Garage Management System

By

Tanjul Sarathe

sarathetanjul@gmail.com

College: VIT Bhopal University

Index Page

1. Project Overview

The Garage Management System is a valuable tool for automotive repair facilities, helping them deliver top-notch service, increase operational efficiency, and build lasting customer relationships. With its user-friendly interface and powerful features, GMS empowers garages to thrive in a competitive market while ensuring a seamless and satisfying experience for both customers and staff.

2. Objectives

Business Goals:

a. Enhance customer satisfaction by providing a seamless experience.
b. Improveoperational efficiency by automating scheduling, service tracking, and payment processes.
c. Centralize data for better decision-making and reporting.

Specific Outcomes:

a. Implement a digital solutionfor managing customerdetails, appointments, service records, and billing.
b. Provide real-time insights through reports and dashboards.
c. Automate workflows to reduce manualeffort and errors.

3. Salesforce Key Features and Concepts Utilized

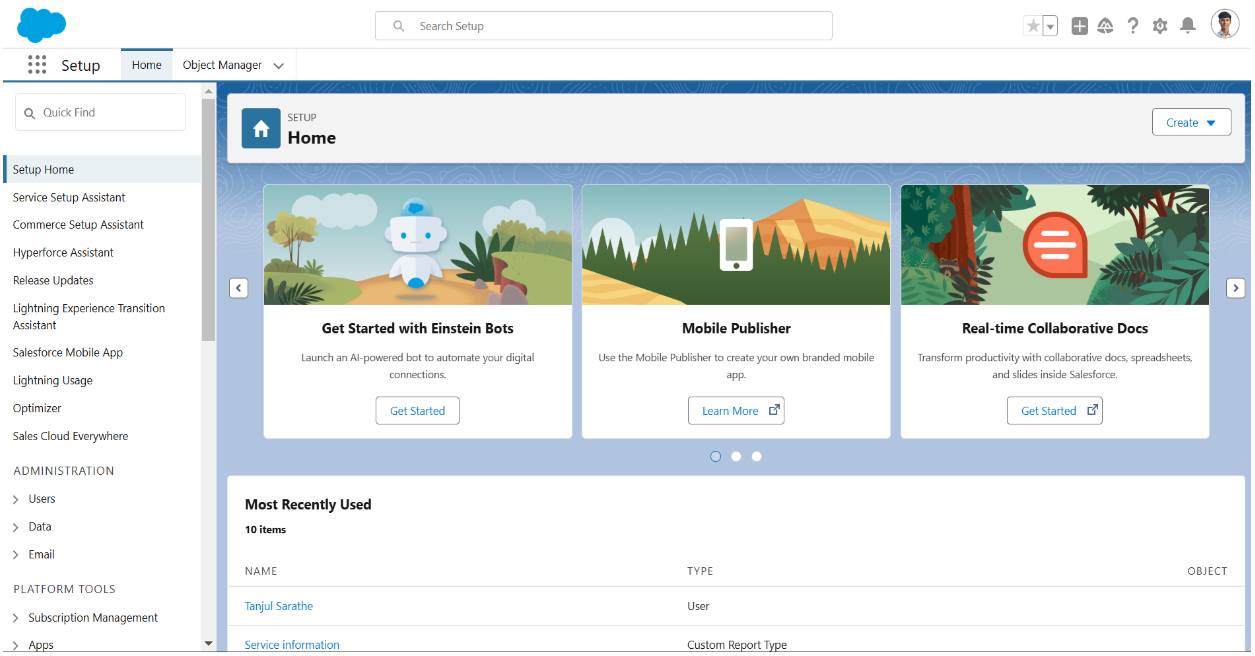
a. **Custom Objects**: CustomerDetails, Appointments, ServiceRecords, Billing Details& Feedback.
b. **Custom Tabs**: User-friendly navigation for accessing key objects.
c. **LightningApp**: Centralized access to essentialtools for the garage management process.
d. **Custom Fields**: Multiplefield types like lookup, picklist,checkbox, and formulafields to ensure robust data capture.
e. **ValidationRules**: Enforceddata integrity and error prevention.
f. **Profiles& Roles**: Role-basedaccess control for managers and salespersons.
g. **Public Groups**: Simplified team-based access to records.
h. **Flows**: Automation of record updates and email alerts.
i. **Apex Triggers**: Automated calculation of service amounts.
j. **Reports& Dashboards**: Data visualization and performance tracking.

4. Detailed Steps to Solution Design

Step 1: Creatingthe Developer Account

1. **Sign Up** at SalesforceDeveloper Signup.
2. **Fill in details**: Name, email, role as 'Developer', and company as 'College Name'.
3. **Username** format: username@organization.com.(eg.- tanjul@vitbhopal.com)
4. **Activate the account** through the email link.

---> Your Setup page would look like this:



Step 2: Creating Custom Objects

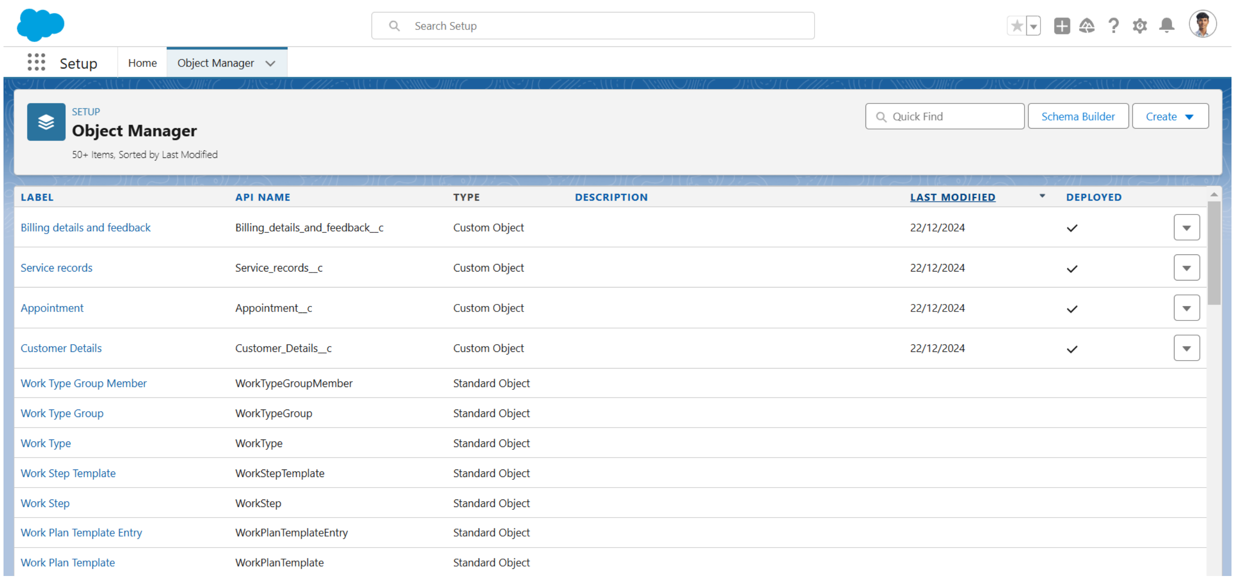
1. To Navigate to Setup page:Click on gear icon ? click setup.
2. To createan object: From the setuppage > Clickon Object Manager> Click on Create > Click on Custom Object.
3. On Custom object definingpage: Enter the label name, plural label name, click on Allow reports, Allow search.
4. Click on Save.

For our project we need to create four objects

- **CustomerDetails**: Text-basedcustomer names with search and report options.
- **Appointment**: Auto-numbering system with fields like appointment name and display format.

- **Service Records**: Auto-numbered service records with fields for tracking service status.

- **BillingDetails & Feedback**: Auto-numbered billing recordswith fields for payment tracking and feedback.



Step 3: Custom Tabs

a. Custom tabs for **Customer Details**, **Appointments**, **Service Records**, and **BillingDetails & Feedback**.

b. Tabs are displayed for easy navigation and user access.

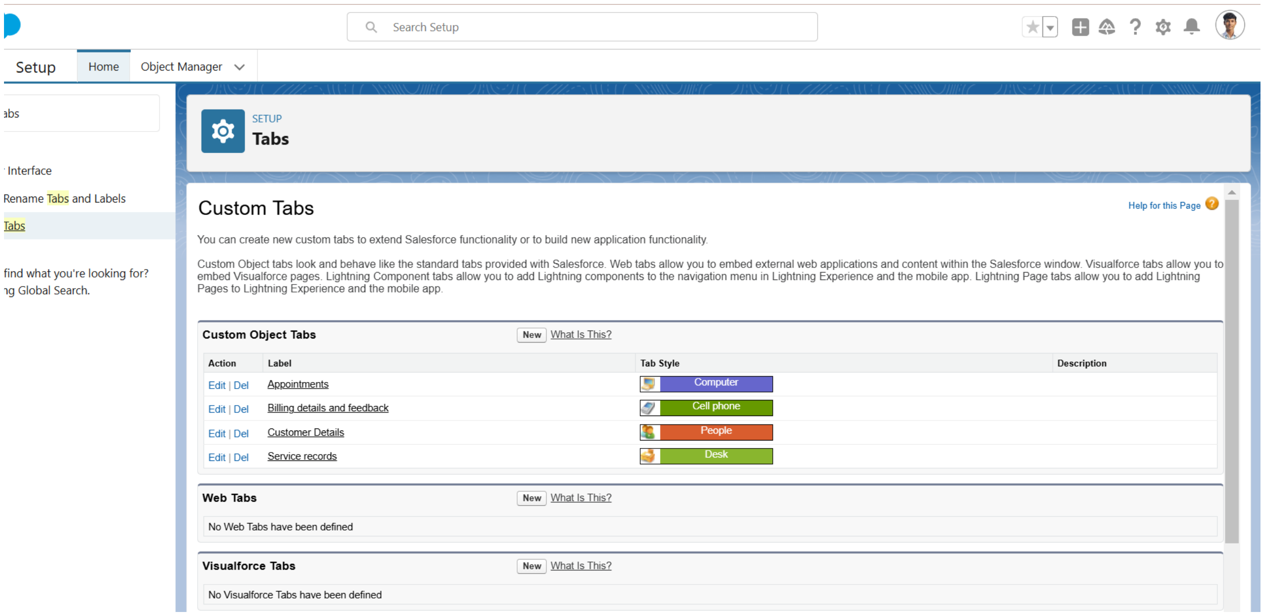
I followed these 4 steps to create all requiredcustom tab

(i) Go to setup page >>type Tabs in Quick Find bar >>click on tabs >> New (under custom object tab).

(ii) Select Object(Customer Details)>> Select the tab style >> Next (Add to profiles page) keep it as default>> Next (Add to CustomApp) uncheck the include tab.

(iii) Make sure that the Appendtab to users' existing personalcustomizations is checked.

(iv) Click save.

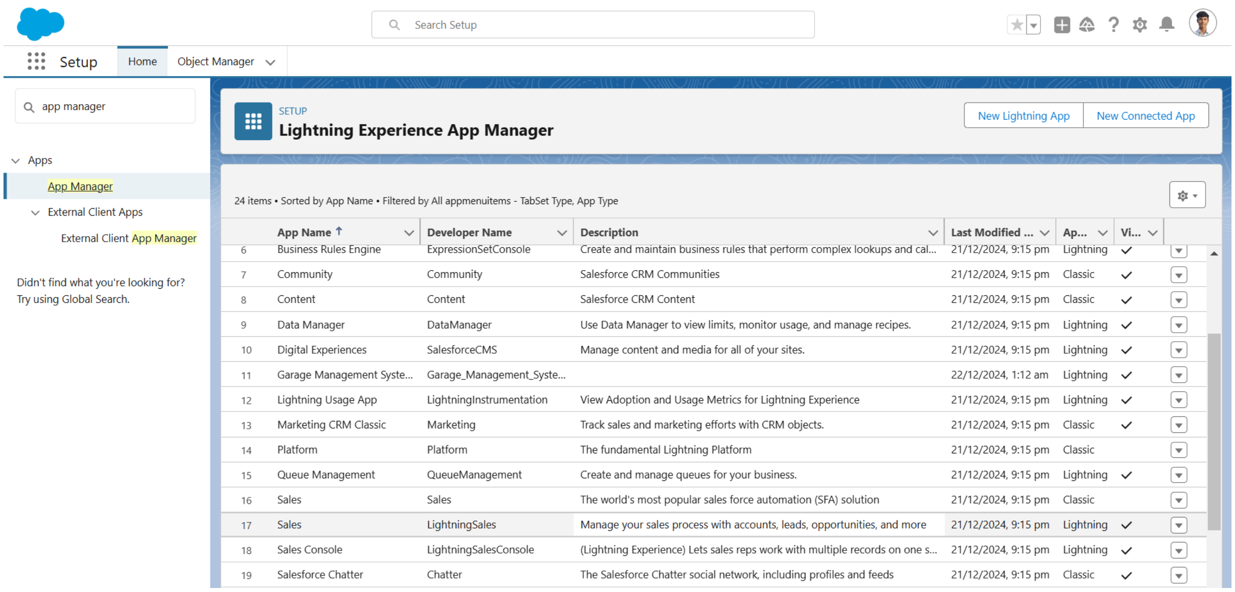


Step 4: Building the Lightning App

1. **Name**: Garage Management Application.
2. **Navigation Items**: Customer Details, Appointments, Service Records, Billing,Reports, and Dashboards.
3. **User Profiles**: Assign access to the System Administrator role.

To create a lightningapp page:

1. Go to setuppage >> search “app manager” in quick find >> select “app manager” >>click on New lightning App.
2. Fillthe 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.
3. To Add Navigation Items
4. Select the items (Customer Details,Appointments, Service records, Billing details and feedback, Reportsand Dashboards) from the searchbar and move it usingthe arrow button >> Next.
5. To Add User Profiles: Searchprofiles (System administrator) in the searchbar >> clickon the arrow button >> save & finish.



Step 5: Creating Custom Fields

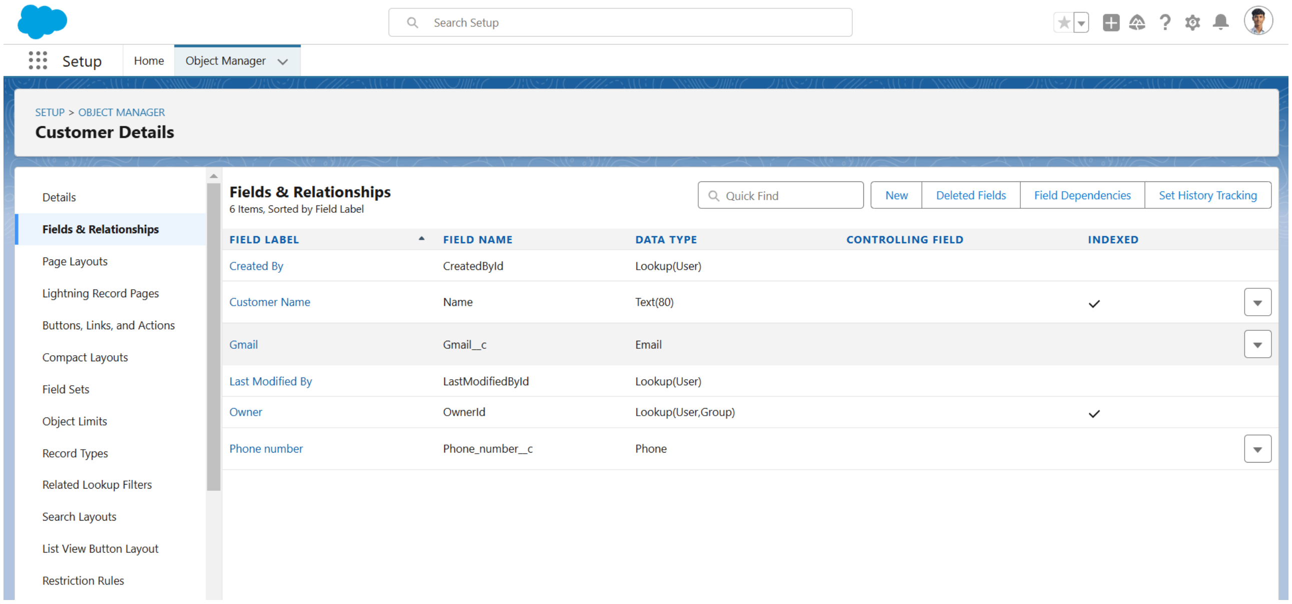
a. **Customer Details**: Phone and email fields.

1. To create fields in an object:

- Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
- Now click on "Fields & Relationships" >> New
- Select Data Type as a "Phone"
- Click on next.
- Fill the Above as following:
  - Field Label: Phone number
  - Field Name: gets auto generated
- Click on Next >> Next >> Save and new.

2. To create another fields in an object:

- Go to setup >> click on Object Manager >> type object name (Customer Details) in search bar >> click on the object.
- Now click on "Fields & Relationships" >> New
- Select Data type as a "Email" and Click on Next
- Fill the Above as following:
  - Field Label: Gmail
  - Field Name: gets auto generated
- Click on Next >> Next >> Save and new.



SETUP > OBJECT MANAGER

Customer Details

Fields & Relationships

6 Items, Sorted by Field Label

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

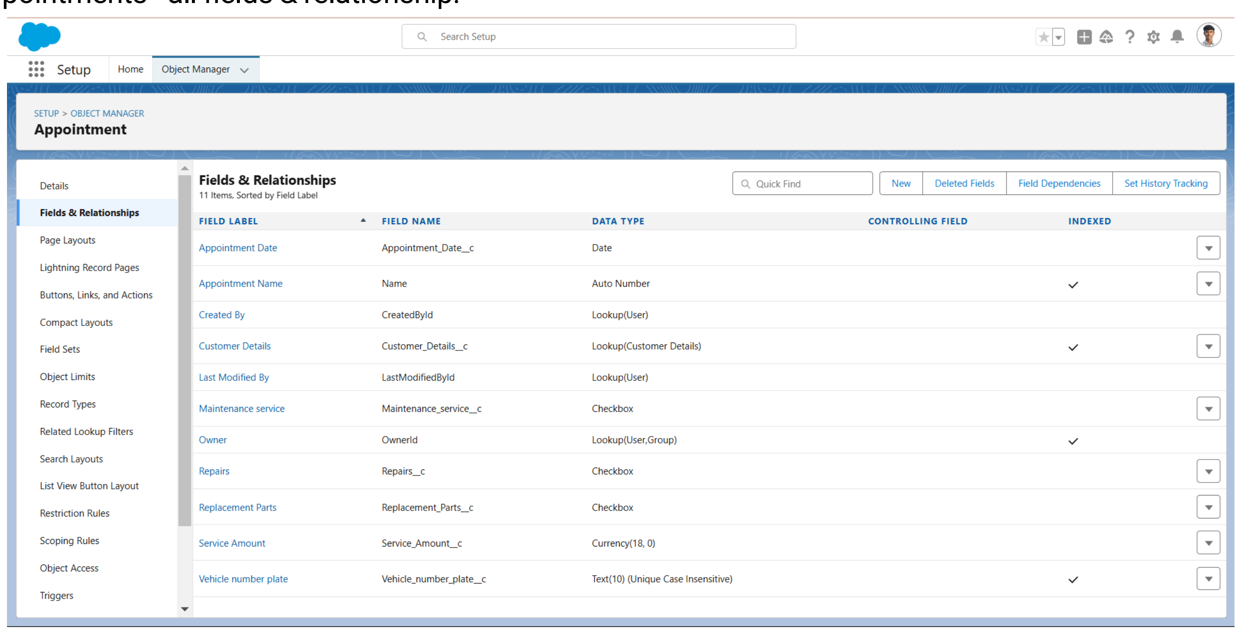
b. **Lookup Fields**: Link Appointments to Customers, ServiceRecords to Appointments, and Billing Details to Service Records.

c. **PicklistFields**: Service status(Started, Completed) and payment status(Pending, Completed).

d. **Formula Fields**: Calculate service date from createddate.

e. **Text Fields**: Vehiclenumber plate (10 characters, unique)and customer feedbackrating (1 character).

Appointments - all fields & relationship:



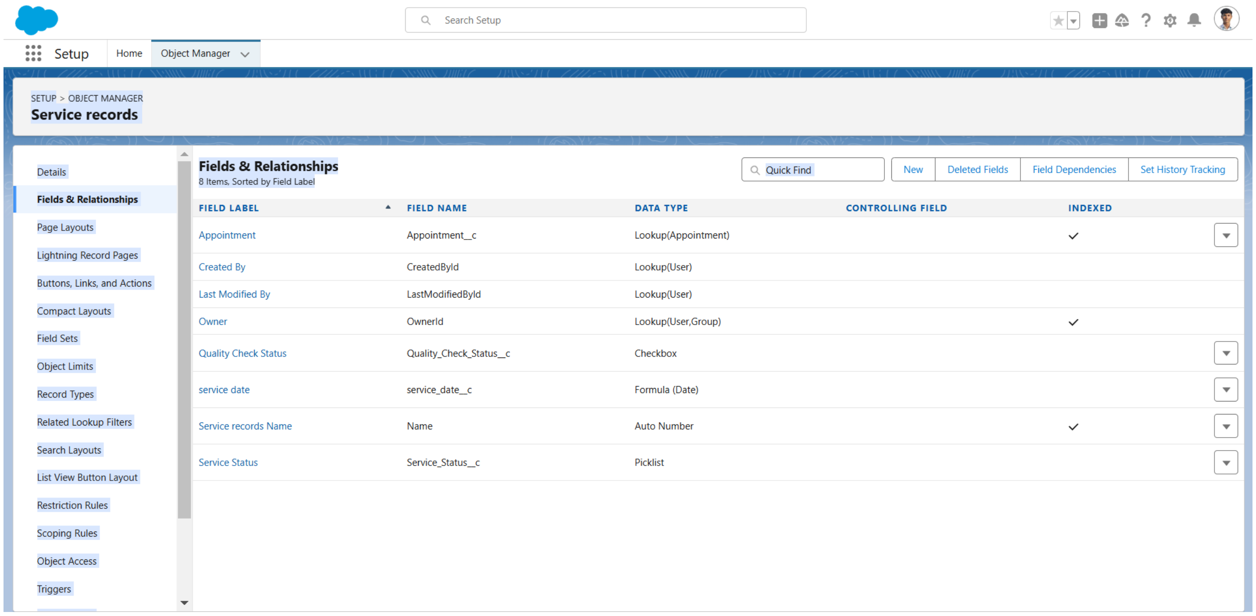
SETUP > OBJECT MANAGER

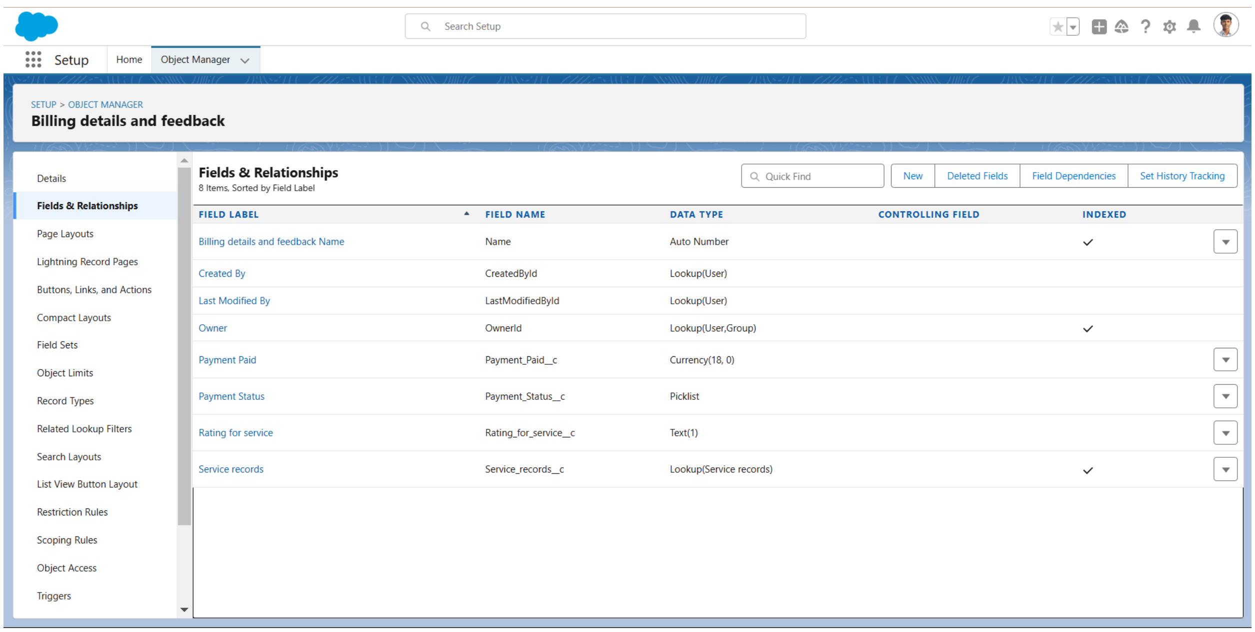
Appointment

Fields & Relationships

11 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)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Maintenance service	Maintenance_service__c	Checkbox		
Owner	OwnerId	Lookup(User,Group)		✓
Repairs	Repairs__c	Checkbox		
Replacement Parts	Replacement_Parts__c	Checkbox		
Service Amount	Service_Amount__c	Currency(18, 0)		
Vehicle number plate	Vehicle_number_plate__c	Text(10) (Unique Case Insensitive)		✓





Step 6: Validation Rules

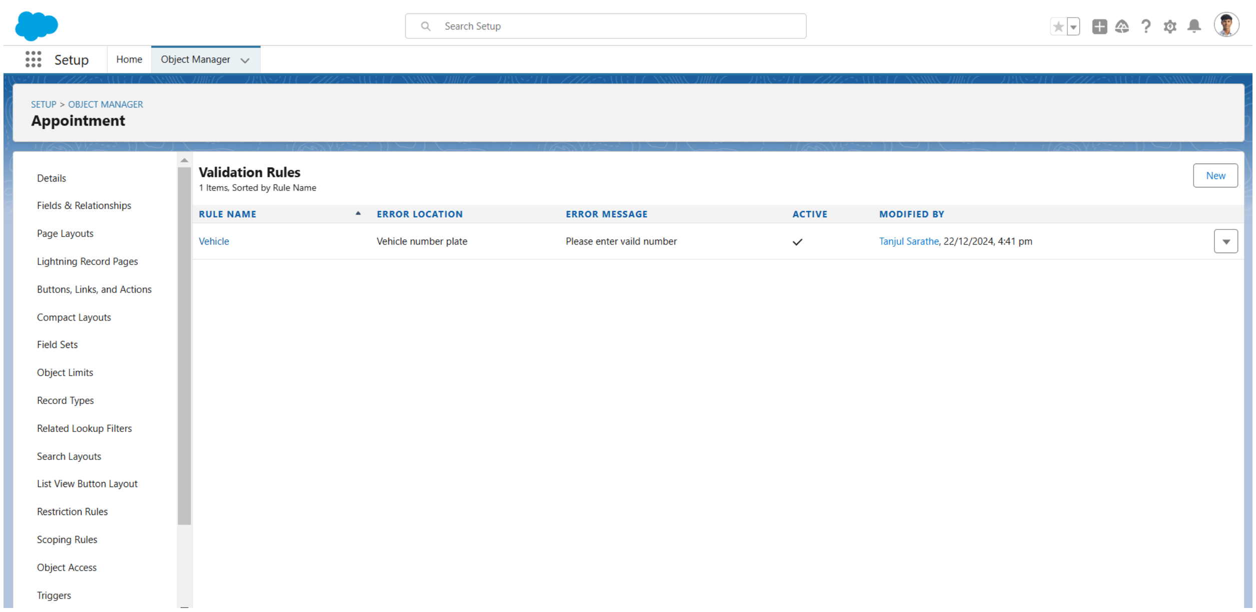
Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria.If the criteria are not met, the validation rule triggers an error messageand prevents the user from saving the record until the issues are resolved.

a. **Vehicle NumberPlate**: Must followa format (e.g.,MH12AB1234).

To create a validation rule to an Appointment Object

1. Go to the setup page >> click on objectmanager >> From drop down click edit for Appointment object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “ Vehicle ”.
4. Insert the Error Condition Formula as : -

   NOT(REGEX(Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}")).
5. Enter the Error Message as “Please enter vaild number ”, select the Error location as Field and selectthe field as “Vehicle numberplate”, and click Save.

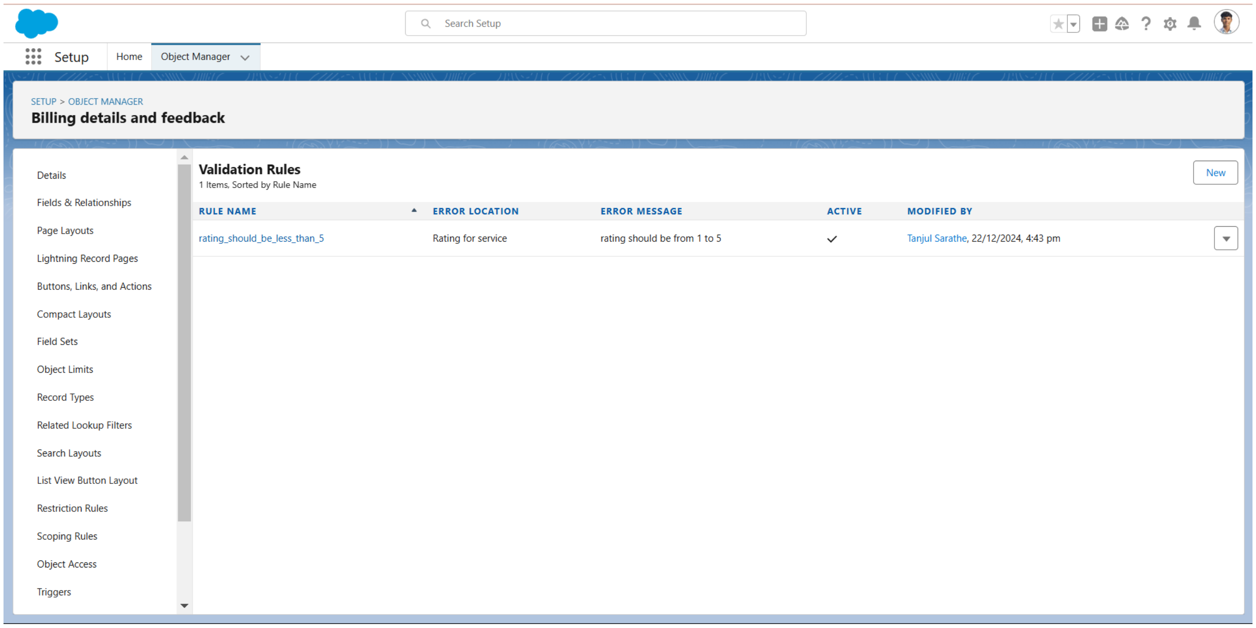


- **Service Status**: Must be set to "Completed" beforerecord can be saved.
- **Rating**: Customerservice rating must be between1 and 5.

To create a validation rule to an Billing details and feedback Object

1. Go to the setup page >> clickon object manager>> From drop down clickedit for Billing details and feedback object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “ rating_should_be_less_than_5”.
4. Insert the Error Condition Formula as : -

   NOT(REGEX(Rating_for_service__c , "[1-5]{1}"))
5. Enter the Error Message as “rating should be from 1 to 5”, select the Error location as Field and select the field as “Ratingfor Service”, and click Save.

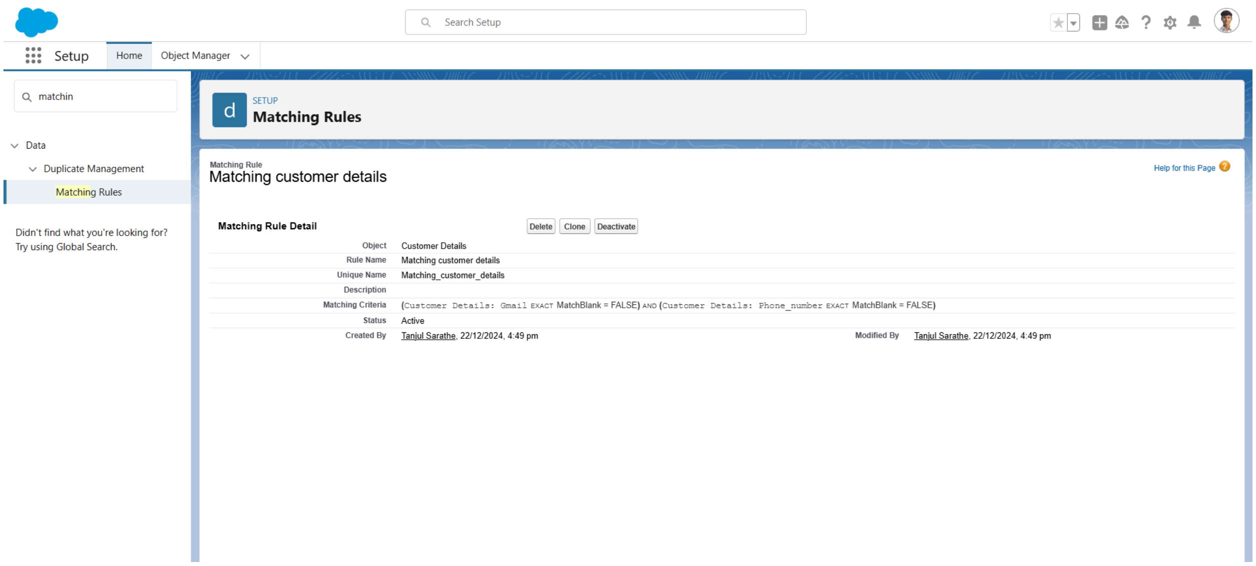


Step 7: Duplicate Rules

- **MatchingRule**: Checks Gmail and phone number to identifyduplicate customer details.

To create a matching rule to an Customer details Object

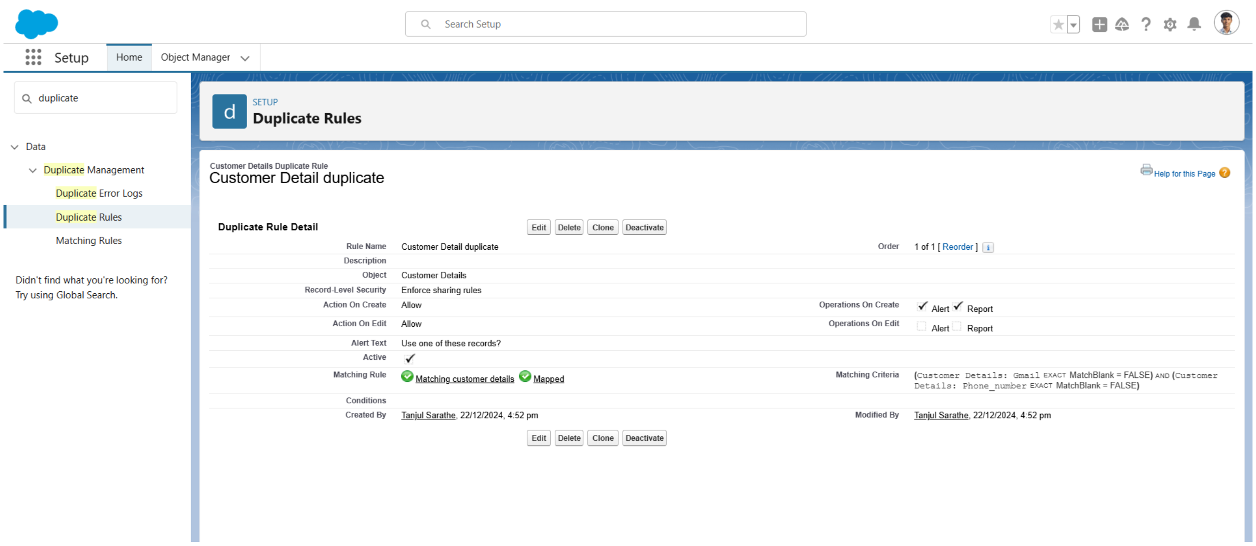
1. Go to quickfind box in setup and search for matching Rule.
2. Click on matching rule >> click on New Rule.
3. Select the objectas Customer detailsand click Next.
4. Give the Rule name : Matching customer details
5. Unique name : isauto populated
6. Define the matchingcriteria as
7. Field Matching Method
   a. Gmail Exact
   b. Phone Number Exact
8. Click save.
9. After Saving Click on Activate.



- **DuplicateRule**: Preventsduplicate customer records.

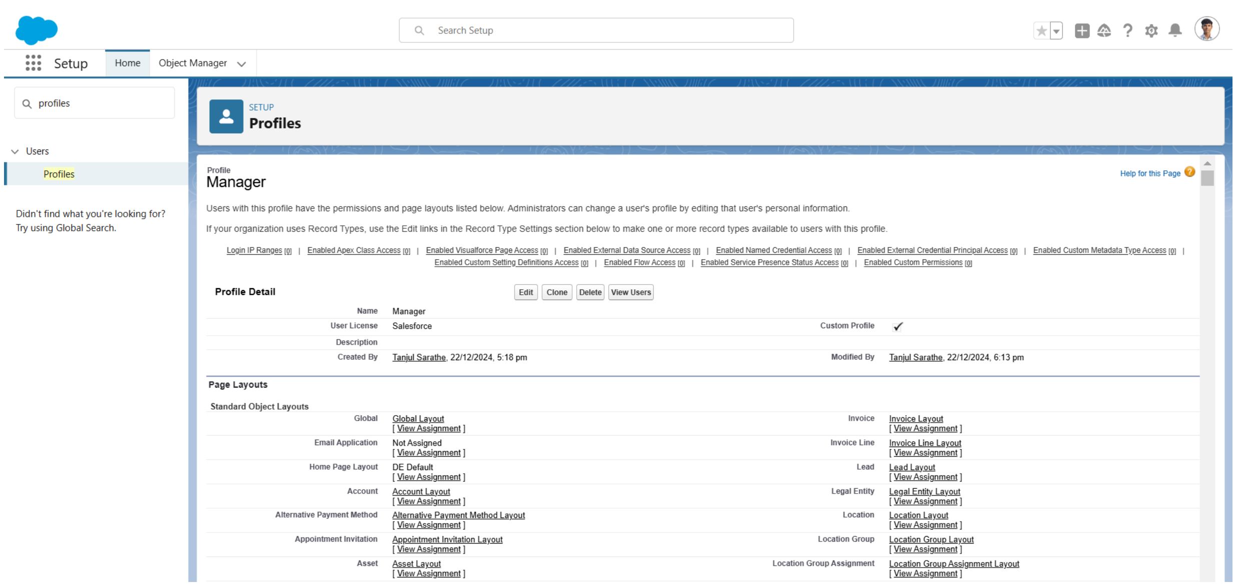
To create a Duplicaterule to an Customer details Object

1. Go to quick find box in setup and searchfor Duplicate rules.
2. Click on Duplicate rule >> click on New Rule >> select customer details object.
3. Give theRulenameas:CustomerDetailduplicate
4. Scroll a little in Matchingrule section
5. Select the matchingrule : Matching customerdetails
6. And Click on save.
7. After saving the Duplicate Rule, Click on Activate.

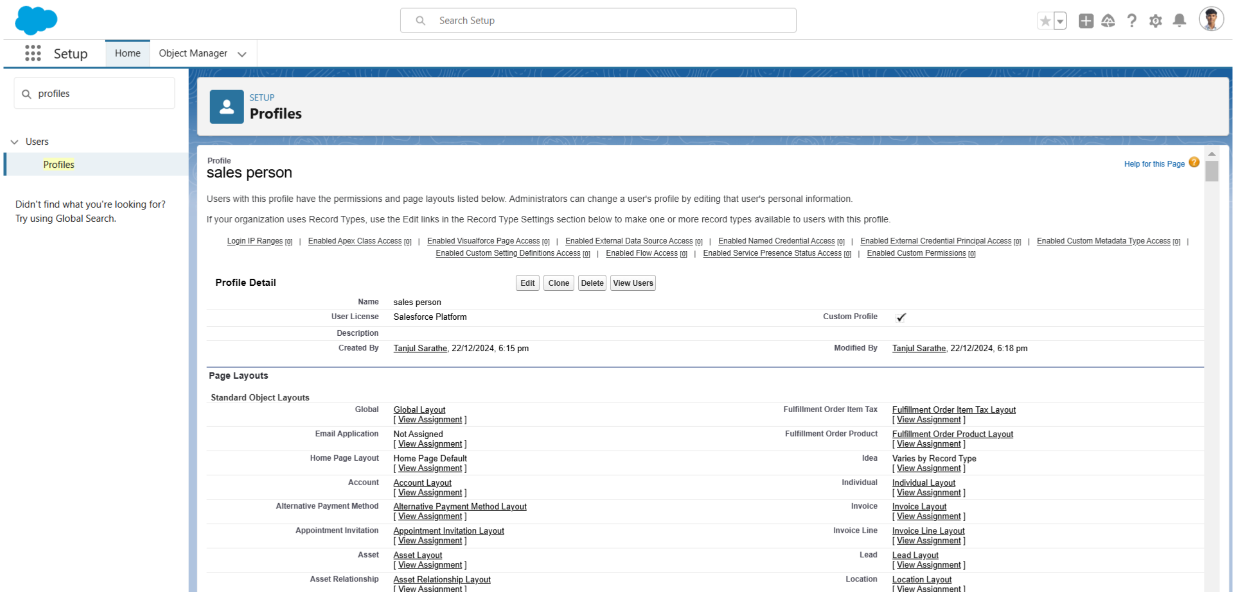


Step 8: Profiles

- **ManagerProfile**: Access to all customobjects with extendedsession time and password settings.



- **Sales Person Profile**: Limited accessto objects with tailored permissions.



Step 9: Roles and Role Hierarchy

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

a. **Manager Role**:Directly under the CEO.

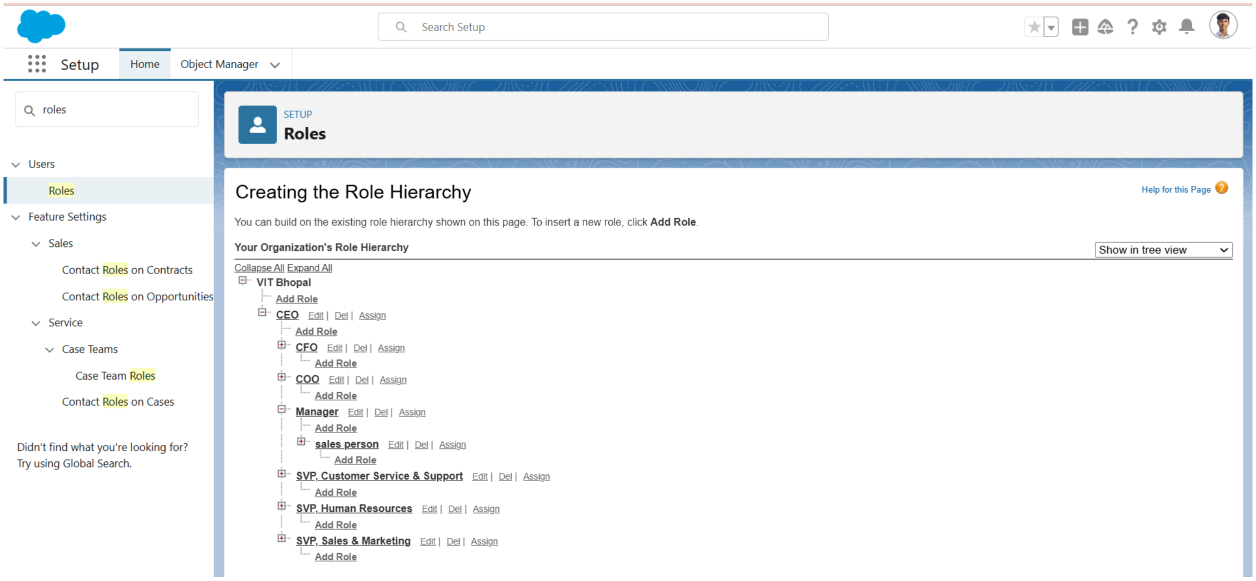
Creating ManagerRole:

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click on ExpandAll and click on add role under whom this role works.
3. Give Label as "Manager" and Role name gets auto populated. Then click on Save.

a. **Sales PersonRole**: Reports to the Managerrole.

Creating anothertwo roles under manager

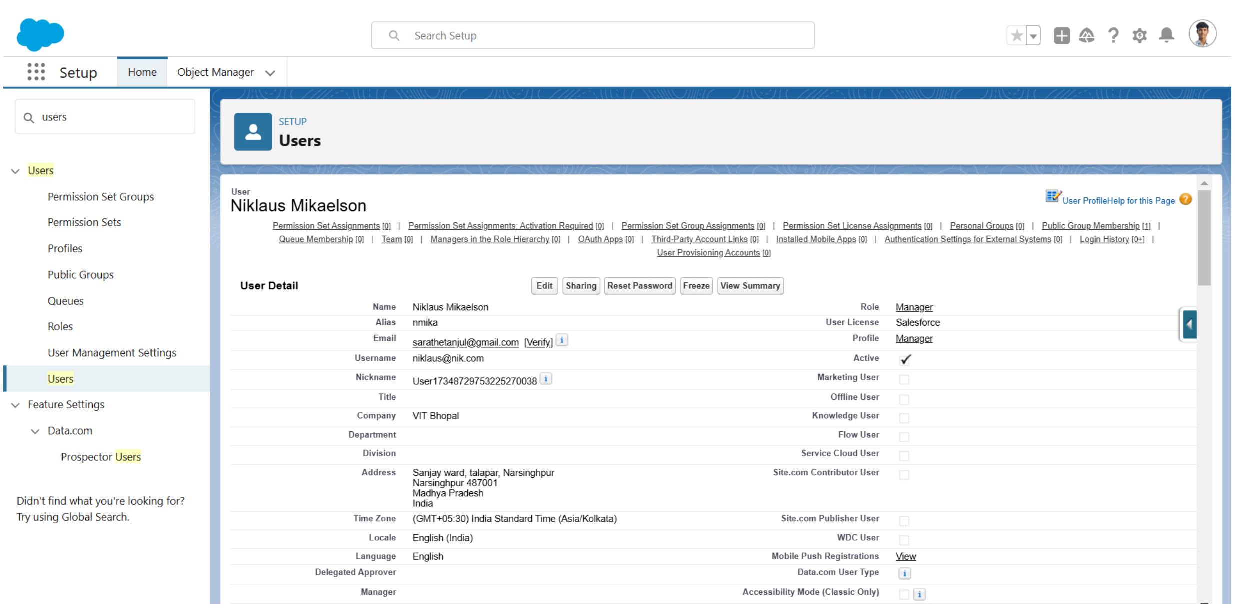
1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click plus on CEO role,and click add role undermanager.
3. Give Label as "sales person" and Role name gets auto populated. Then click on Save.



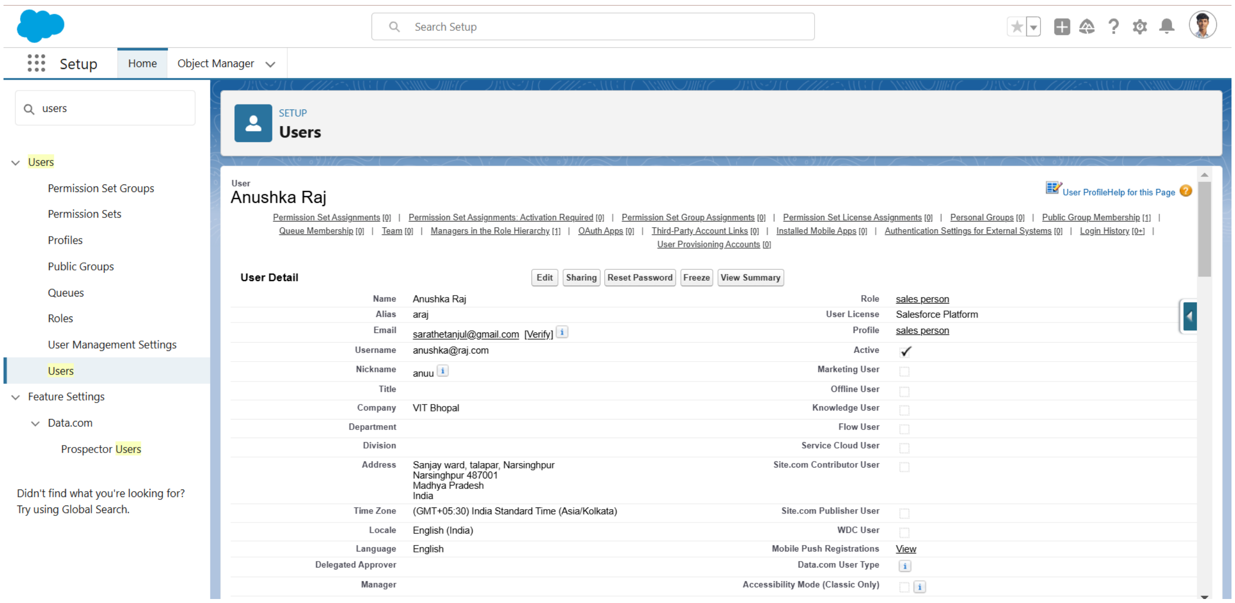
Step 10: Users

A user is anyone who logs in to Salesforce. Users are employeesat your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in
Salesforce has a user account. The user accountidentifies the user, and the user accountsettings determine what features and records the user can access.

a. **ManagerUser**: Role as Manager,Profile as Manager, Salesforce license.



- **Sales Person Users**: Role as SalesPerson, Profile as Sales Person,Salesforce Platform license.



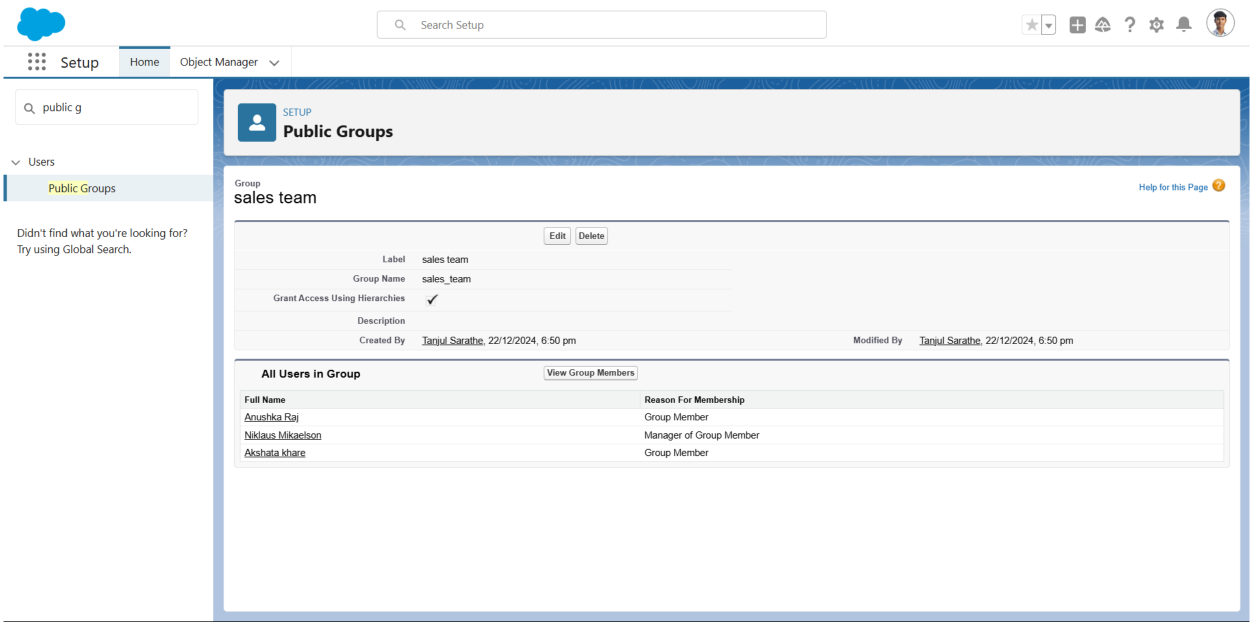
Step 11: Public Groups

Public groups are a valuable tool for Salesforce administrators and developers to streamline user management, data access,and security settings. By creating and using publicgroups effectively, you can maintain a secure and organized Salesforce environment while ensuring that users have
appropriate access to the resources they need.

a. **Sales Team Group**: Containsall Sales Personroles for sharingaccess.

Creating New Public Group

1. Go to setup >> type users in quick find box >> select public groups >> click New.
2. Give theLabel as“sales team”.
3. Group name is autopopulated.
4. Search for Roles.
5. In Available Membersselect Sales personand click on add it will be moved to selected member.
6. Click on save.



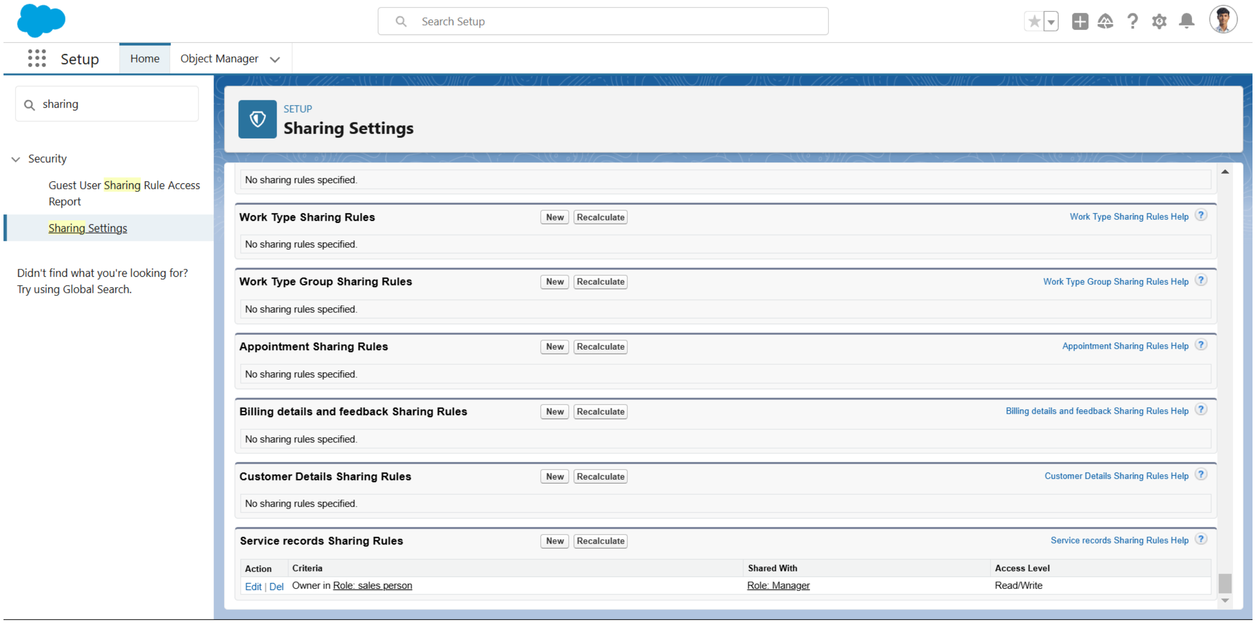
Step 12: Sharing Settings

- **Service RecordsOWD**: Set to Private.
- **Sharing Rule**: Grants read/write access to Manager for Sales Person's service records.

CreatingSharing settings

1. Go to setup >> type users in quick find box >> select Sharing Settings >> click Edit.
2. Change the OWD setting of the Servicerecords Object to private as shown in fig.
3. Click on save and refresh.
4. Scroll down a bit, Clicknew on Service records sharingRules.

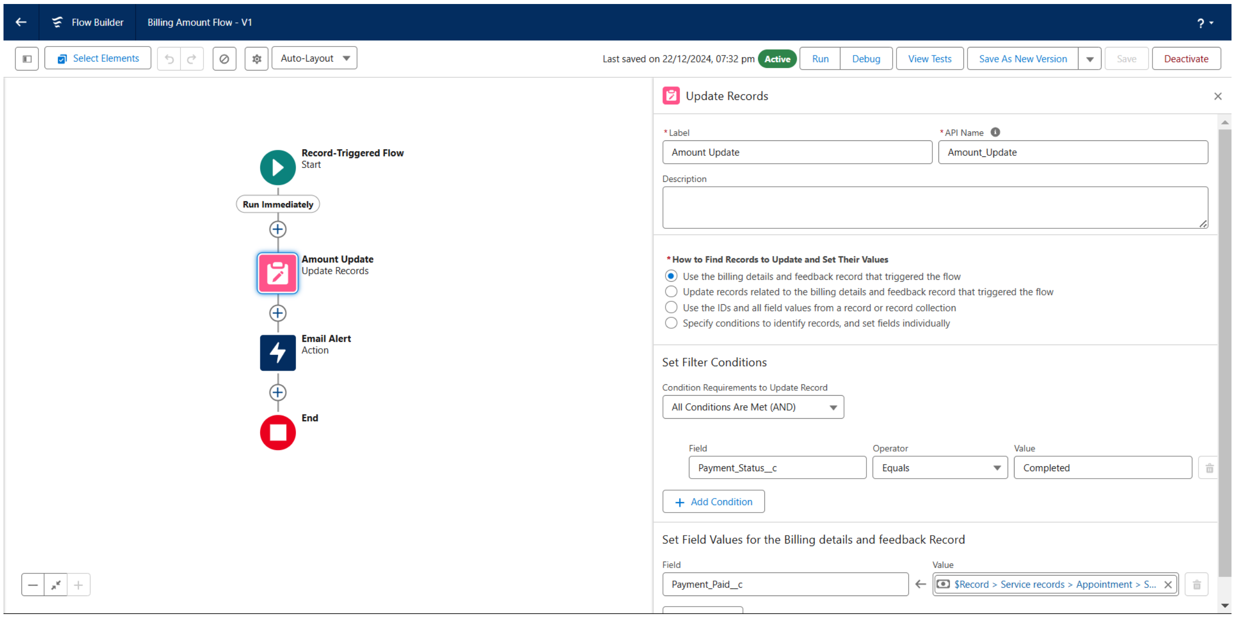
1. Give the Label name as " Sharing setting"
2. Rule name is auto populated.
3. In step 3 : Select which recordsto be shared, members of " Roles" >> " Sales person"
4. In step4: sharewith, select" Roles" >> " Manager "
5. In step 5 : Change the access level to " Read / write ".
6. Click on save.



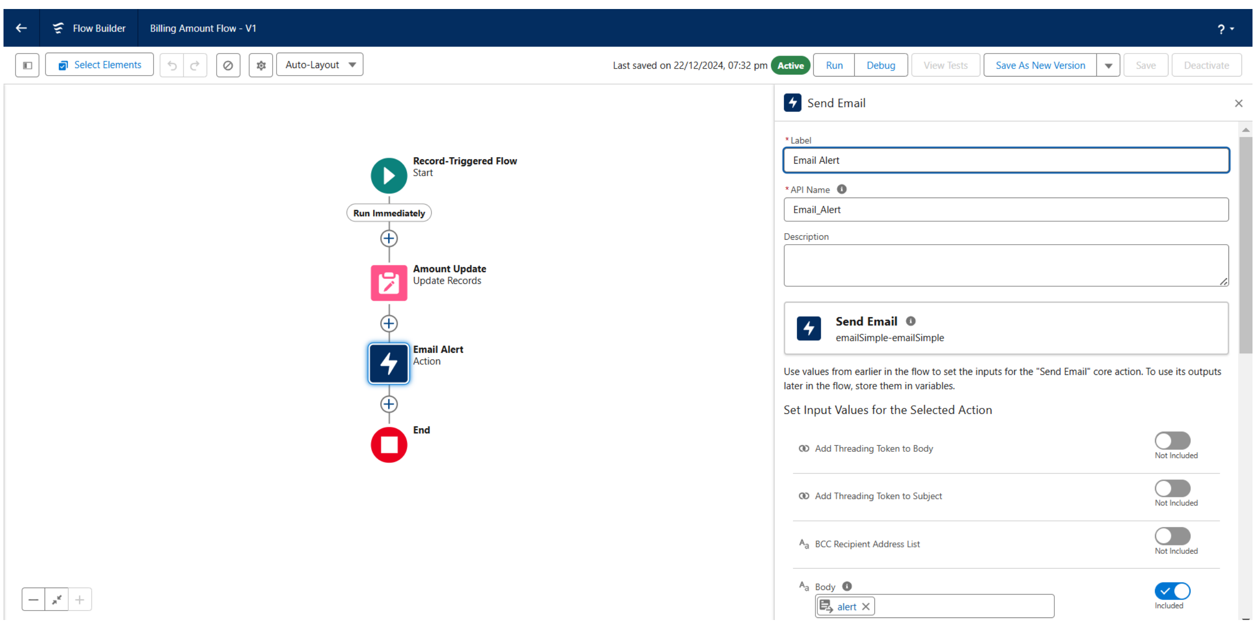
Step 13: Flows

In Salesforce, a flow is a powerfultool that allowsyou to automate business processes, collect and update data, and guide users through a series of screens or steps. Flows are built using a visual interface and can be created withoutany coding knowledge.

a. **Record-Triggered Flow**: Automatesupdates and emailalerts for billingcompletion.
b. Under the Record-triggered Flow Click on “+” Symboland In the Drop down List selectthe “Update records Element”



Email Alert: Sends a "Thank you" message when payments are completed.



- **Another Flow - Update Service Status**

Step 14: Apex Triggers

Apex can be invokedby using triggers.Apex triggers enable you to perform custom actions before or after changes to Salesforce records, such as insertions, updates, or deletions.

- **HandlerClass (AmountDistributionHandler)**: Calculates service amounts basedon selected services.

Code:

"AccountDistributionHandler.apxc"

```
public class AmountDistributionHandler {
        public staticvoid 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(AmountDistribution)**: Runs on appointment insertor update to update service amounts.

  Code:

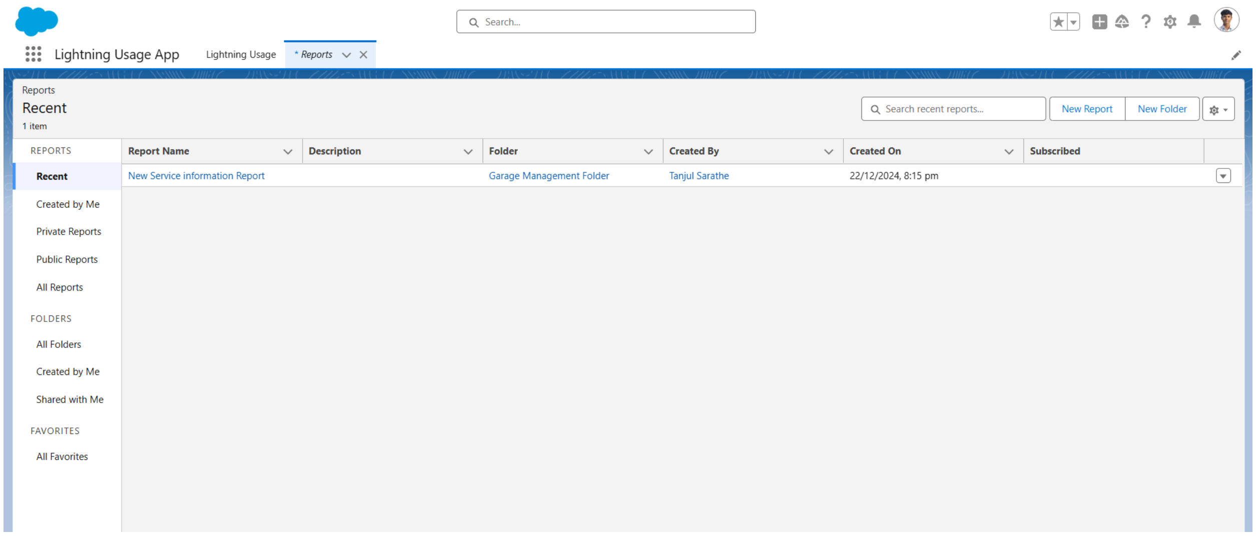
  "**AmountDistribution.apxt**"

```
trigger AmountDistribution on Appointment__c (before
insert,beforeupdate) {
        if(trigger.isbefore && trigger.isinsert ||
trigger.isupdate){
                AmountDistributionHandler.amountDist(trigger.new);
        }
}
```

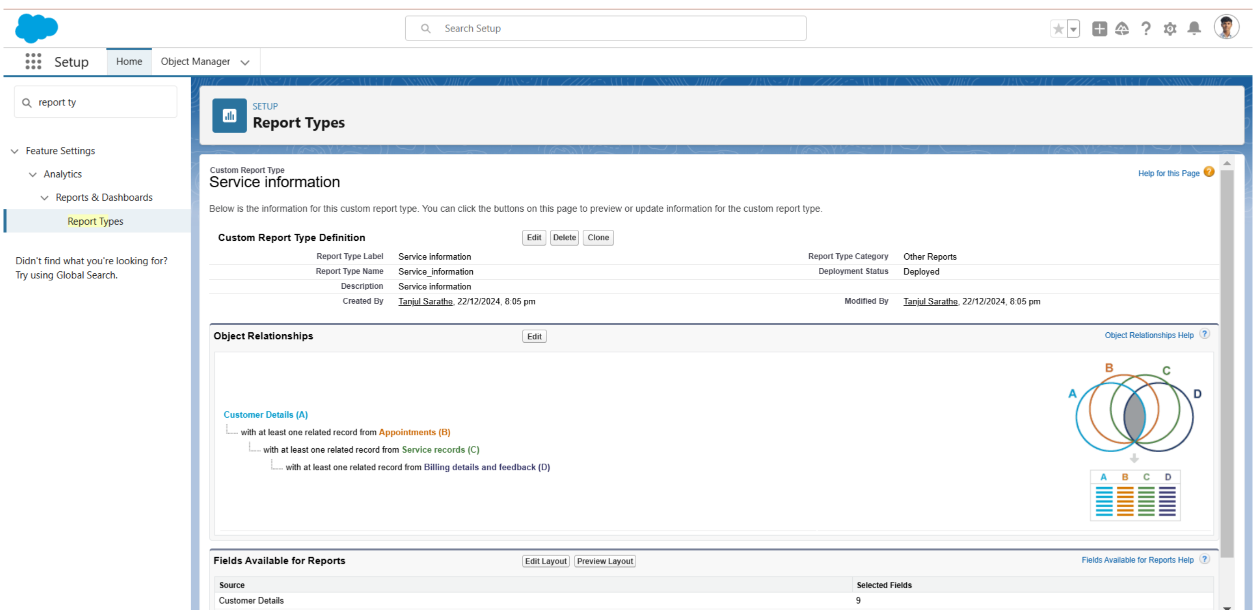
Step 15: Reports

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

- **Report Folder**: Organize all reports under "Garage Management Folder".



- **Custom Report Type**: Combines Customer Details,Appointments, Service Records,and Billing.



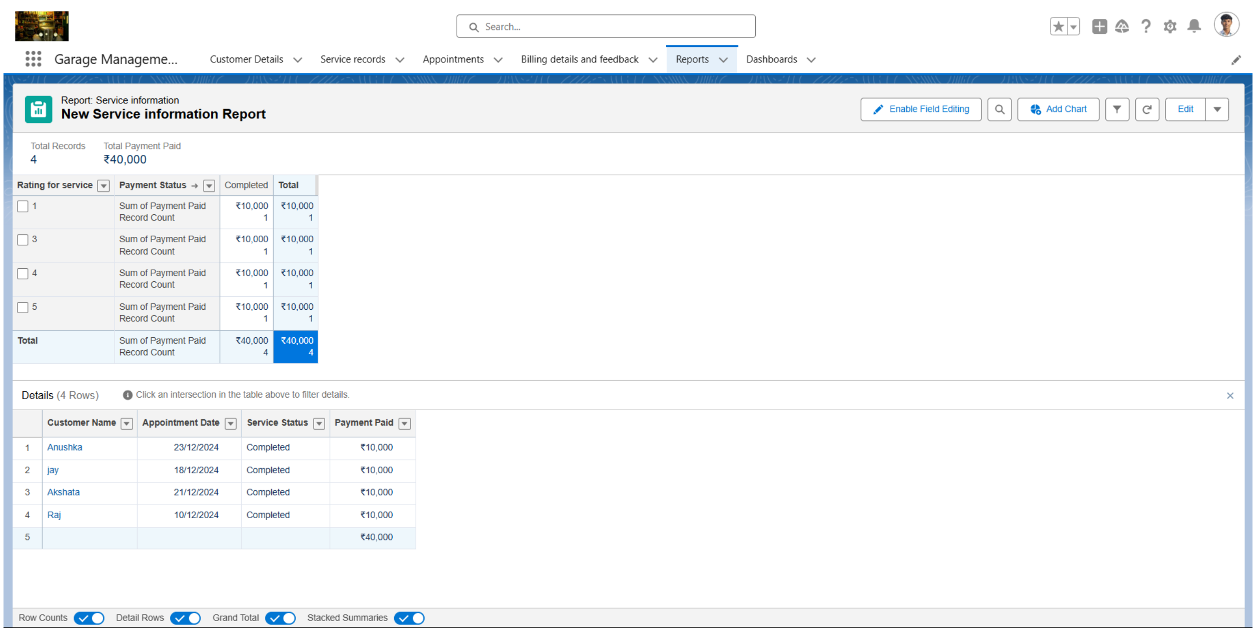
- **Reports**: Custom report "Service Information Report" with fields for customer, appointment date, service status, and payments.

CreateReport

1. Go to the app >>click on the reports tab
2. Click New Report.

3. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.

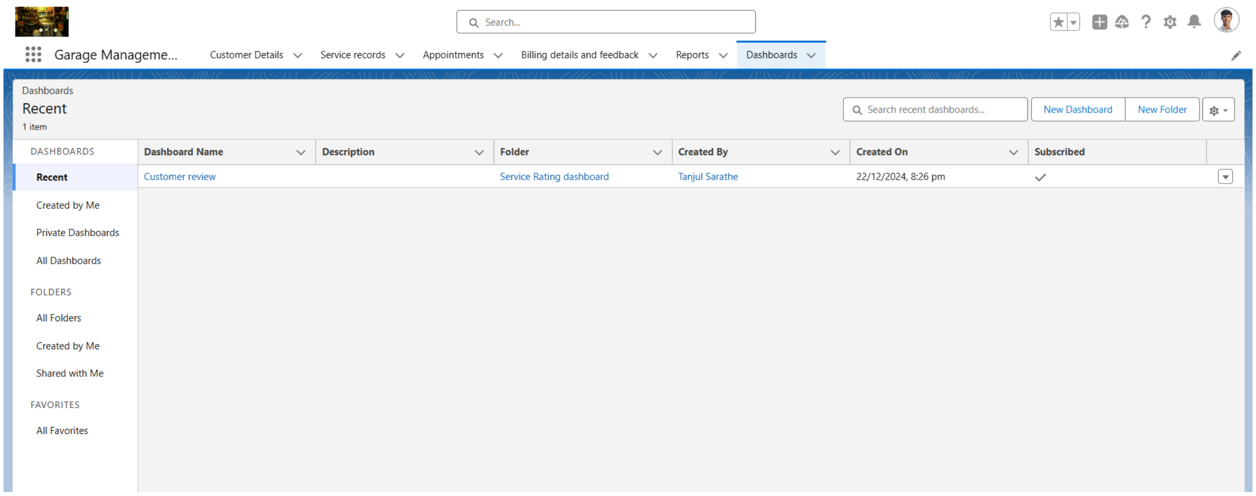
4. Their outline pane is openedalredy, select the fields that mentioned below in column section.
    a. Customer name
    b. Appointment Date
    c. Service Status
    d. Payment paid
5. Remove the unnecessary fields.
6. Select the fieldsthat mentioned belowin GROUP ROWS section.
    i. Rating for Service
7. Select the fieldsthat mentioned belowin GROUP ROWS section.
    i. Payment Status
8. Click on Add Chart , Select the Line Chart.
9. Click on save, Give the report Name : New Service information Report
10. Report unique Name is auto populated.
11. Select the folder the created and Click on save.



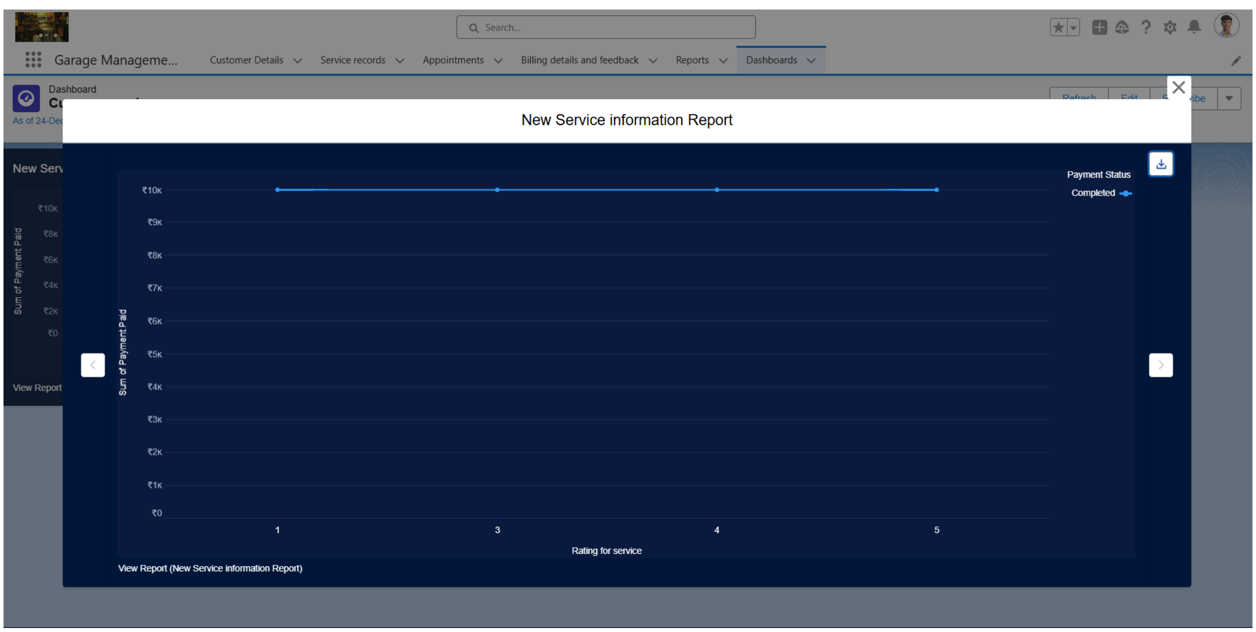
Step 16: Dashboards

Dashboards help you visually understand changing business conditions so you can make decisions basedon the real-time data you've gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading,and sharing dashboards, review these dashboard basics.

- **DashboardFolder**: "Service Rating Dashboard" for organizing dashboards.



- **Dashboard Components**: Visualizes service ratings, payment statuses, and operational KPIs.

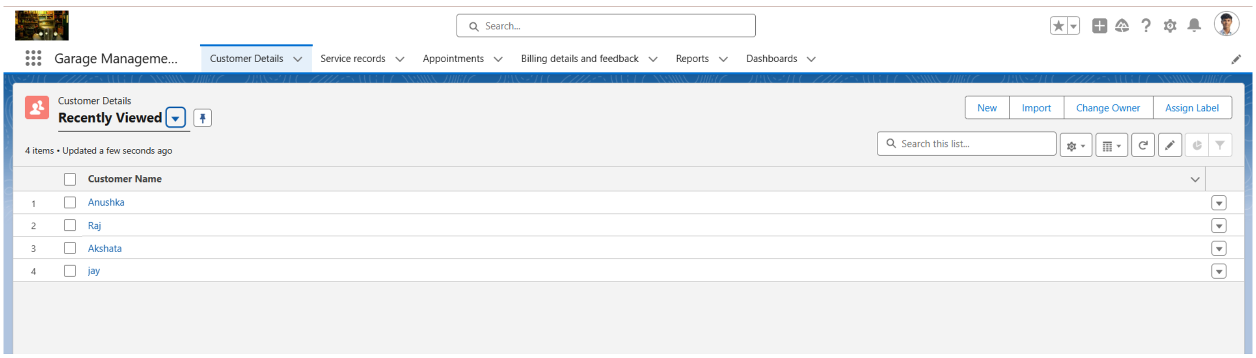


a. Also Subscribe added:-Set the Frequency as " weekly ". >>Set a day as monday and then we saved it.

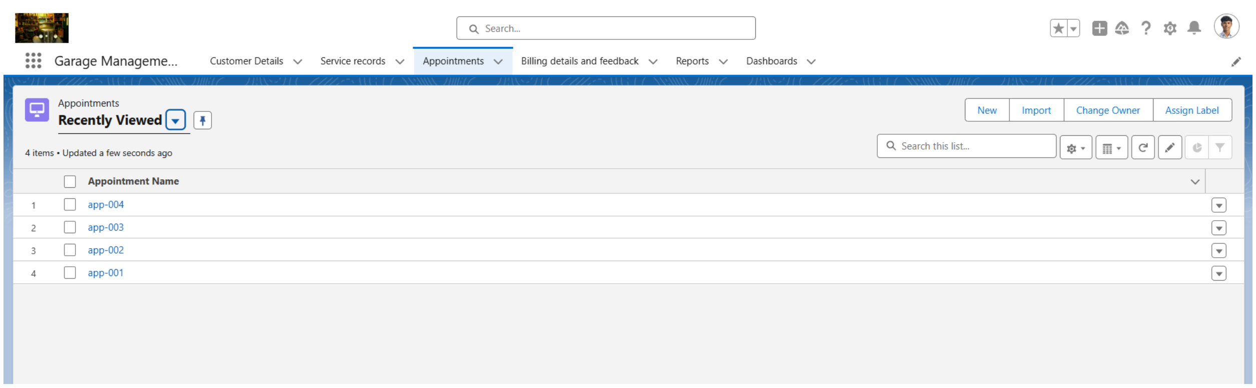
User Adoption

In our GarageManagement System we created recordsfor all objects.

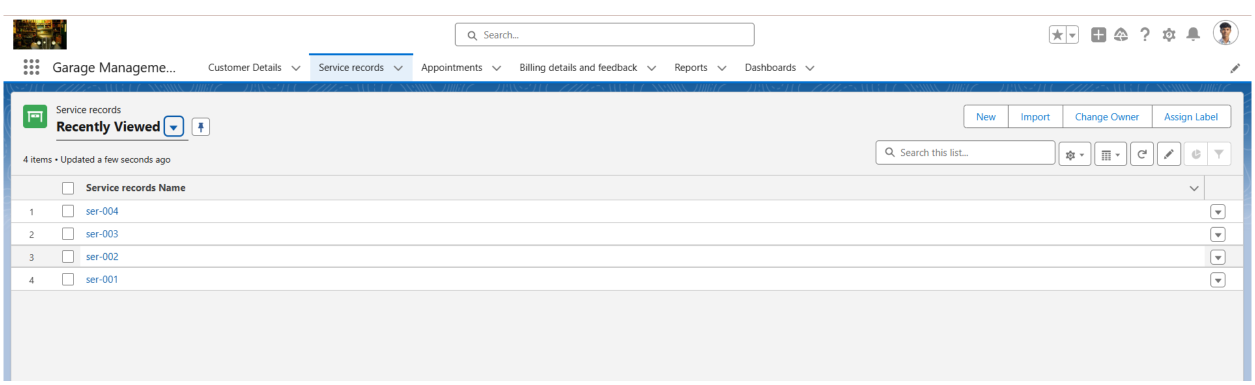
Customer Details



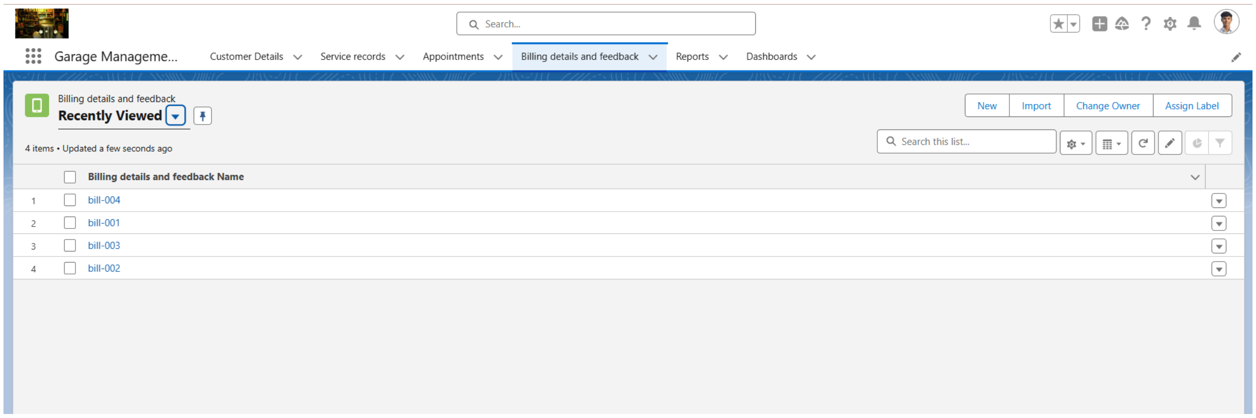
Appointments



Service Records



Billing Details and feedback



1. Testing and Validation

Unit Testing

i. Comprehensivetesting of individual components, including Apex classes and triggers, to ensure code accuracy and functionality.

ii. Each module will be testedin isolation to detect and fix issues at an early stage.

User Interface Testing

i. Ensure the user interface is user-friendly, responsive, and works on different devices (desktop, tablet, mobile).

ii. Validate the functionality of buttons, links, forms, and navigation.

2. Key Scenarios Addressed by Salesforce in the Implementation Project

This sectionoutlines the key scenarios that Salesforce addressesin the implementation of the Garage Management System:

1. **Service Appointment Booking**: Customers can book appointments via a self-service portal.
2. **Customer Communication**: Automated notifications and reminders are sent to customers.
3. **InventoryManagement**: Salesforcetracks spare parts and sends reordering alerts.

4. **Job Tracking**: Service advisors and technicians can track the status of jobs and update customers in real time.
5. **Billing and Payments**: Generateinvoices and processpayments through an integrated payment gateway.

6. **Customer FeedbackCollection**: Customers providefeedback after servicecompletion, which is used to improve service quality.

Conclusion

The GarageManagement System builton Salesforce providesa comprehensive platformfor managing appointments, services,and billing. Throughthe use of custom objects,tabs, profiles, flows, validation rules, and Apex triggers, the system ensuressmooth operations, efficient record management, and enhancedcustomer satisfaction. This project enablesbetter tracking of garage operations, fosters data-driven decision-making, and supports the long-term growth of garage businesses.

Thank You