Garage Management System

by

Devendra Kumar Sahu
devendrasahu3837@gmail.com
College: VIT Bhopal

Index Page

1. Project Overview

This project is focused on developing a Garage Management System using Salesforce. The primary objective is to streamline garage operations, enhance service efficiency, and improve customer experience. The solution aims to digitize and automate tasks related to appointment scheduling, service tracking, billing, and customer feedback. By leveraging Salesforce's features, the project intends to boost operational efficiency, ensure accurate data management, and support the long-term goals of garage businesses.

2. Objectives

Business Goals:

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

Specific Outcomes:

- Implement a digital solution for managing customer details, appointments, service records, and billing.
- Provide real-time insights through reports and dashboards.
- Automate workflows to reduce manual effort and errors.

3. Salesforce Key Features and Concepts Utilized

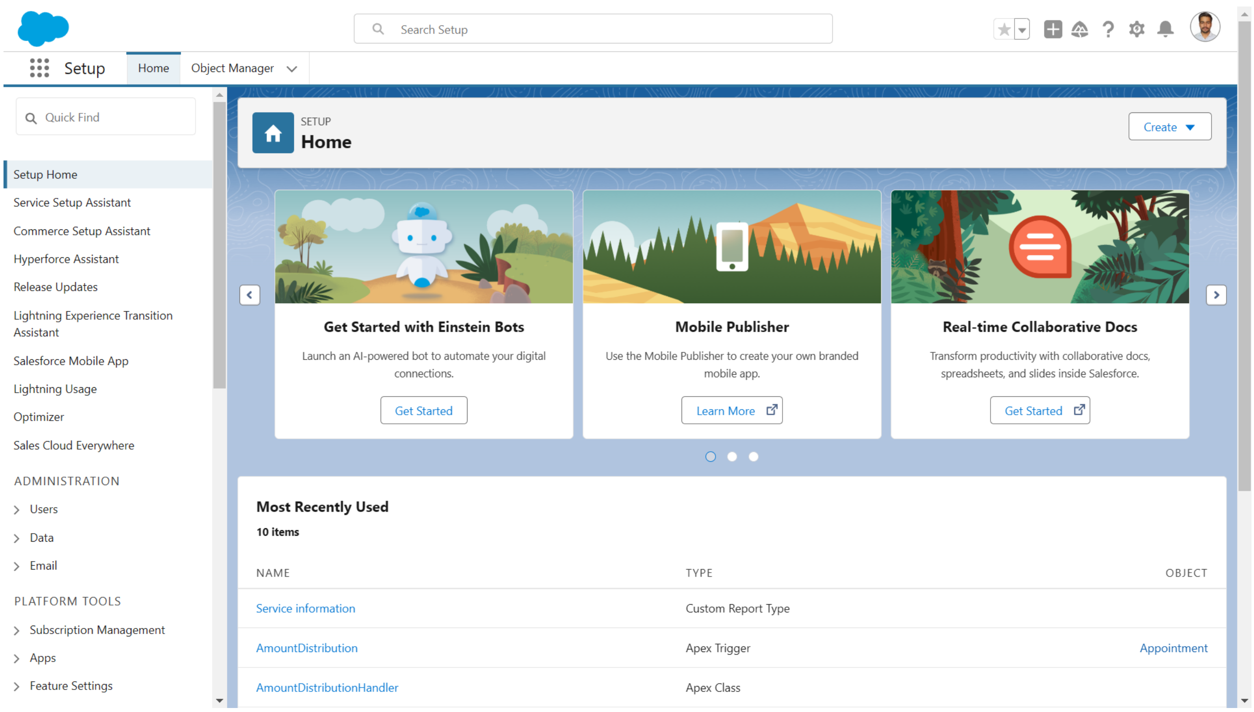
- **Custom Objects**: Customer Details, Appointments, Service Records, Billing Details & Feedback.
- **Custom Tabs**: User-friendly navigation for accessing key objects.
- **Lightning App**: Centralized access to essential tools for the garage management process.
- **Custom Fields**: Multiple field types like lookup, picklist, checkbox, and formula fields to ensure robust data capture.
- **Validation Rules**: Enforced data integrity and error prevention.
- **Profiles & Roles**: Role-based access control for managers and salespersons.
- **Public Groups**: Simplified team-based access to records.
- **Flows**: Automation of record updates and email alerts.
- **Apex Triggers**: Automated calculation of service amounts.
- **Reports & Dashboards**: Data visualization and performance tracking.

4. Detailed Steps to Solution Design

Step 1: Creating the Developer Account

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

Your Setup page would look like this:



Step 2: Creating Custom Objects

- To Navigate to Setup page: Click on gear icon ? click setup.

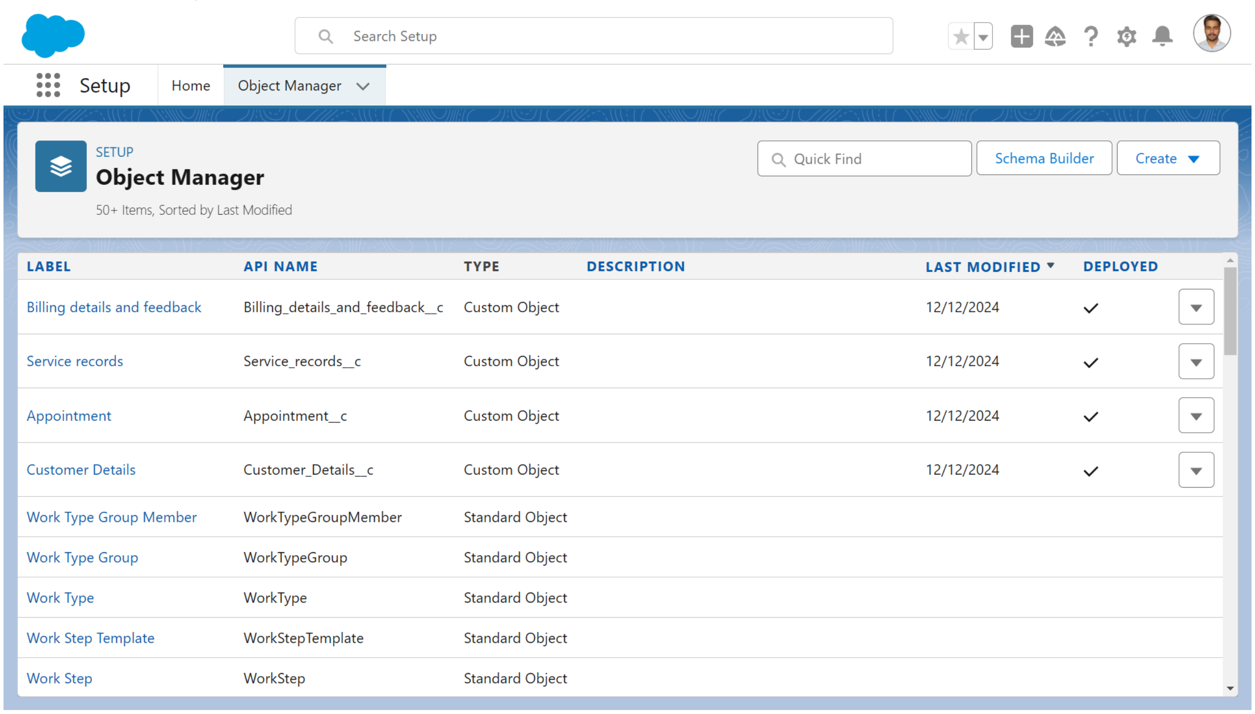
- To create an object: From the setup page > Click on Object Manager > Click on Create > Click on Custom Object.

- On Custom object defining page: Enter the label name, plural label name, click on Allow reports, Allow search.

- Click on Save.

For our project we need to create four objects

- **Customer Details**: Text-based customer 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.
- **Billing Details & Feedback**: Auto-numbered billing records with fields for payment tracking and feedback.

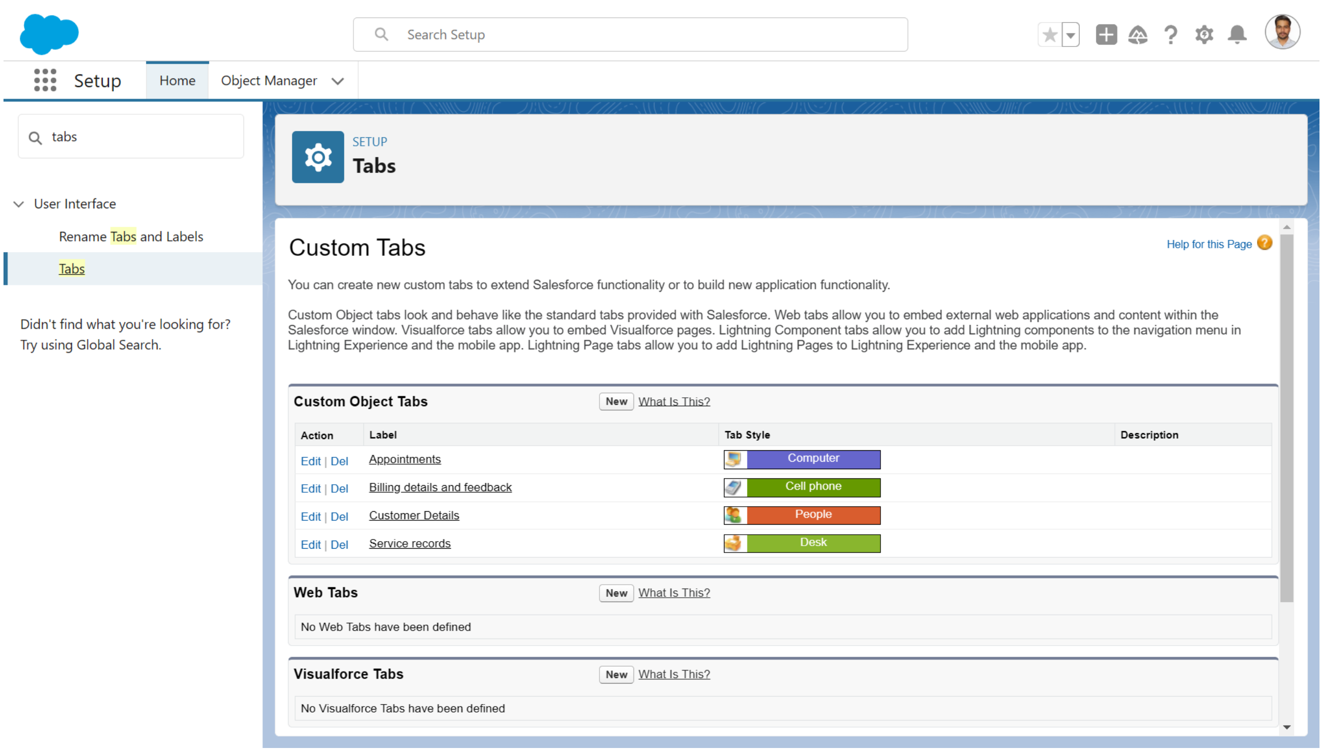


Step 3: Custom Tabs

- Custom tabs for **Customer Details**, **Appointments**, **Service Records**, and **Billing Details & Feedback**.
- Tabs are displayed for easy navigation and user access.

I followed these 4 steps to create all required custom 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 Custom App) uncheck the include tab .
- (iii) Make sure that the Append tab to users' existing personal customizations is checked.
- (iv) Click save.



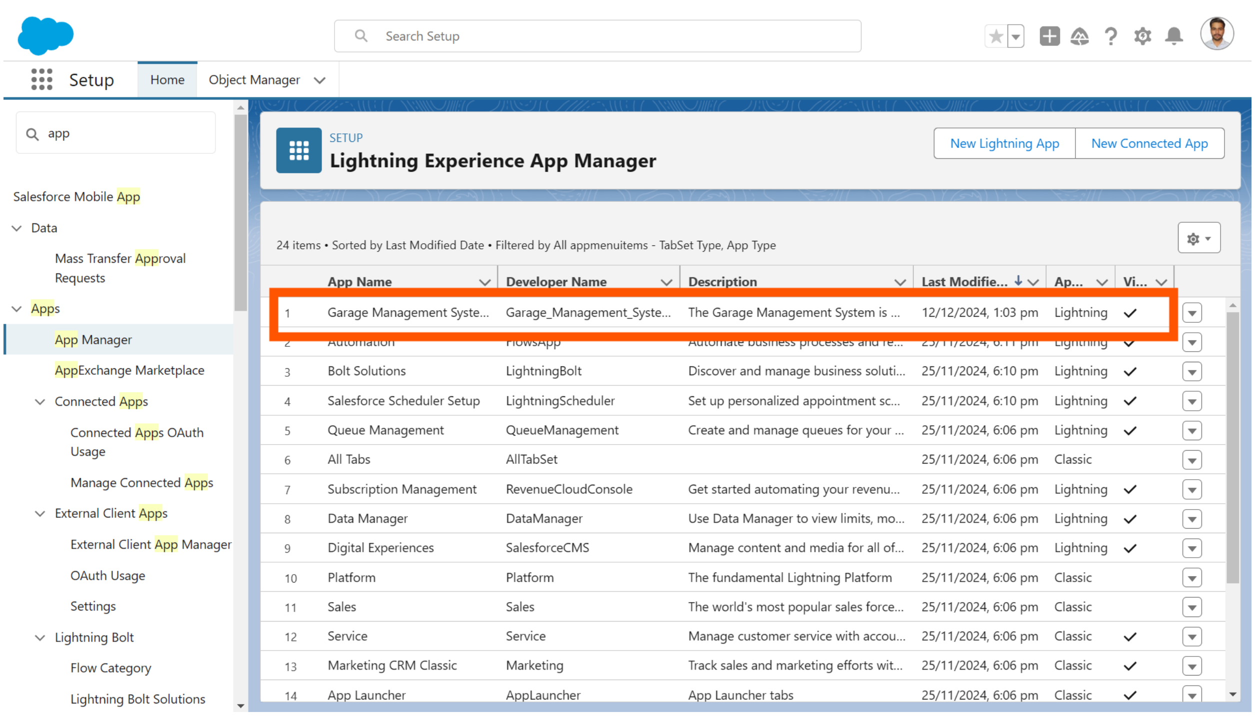
Step 4: Building the Lightning App

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 lightning app page:

1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.

2. 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.
3. To Add Navigation Items
4. 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.
5. To Add User Profiles: Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.



Step 5: Creating Custom Fields

- **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:

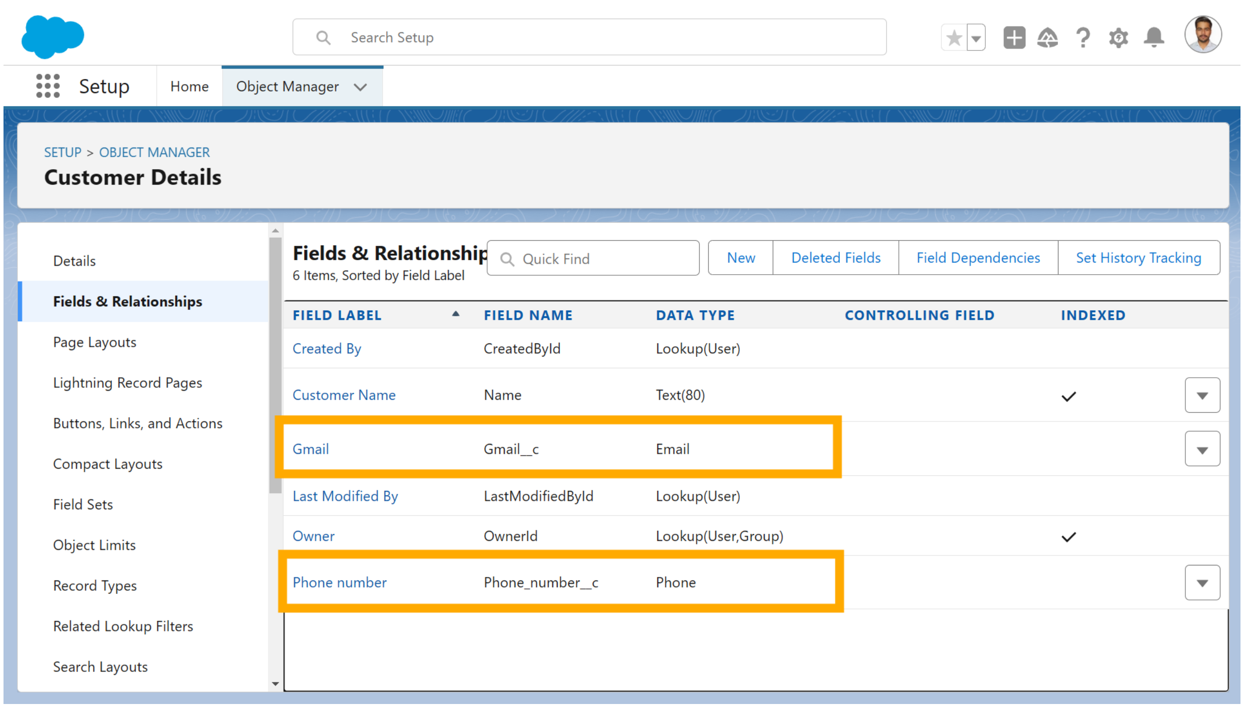
    - Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.

    - Now click on "Fields & Relationships" >> New
    - Select Data type as a "Email" and Click on Next

    - Fill the Above as following:

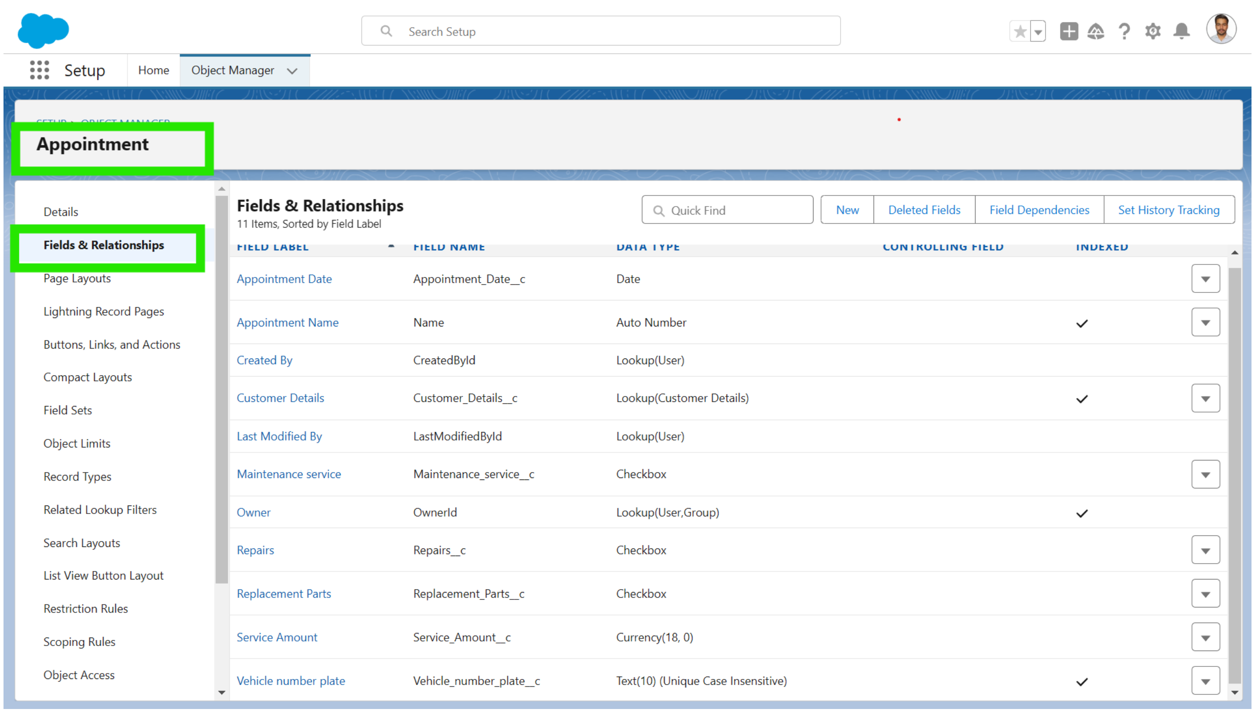
        Field Label : Gmail
        Field Name : gets auto generated

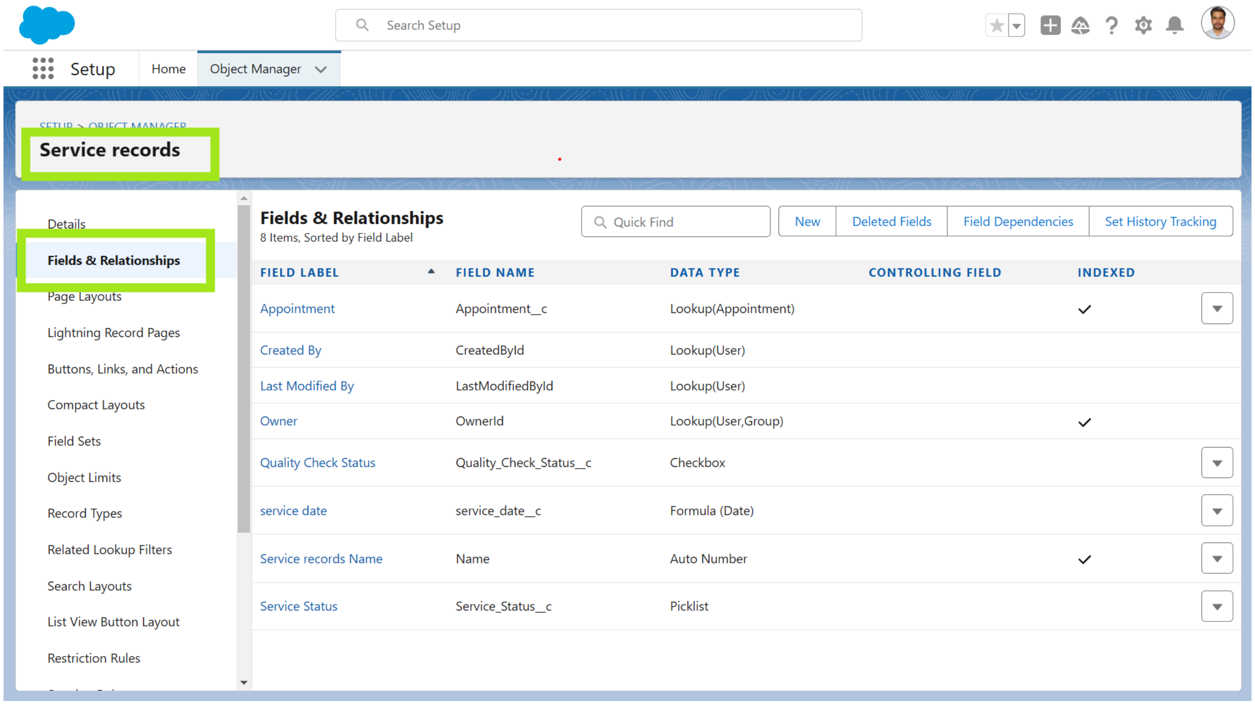
    - Click on Next >> Next >> Save and new.

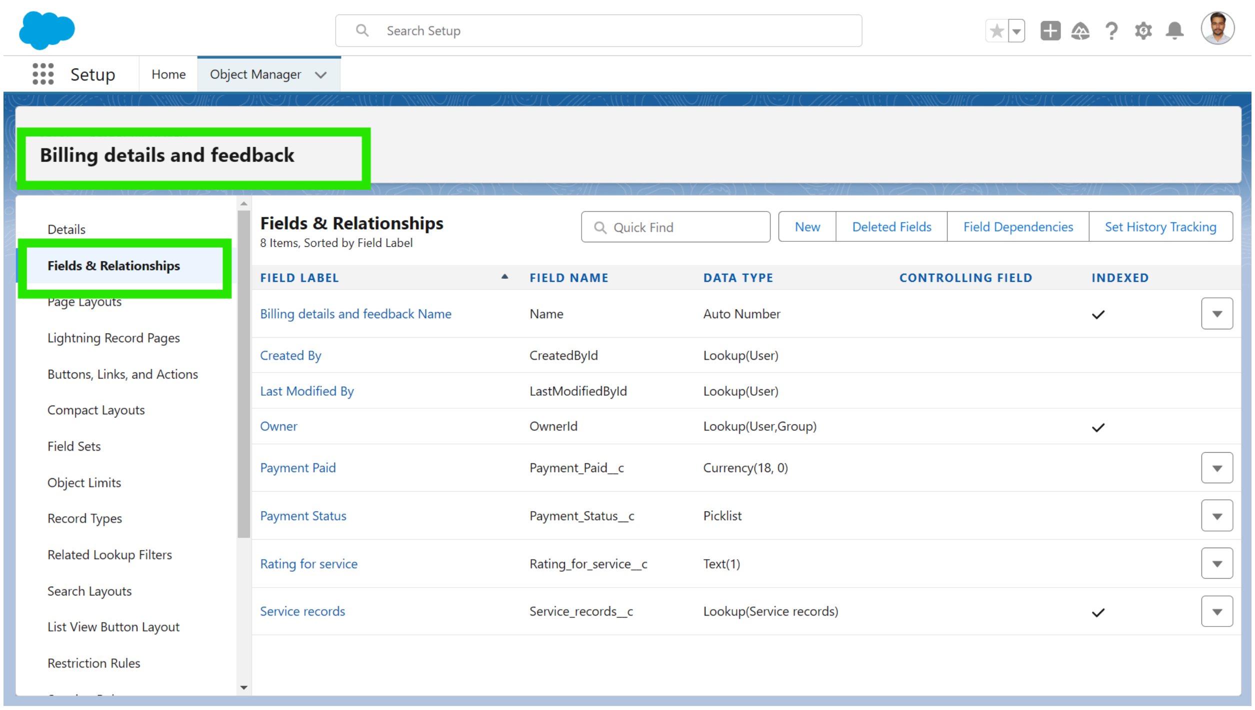


- **Lookup Fields**: Link Appointments to Customers, Service Records to Appointments, and Billing Details to Service Records.
- **Picklist Fields**: Service status (Started, Completed) and payment status (Pending, Completed).
- **Formula Fields**: Calculate service date from created date.
- **Text Fields**: Vehicle number plate (10 characters, unique) and customer feedback rating (1 character).

Appointments - all fields & relationship:




Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Service records
Details
Fields & Relationships
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
Fields & Relationships
8 Items, Sorted by Field Label
Quick Find
New
Deleted Fields
Field Dependencies
Set History Tracking
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


Search Setup
Setup
Home
Object Manager
Billing details and feedback
Details
Fields & Relationships
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
Fields & Relationships
8 Items, Sorted by Field Label
Quick Find
New
Deleted Fields
Field Dependencies
Set History Tracking
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) | | ✓

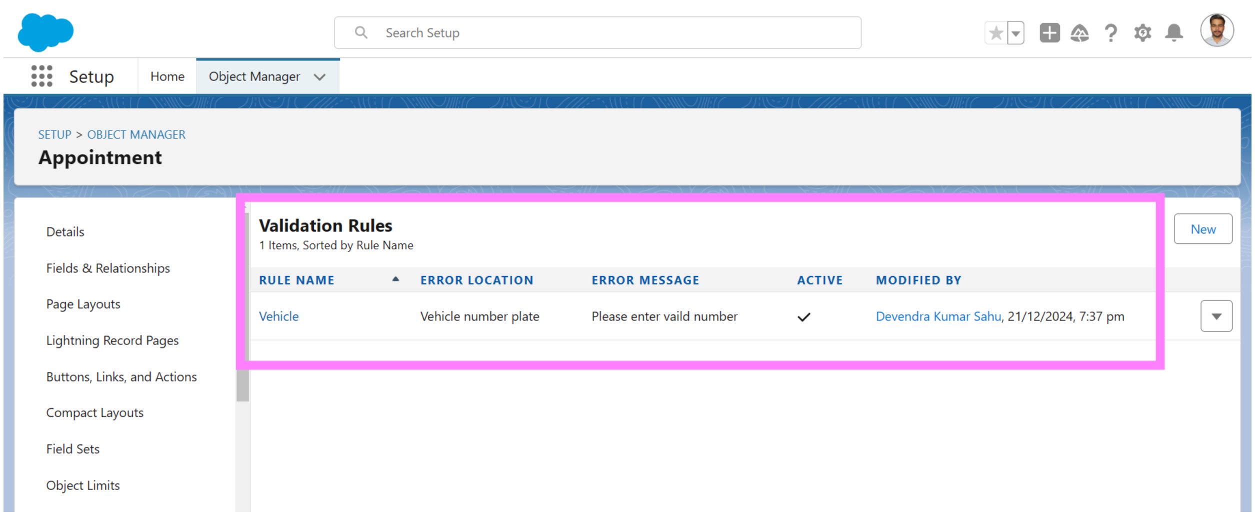
Step 6: Validation Rules

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 message and prevents the user from saving the record until the issues are resolved.

- **Vehicle Number Plate**: Must follow a format (e.g., MH12AB1234).

To create a validation rule to an Appointment Object

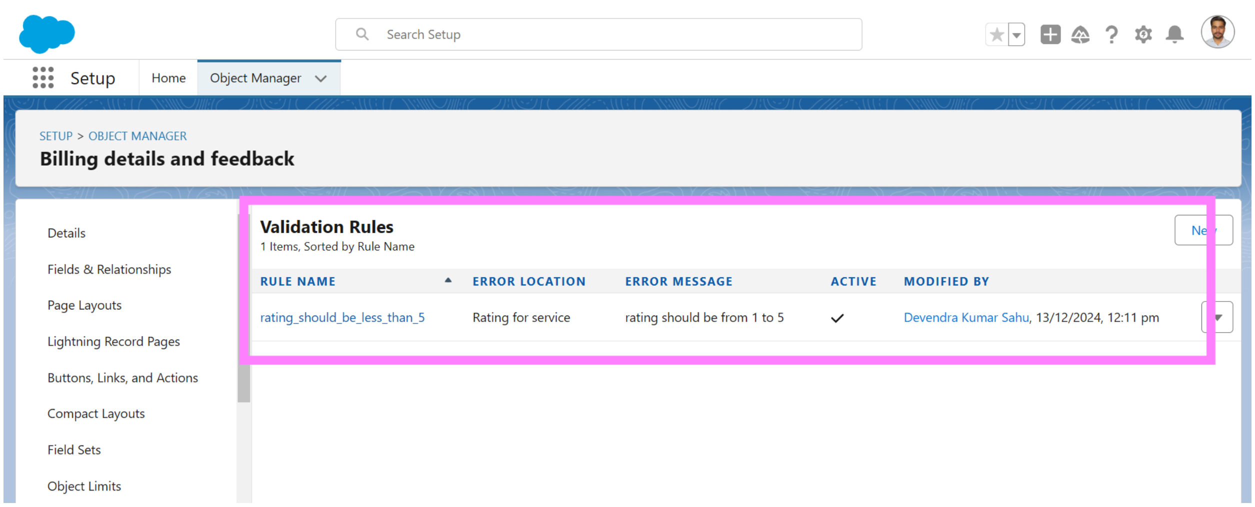
1. Go to the setup page >> click on object manager >> From drop down click edit for Appointment object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as " Vehicle ".
4. Insert the Error Condition Formula as : -
   NOT(REGEX(Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))
5. Enter the Error Message as "Please enter vaild number ", select the Error location as Field and select the field as "Vehicle number plate", and click Save.



- **Service Status**: Must be set to "Completed" before record can be saved.
- **Rating**: Customer service rating must be between 1 and 5.

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

1. Go to the setup page >> click on object manager >> From drop down click edit 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 "Rating for Service", and click Save.

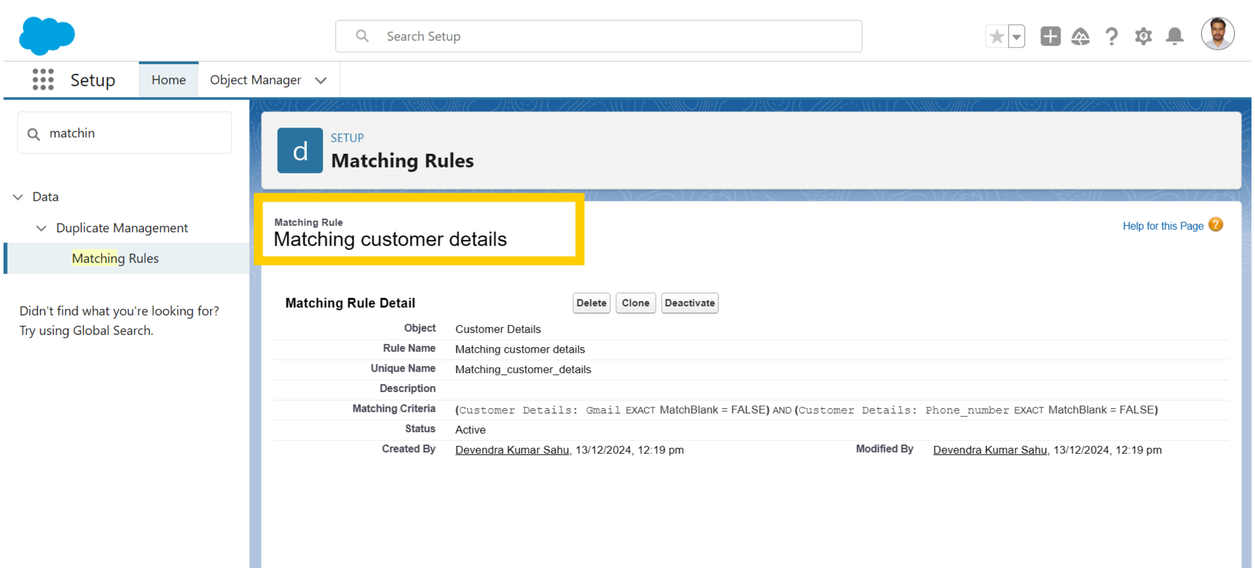


Step 7: Duplicate Rules

- **Matching Rule**: Checks Gmail and phone number to identify duplicate customer details.

To create a matching rule to an Customer details Object

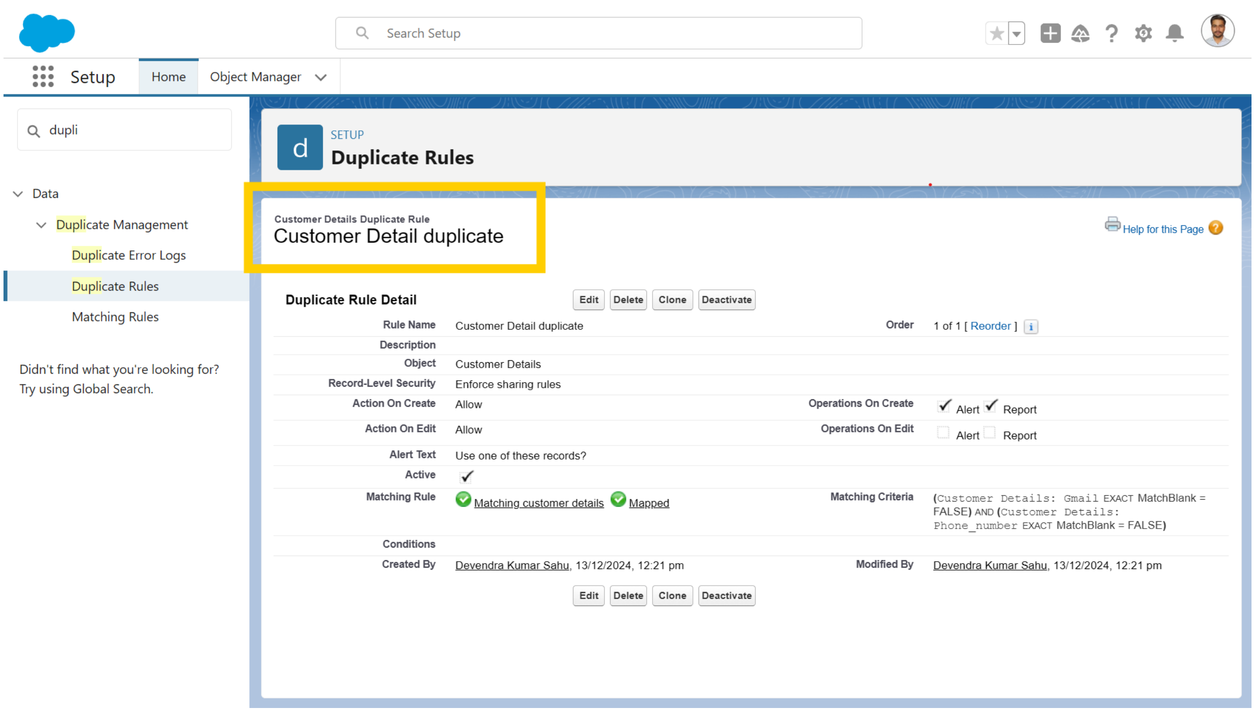
1. Go to quick find box in setup and search for matching Rule.
2. Click on matching rule >> click on New Rule.
3. Select the object as Customer details and click Next.
4. Give the Rule name : Matching customer details
5. Unique name : is auto populated
6. Define the matching criteria as
7. Field Matching Method
   1. Gmail Exact
   2. Phone Number Exact
8. Click save.
9. After Saving Click on Activate.



- **Duplicate Rule**: Prevents duplicate customer records.

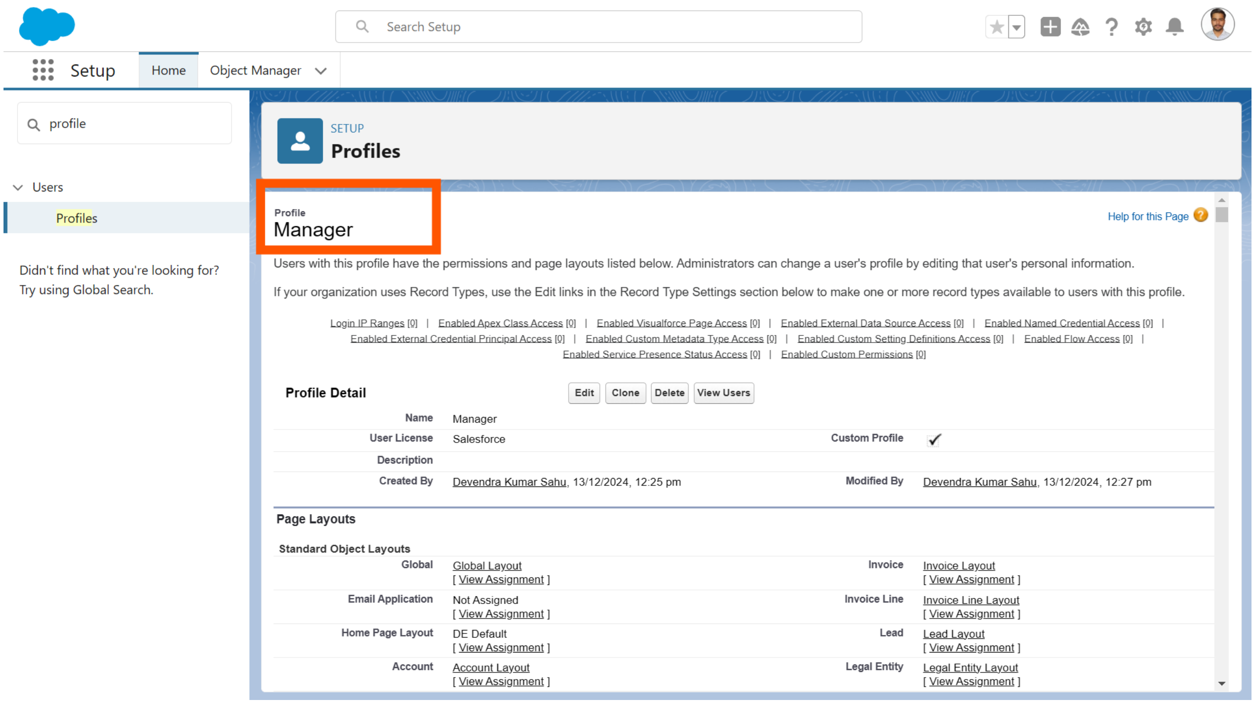
To create a Duplicate rule to an Customer details Object

1. Go to quick find box in setup and search for Duplicate rules.
2. Click on Duplicate rule >> click on New Rule >> select customer details object.
3. Give the Rule name as : Customer Detail duplicate
4. Scroll a little in Matching rule section
5. Select the matching rule : Matching customer details
6. And Click on save.
7. After saving the Duplicate Rule, Click on Activate.

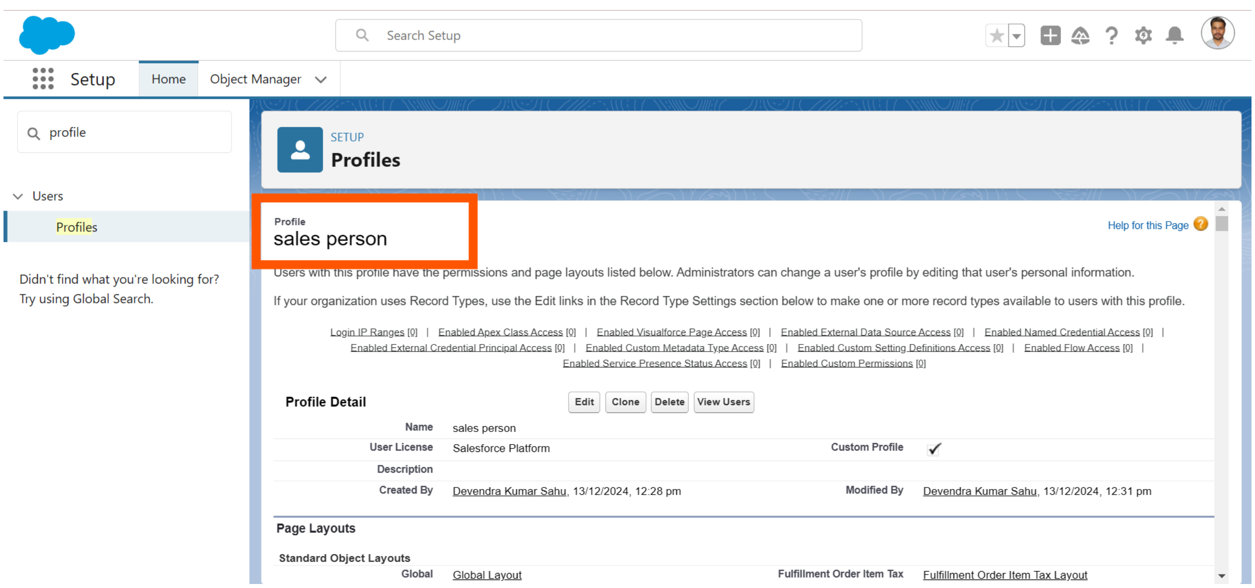


Step 8: Profiles

- **Manager Profile**: Access to all custom objects with extended session time and password settings.



- **Sales Person Profile**: Limited access to 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.

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

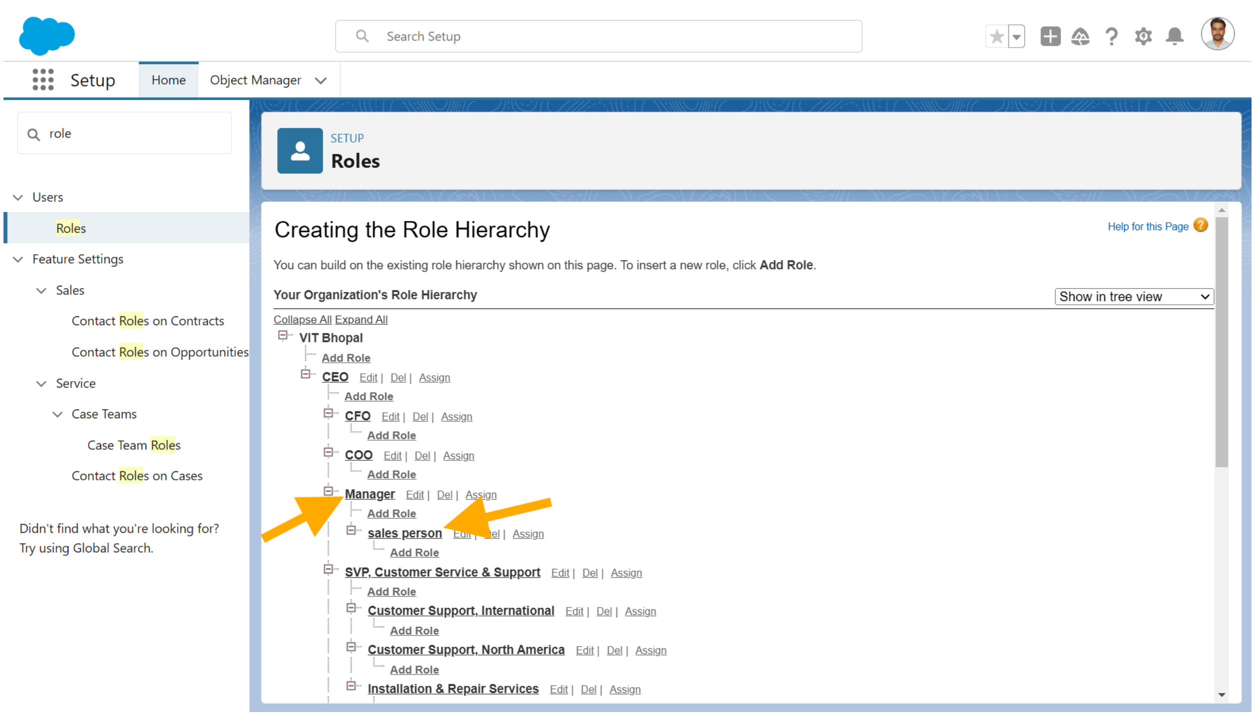
Creating Manager Role:

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click on Expand All 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.

- **Sales Person Role**: Reports to the Manager role.

Creating another two roles under manager

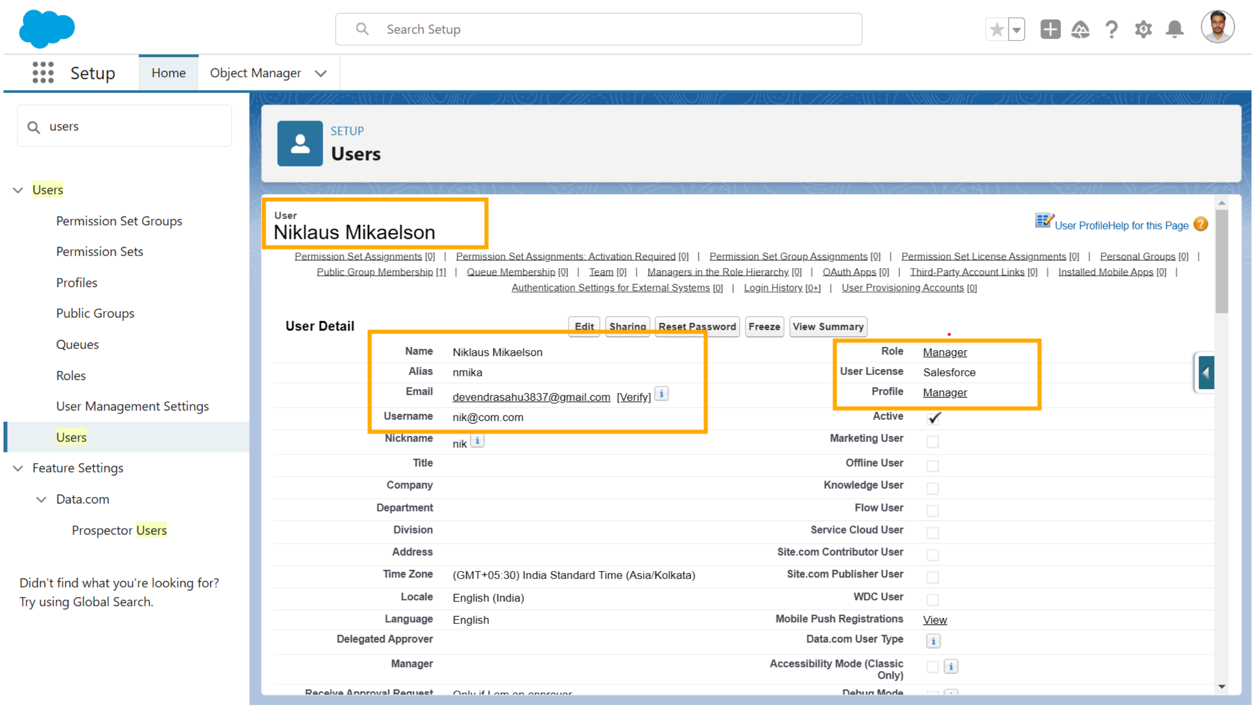
1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click plus on CEO role, and click add role under manager.
3. Give Label as "sales person" and Role name gets auto populated. Then click on Save.



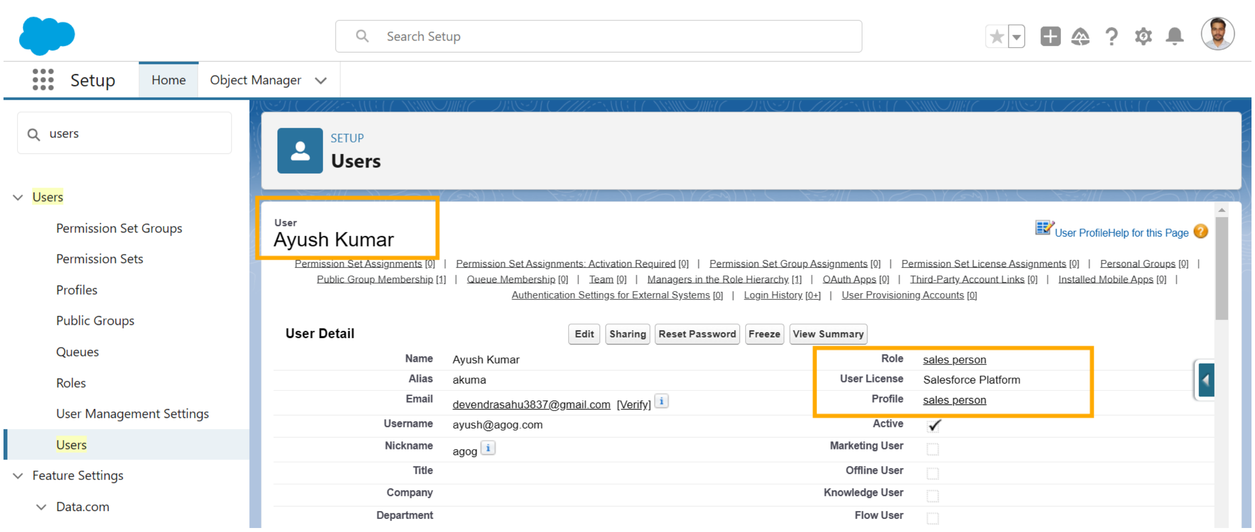
Step 10: Users

A user is anyone who logs in to Salesforce. Users are employees at 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 account identifies the user, and the user account settings determine what features and records the user can access.

- **Manager User**: Role as Manager, Profile as Manager, Salesforce license.



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



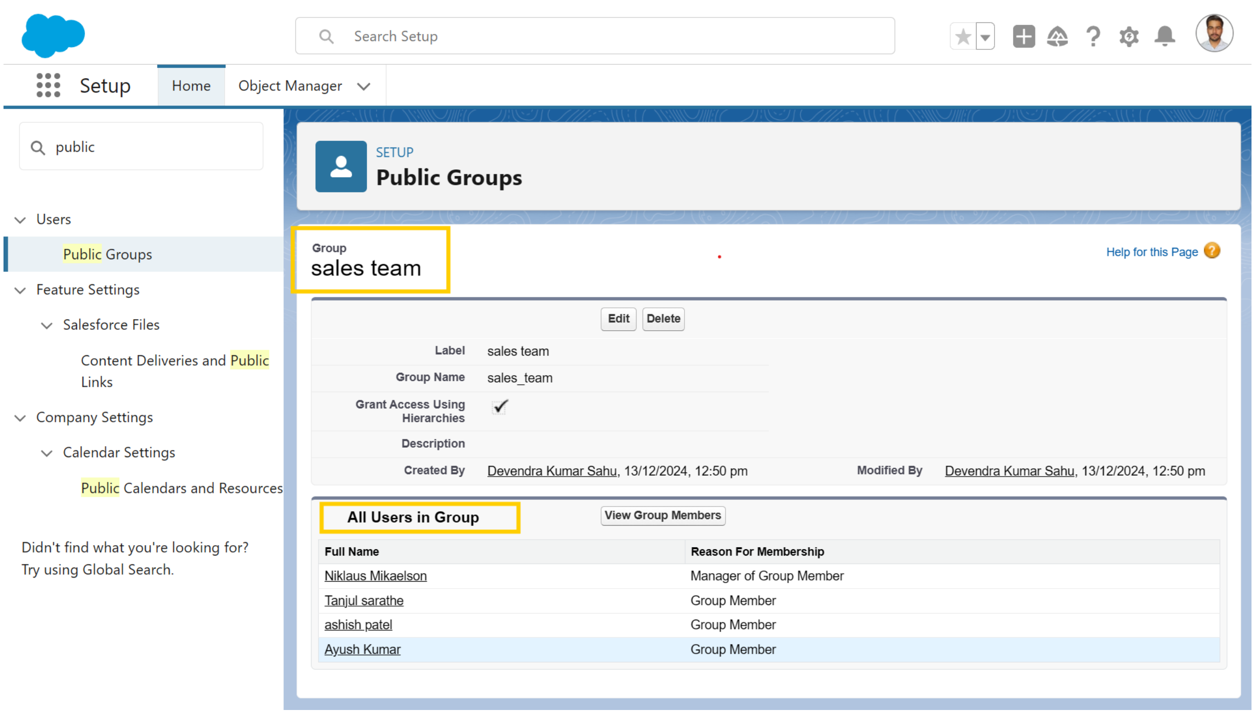
Step 11: Public Groups

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

- **Sales Team Group**: Contains all Sales Person roles for sharing access.

Creating New Public Group

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

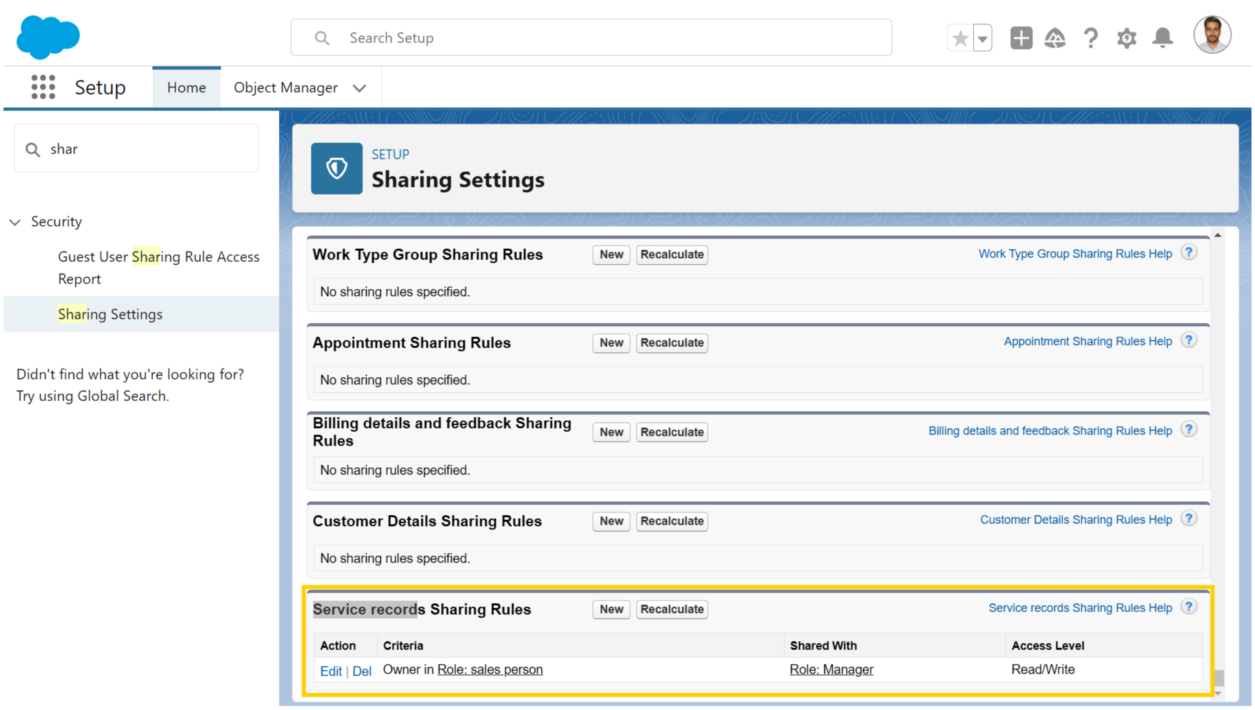


Step 12: Sharing Settings

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

Creating Sharing settings

1. Go to setup >> type users in quick find box >> select Sharing Settings >> click Edit.
2. Change the OWD setting of the Service records Object to private as shown in fig.
3. Click on save and refresh.
4. Scroll down a bit, Click new on Service records sharing Rules.
6. Give the Label name as " Sharing setting"
7. Rule name is auto populated.
8. In step 3 : Select which records to be shared, members of " Roles " >> " Sales person"
9. In step 4: share with, select " Roles " >> " Manager "
10. In step 5 : Change the access level to " Read / write ".
11. Click on save.

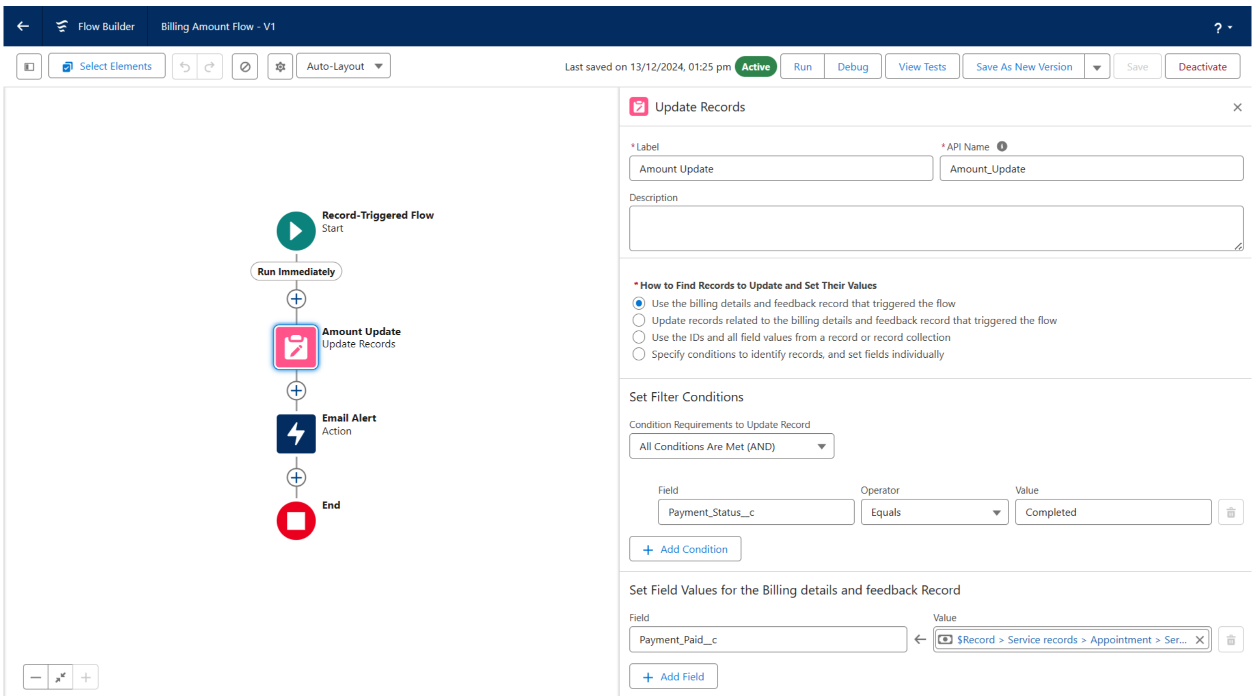


Step 13: Flows

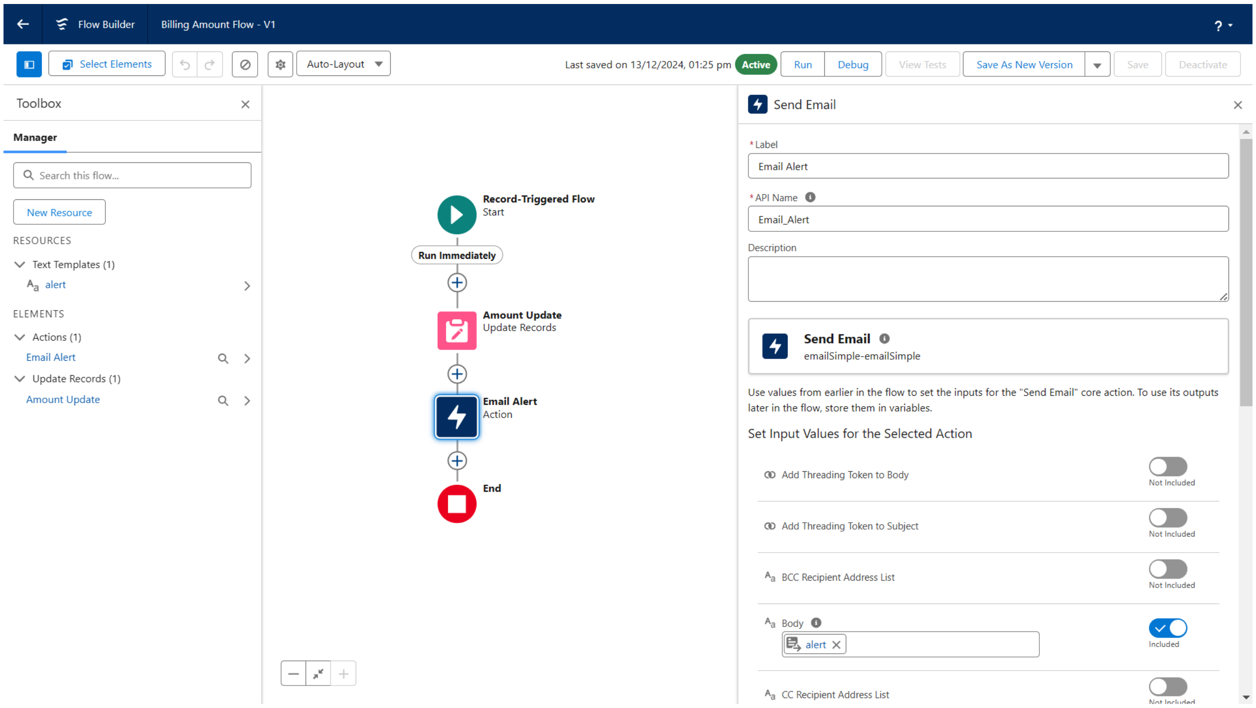
In Salesforce, a flow is a powerful tool that allows you to automate business processes, collect and update data, and guide users through a series of screens or steps. Flows are built using a visual interface and can be created without any coding knowledge.

- **Record-Triggered Flow**: Automates updates and email alerts for billing completion.

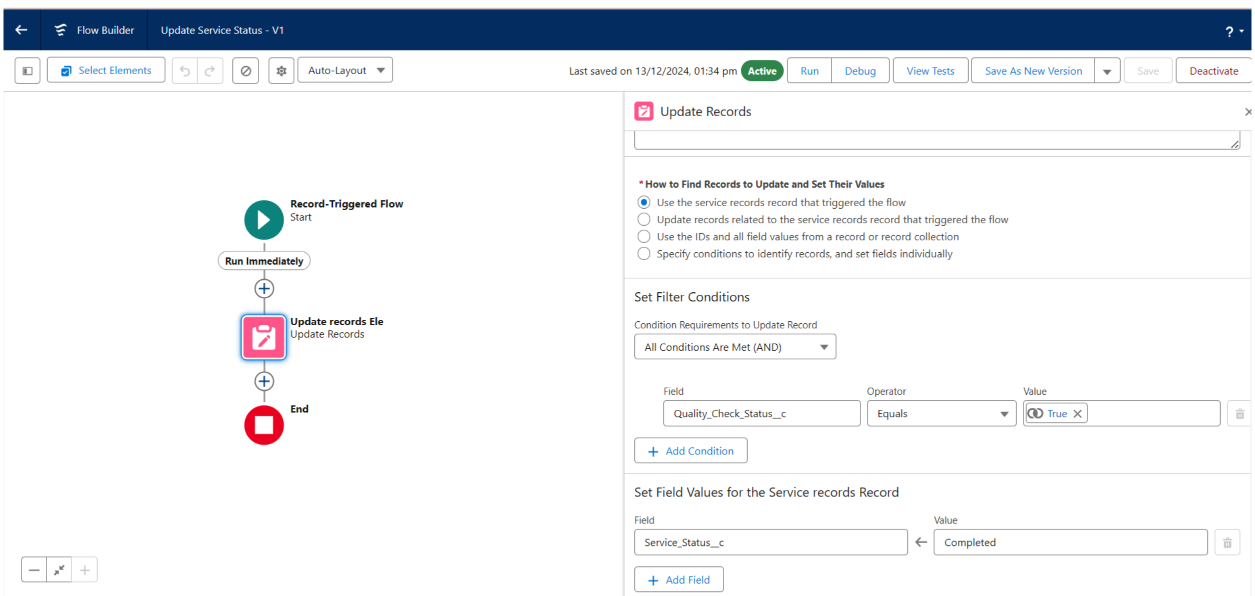
- Under the Record-triggered Flow Click on "+" Symbol and In the Drop down List select the "Update records Element".



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



- **Another Flow - Update Service Status**



Step 14: Apex Triggers

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

- **Handler Class (AmountDistributionHandler)**: Calculates service amounts based on selected services.

Code:

"AccountDistributionHandler.apxc"

```
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 (AmountDistribution)**: Runs on appointment insert or update to update service amounts.

Code:

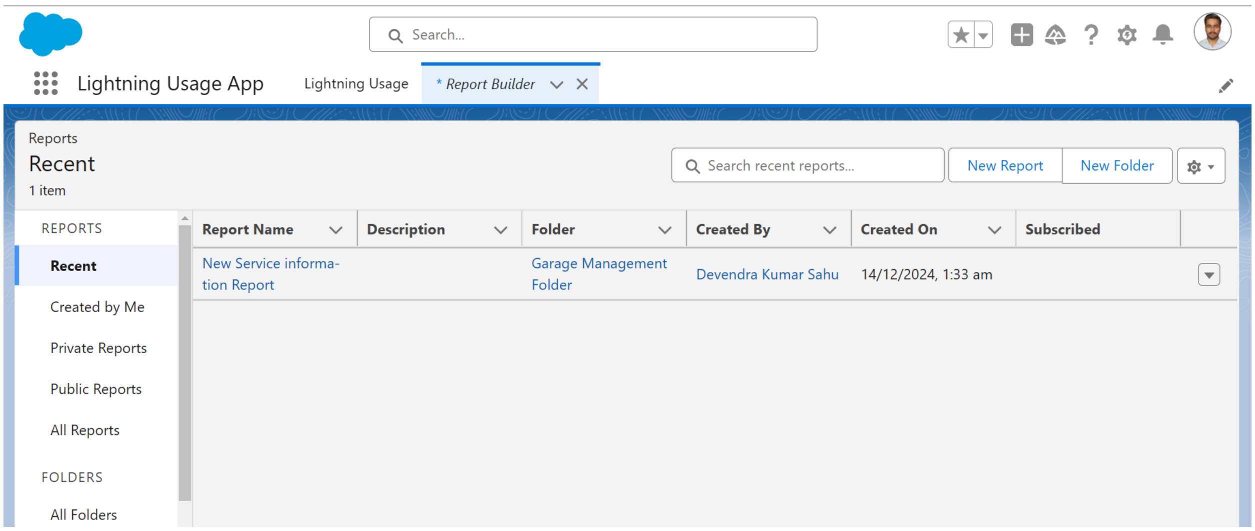
"**AmountDistribution.apxt**"

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

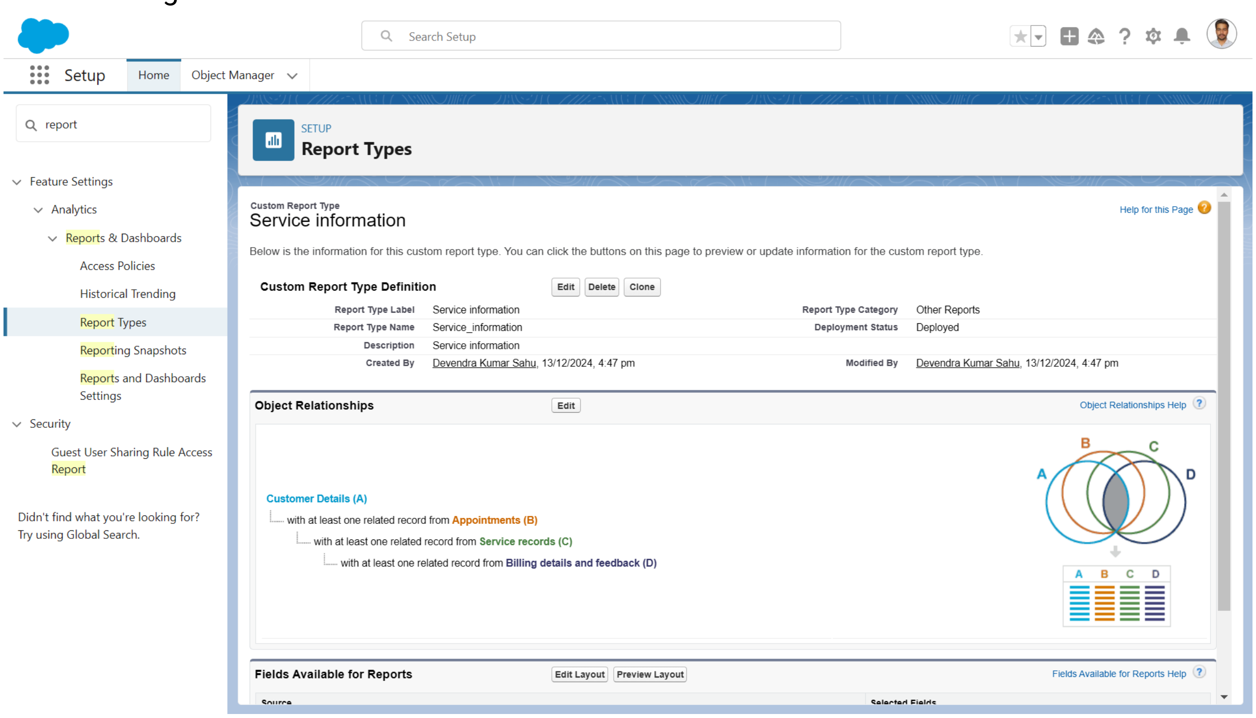
Step 15: Reports

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".



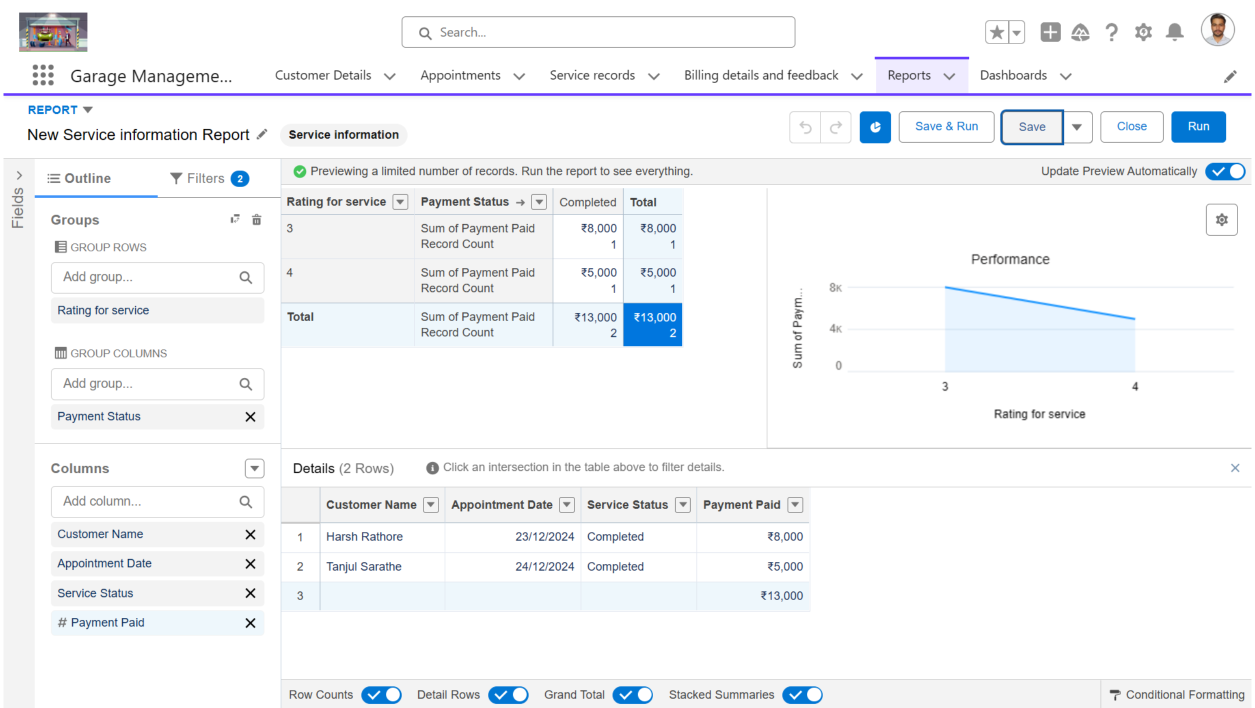
- **Custom Report Type**: Combines Customer Details, Appointments, Service Records, and Billing.



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

Create Report

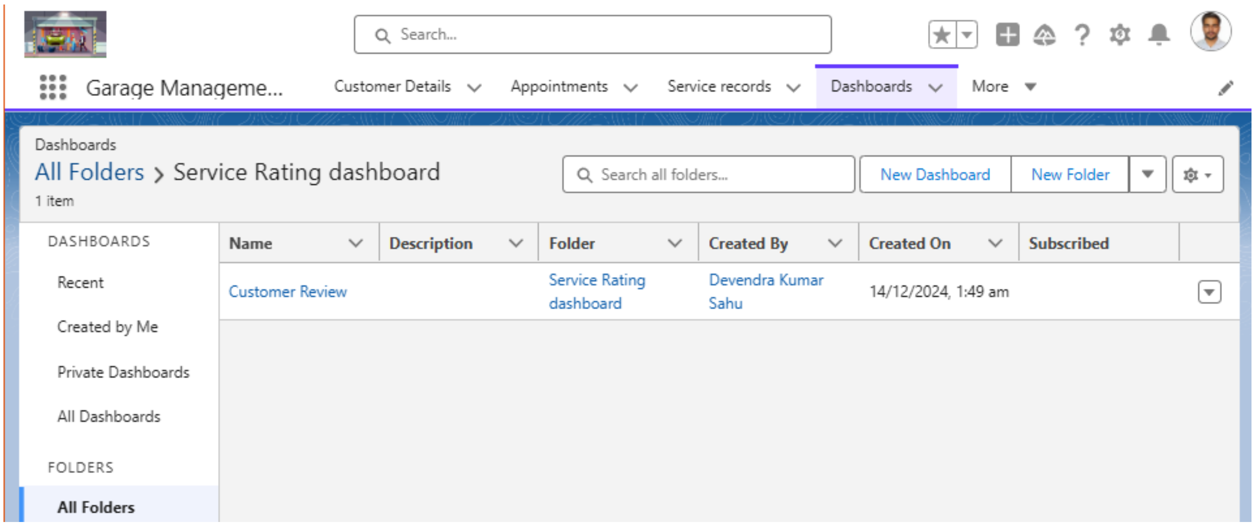
1. Go to the app >> click on the reports tab
2. Click New Report.
3. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.
4. Their outline pane is opened alredy, select the fields that mentioned below in column section.
    1. Customer name
    2. Appointment Date
    3. Service Status
    4. Payment paid
5. Remove the unnecessary fields.
6. Select the fields that mentioned below in GROUP ROWS section.
    1. Rating for Service
7. Select the fields that mentioned below in GROUP ROWS section.
    1. 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.



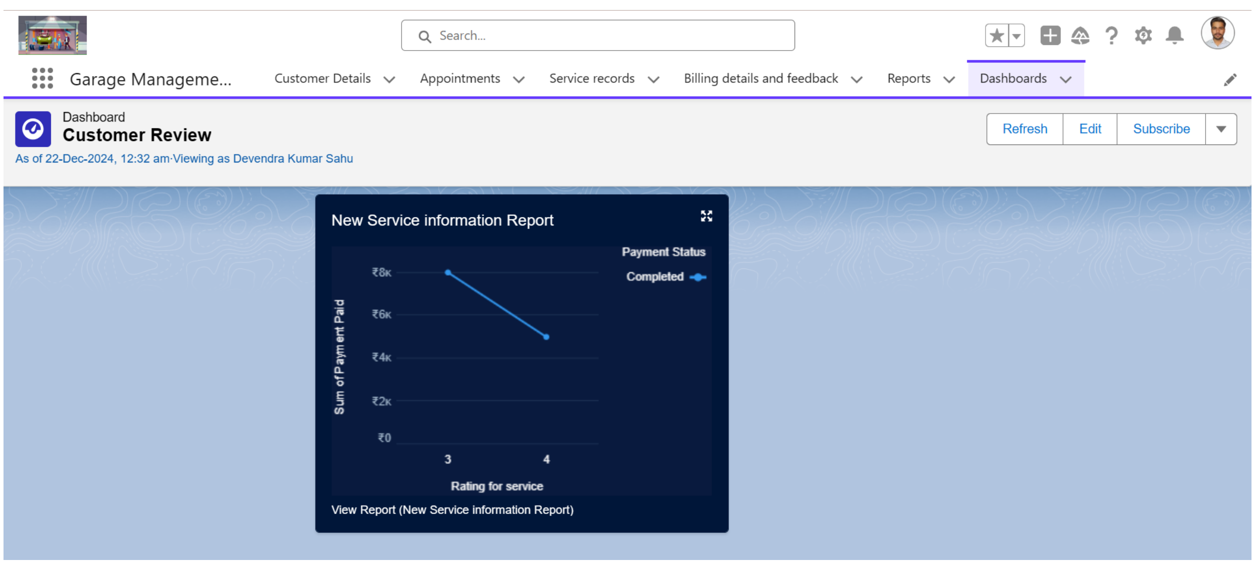
Step 16: Dashboards

Dashboards help you visually understand changing business conditions so you can make decisions based on 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.

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



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

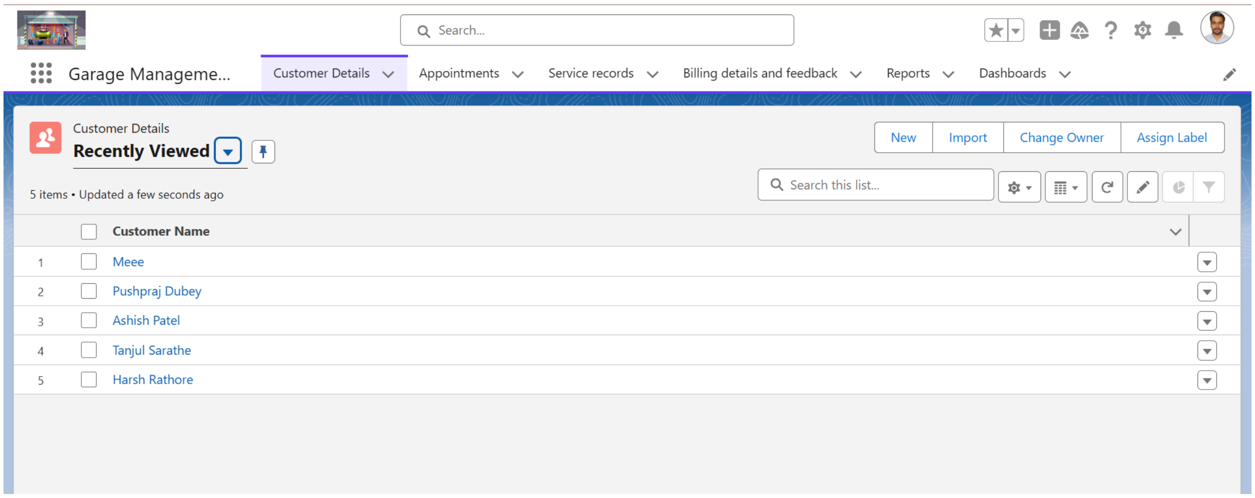


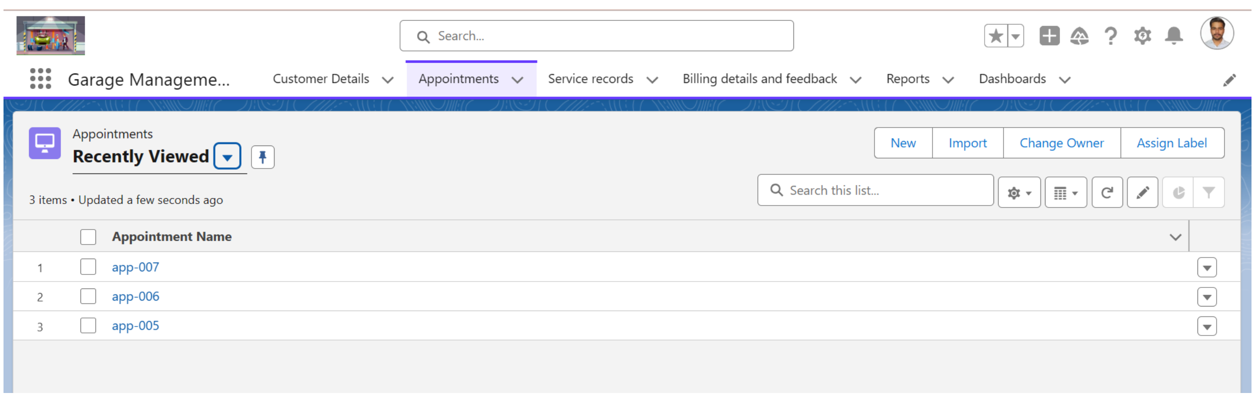
Also Subscribe added:- Set the Frequency as " weekly ". >> Set a day as monday.

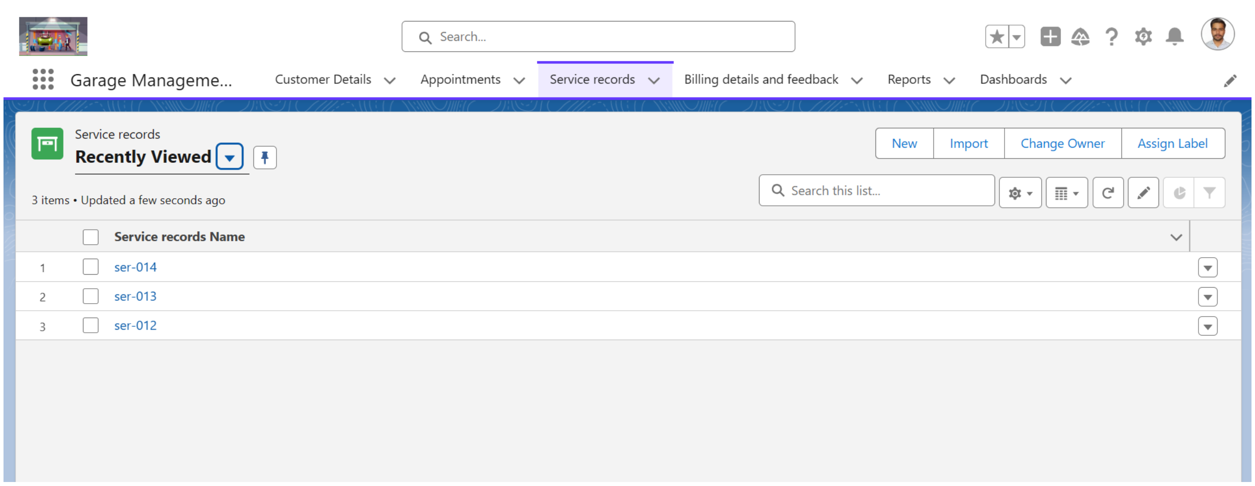
- and then we saved it.

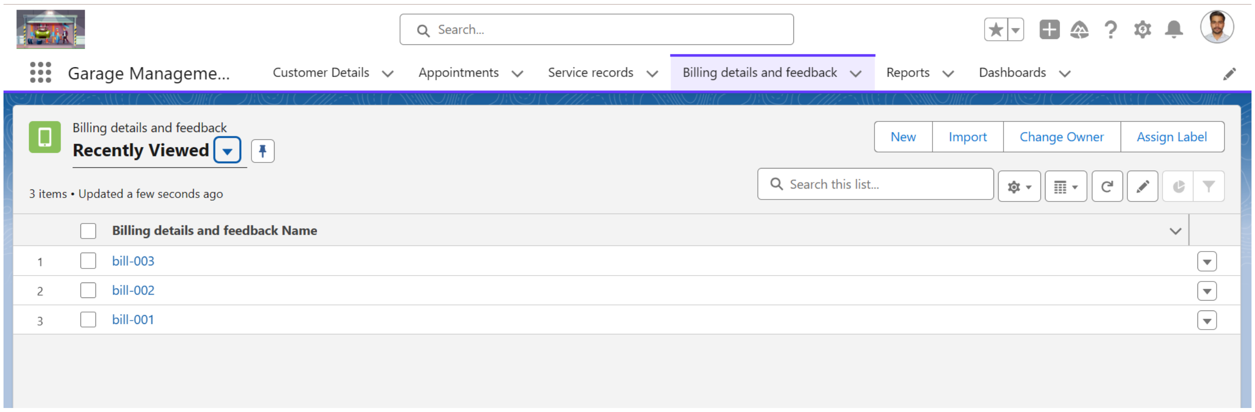
User Adoption

In our Garage Management System we created records for all objects.









5. Testing and Validation

5.1. Unit Testing

- Comprehensive testing of individual components, including Apex classes and triggers, to ensure code accuracy and functionality.
- Each module will be tested in isolation to detect and fix issues at an early stage.

5.2. User Interface Testing

- Ensure the user interface is user-friendly, responsive, and works on different devices (desktop, tablet, mobile).
- Validate the functionality of buttons, links, forms, and navigation.

6. Key Scenarios Addressed by Salesforce in the Implementation Project

This section outlines the key scenarios that Salesforce addresses in the implementation of the Garage Management System:

- **Service Appointment Booking**: Customers can book appointments via a self-service portal.
- **Customer Communication**: Automated notifications and reminders are sent to customers.
- **Inventory Management**: Salesforce tracks spare parts and sends reordering alerts.
- **Job Tracking**: Service advisors and technicians can track the status of jobs and update customers in real time.
- **Billing and Payments**: Generate invoices and process payments through an integrated payment gateway.

- **Customer Feedback Collection**: Customers provide feedback after service completion, which is used to improve service quality.

Conclusion

The Garage Management System built on Salesforce provides a comprehensive platform for managing appointments, services, and billing. Through the use of custom objects, tabs, profiles, flows, validation rules, and Apex triggers, the system ensures smooth operations, efficient record management, and enhanced customer satisfaction. This project enables better tracking of garage operations, fosters data-driven decision-making, and supports the long-term growth of garage businesses.

Thank You