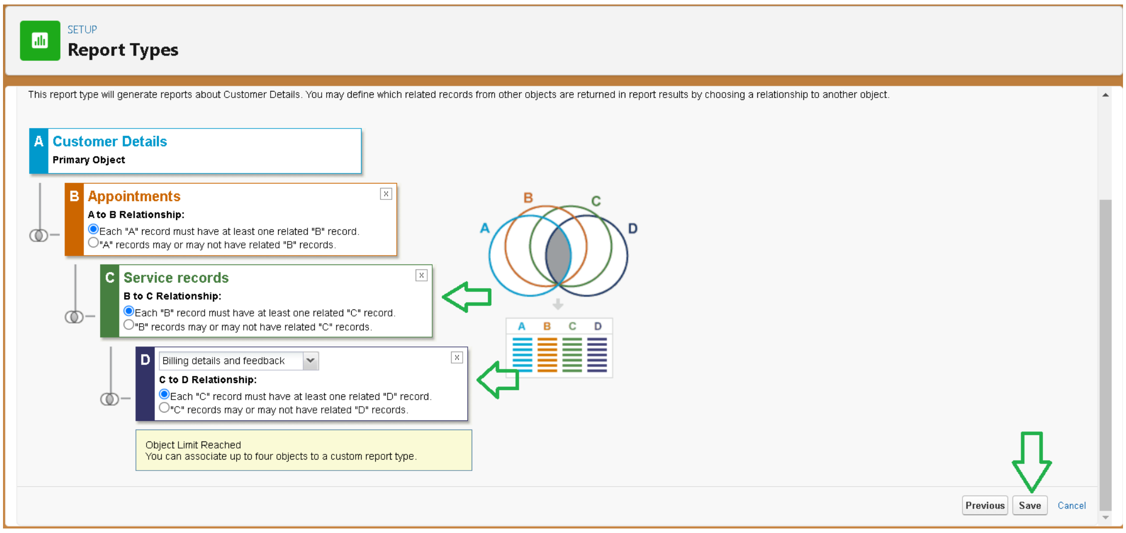
This report type will generate reports about Customer Details. You may define which related records from other objects a

A **Customer Details**
Primary Object

B Appointments
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.

(Click to relate another object)

11.
12. Again Click to relate another object.
13. And select the related object as “ service records”.
14. Repeat the process and select the related object as “ Billing details and feedback”.
15.



SETUP
Report Types

This report type will generate reports about Customer Details. You may define which related records from other objects are returned in report results by choosing a relationship to another object.

A **Customer Details**
Primary Object

B **Appointments**
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.

C **Service records**
B to C Relationship:
● Each "B" record must have at least one related "C" record.
○ "B" records may or may not have related "C" records.

D Billing details and feedback
C to D Relationship:
● Each "C" record must have at least one related "D" record.
○ "C" records may or may not have related "D" records.

Object Limit Reached
You can associate up to four objects to a custom report type.