Garage Management System

COLLEGE NAME: Sri Krishna Adithya College of Arts and Science

COLLEGE CODE: Bruag

TEAM ID: NM2025TMID22663

TEAM MEMBERS: 4

Team Leader Name: SHAHITH NAFAL D

Email: 23bsds153shahithnafald@skacas.ac.in

Team Member 1: SANJAY S

Email: 23bsds152sanjays@skacas.ac.in

Team Member 2: ADITHYA KRISHNA M

Email:23bsds102adithyakrishnam@skacas.ac.in

Team Member 3: NITHISHVAR M

Email:23bsds141nithishvarm@skacas.ac.in

Title:- Garage Management System

Project Overview:

The **Garage Management System** is a Salesforce-based solution designed to streamline the management of automotive service centers. It connects **customers, service advisors, mechanics, inventory managers, and billing staff** to ensure smooth operations. Using **custom objects, workflows, automation, and reports**, the system digitizes the entire process of **vehicle service booking, job tracking, spare parts management, and invoicing**.

The solution improves efficiency, enhances customer satisfaction, and ensures transparency in service history, cost estimation, and resource utilization.

Objectives:

1. **Streamline Service Operations**
    - Manage vehicle service bookings, job assignments, and progress tracking digitally.
2. **Enhance Customer Experience**
    - Provide real-time updates on service status, cost estimates, and delivery timelines.
3. **Optimize Resource & Inventory Management**
    - Track spare parts availability, consumption, and procurement to reduce delays.
4. **Improve Staff Coordination**
    - Automate task allocation, notifications, and approvals for mechanics and service advisors.
5. **Ensure Transparency & Accountability**
    - Maintain detailed records of service history, invoices, and payments for future reference.
6. **Enable Data-Driven Decisions**
    - Generate reports and dashboards to monitor revenue, service trends, and customer satisfaction.

Student Outcomes:

- **Hands-on Experience with Garage Operations Automation:** Students gain practical skills in configuring Salesforce objects, automating service workflows, and managing vehicle service records.
- **Understanding of Project Lifecycle in Automotive CRM:** Students learn the complete process from requirement gathering to deployment, enhancing their ability to deliver real-world Salesforce projects.
- **Enhanced Analytical and Problem-Solving Skills:** Students develop the ability to address operational challenges such as job tracking, inventory shortages, and billing errors.
- **Improved Collaboration Skills:** Students practice teamwork by coordinating tasks across modules like service scheduling, inventory management, and reporting.
- **Industry-Relevant Exposure:** Students explore practical use cases of Salesforce CRM in the automotive service industry, preparing them for careers in CRM and service management.

System Requirements:

Hardware Requirements:

- Computer with minimum 4 GB RAM, Dual-core processor
- Stable internet connection

Software Requirements:

- Salesforce Developer Edition Org
- Modern Web Browser (e.g., Google Chrome, Firefox)

Project Duration: 31 Hours

Phases Overview:

Phase No.	Phase Name	Description	Page Numbers
1	Requirement Analysis & Planning	Gathering requirements from donors, volunteers, and receivers; defining scope and goals; planning data model and workflows.	4
2	Salesforce Development – Backend & Configurations	Creating custom objects, fields, relationships; setting up Flows and Apex Triggers for automation.	4 - 11
3	UI/UX Development & Customization	Building Lightning App, customizing layouts, adding fields, implementing Flows, and developing UI logic.	11 - 28
4	Data Migration, Testing & Security	Creating Users, Profiles, Public Groups, Sharing Rules; configuring Report Types, Reports, Dashboards; testing functionalities and ensuring data security.	28 - 37
5	Deployment, Documentation & Maintenance	Designing and finalizing Home Page, deploying solution to live environment, preparing documentation, conclusion, and ongoing system maintenance.	37 - 40

Phase 1: Requirement Analysis & Planning:-

To make garage management system:

Utilizing Salesforce, our project streamlines vehicle service management, spare parts tracking, and customer coordination, ensuring efficiency and transparency.

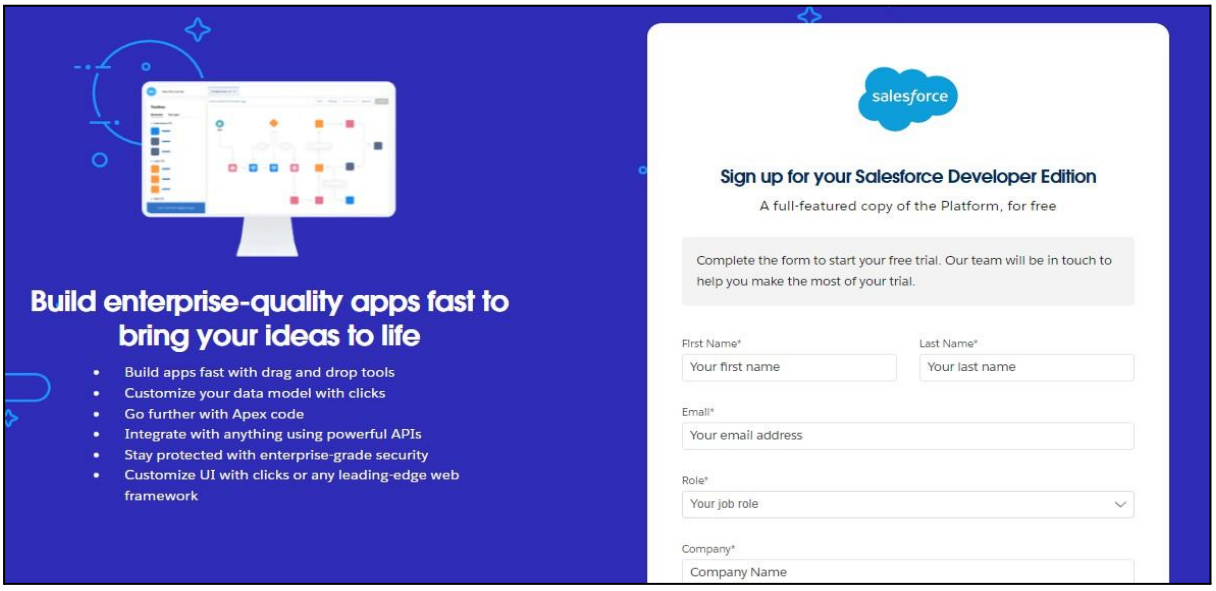
Phase 2: Salesforce Development – Backend & Configurations:-

Milestone 1: Salesforce developer account creation Activity

1: Creating Developer Account

Creating a developer org in salesforce.

1. Go to https://developer.salesforce.com/signup



2. On the sign up form, enter the following details :
    1. First name & Last name
    2. Email
    3. Role : Developer
    4. Company : College or Company Name

5. County: India

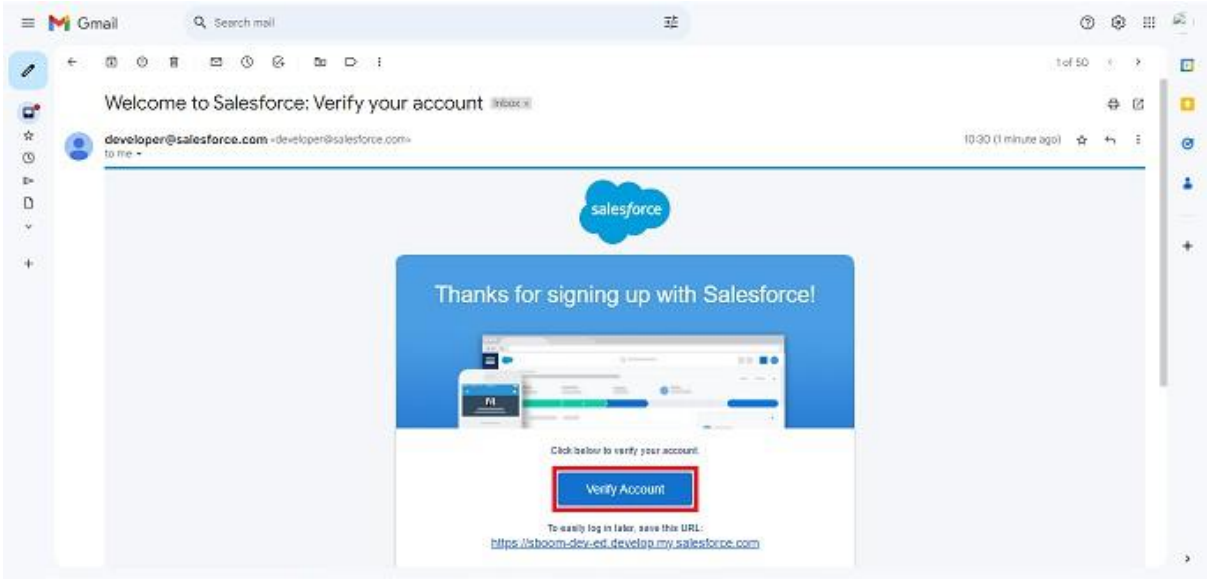
G. Postal Code: pin code

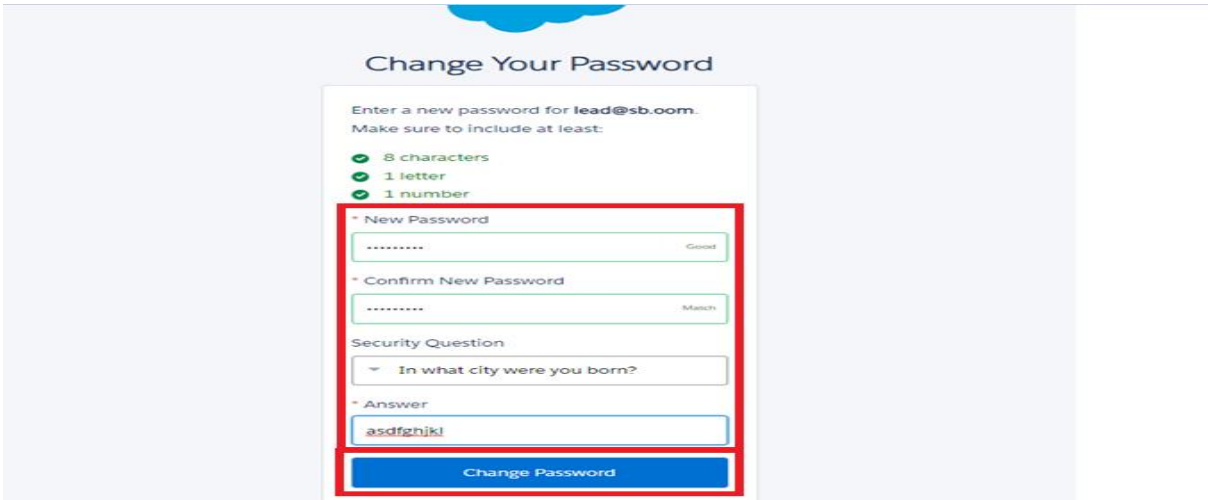
7. Username: should be a combination of your name and company This need not be an actual email id, you can give anything in the format: username@organization.com

Click on sign me up after filling these.

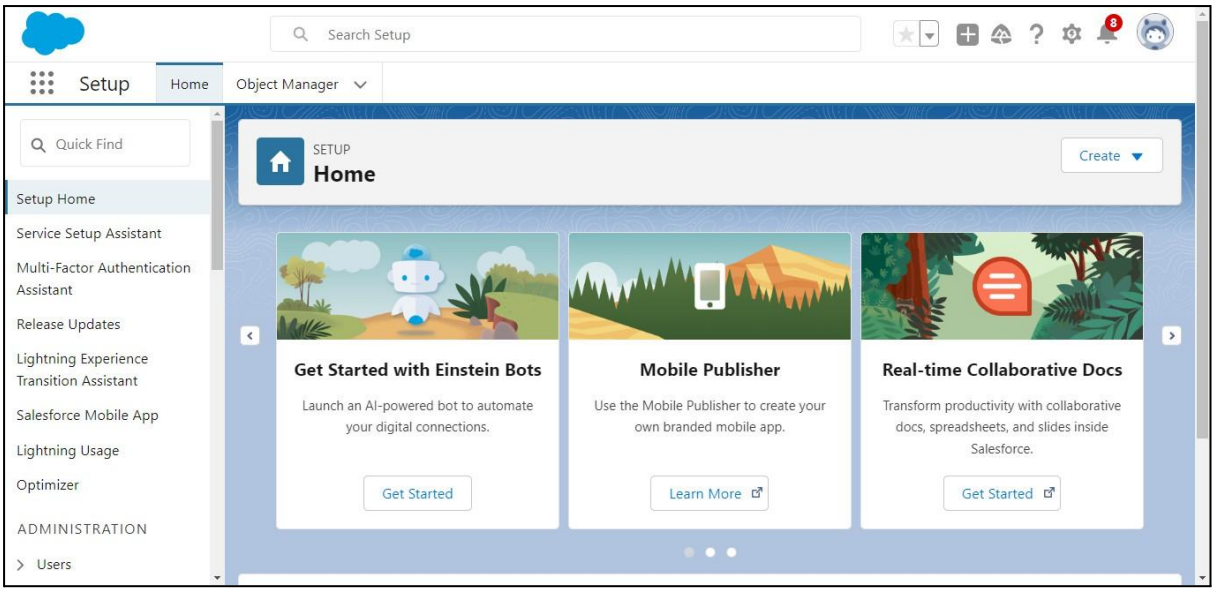
Activity 2: Account Activation

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.





1. Click on Verify Account
2. Give a password and answer a security question and click on change password.
3. Then you will redirect to your salesforce setup page.

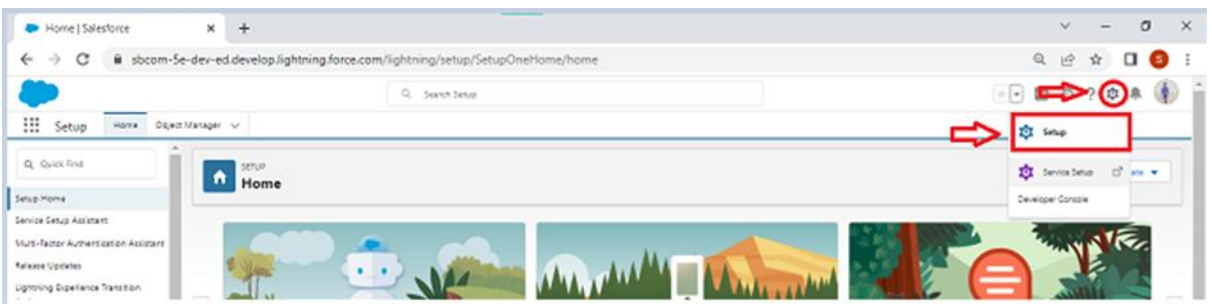


OBJECT:

1. **Standard Objects**: Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
2. **Custom Objects**: Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.

To Navigate to Setup page:

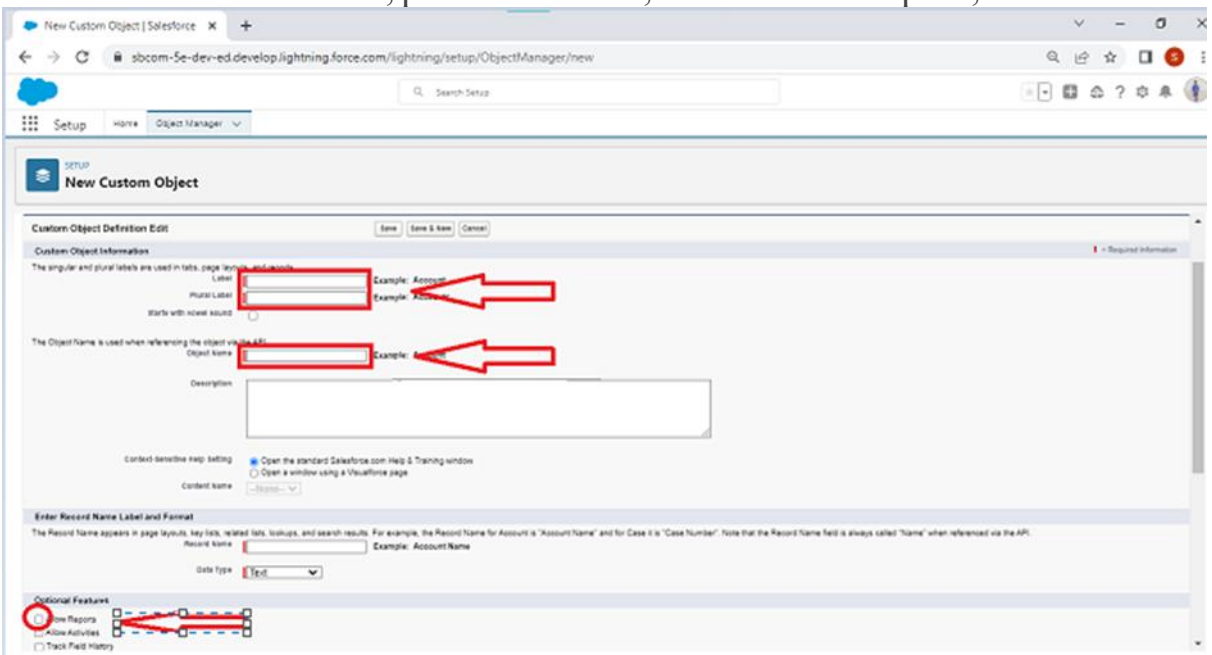
Click on gear icon? click setup.

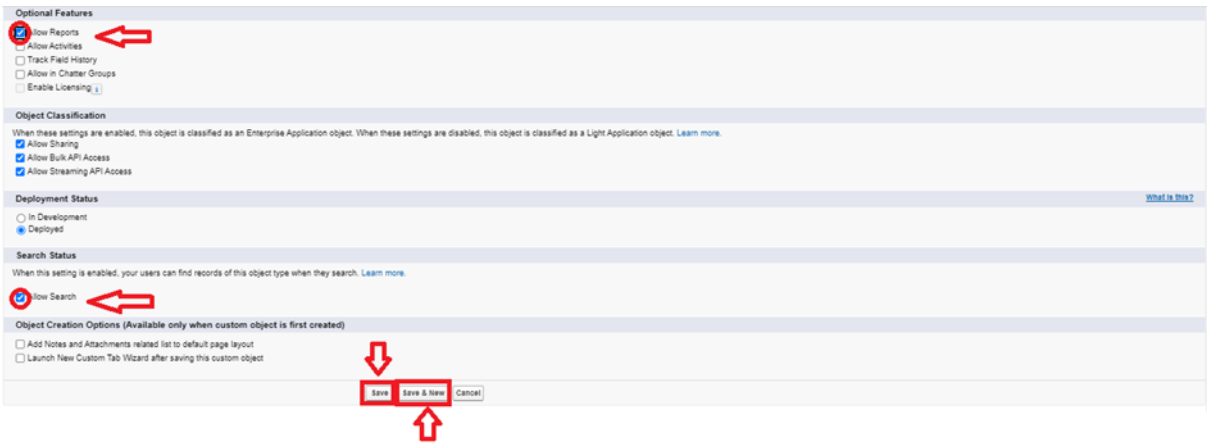


To create an object:

1. From the setup page?
2. Click on Object Manager?
3. Click on Create? Click on Custom Object.

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





3. Click on Save.

Activity 1: Create Customer Details Object:

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

1. Enter the label name >> Customer Details
2. Plural label name >> Customer Details
3. Enter Record Name Label and Format

- Record Name >> Customer Name
- Data Type >> Text
- Click on Allow reports and Track Field History,
- Allow search >> Save.

Activity 2: Create Appointment Object:

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

1. Enter the label name >> Appointment
2. Plural label name >> Appointments
3. Enter Record Name Label and Format

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

2. Click on Allow reports and Track Field History,
3. Allow search >> Save.

Activity 3: Create Service Records Object:

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

1. Enter the label name >> Service records
2. Plural label name >> Service records
3. Enter Record Name Label and Format

- Record Name >>Service records Name

- Data Type >> Auto Number
- Display Format >> ser-{000}
- Starting number >> 1
- Click on Allow reports and Track Field History,
- Allow search >> Save.

Activity 4: Create Billing Details and Feedback Object:

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

1. Enter the label name >> Billing details and feedback
2. Plural label name >> Billing details and feedback
3. Enter Record Name Label and Format

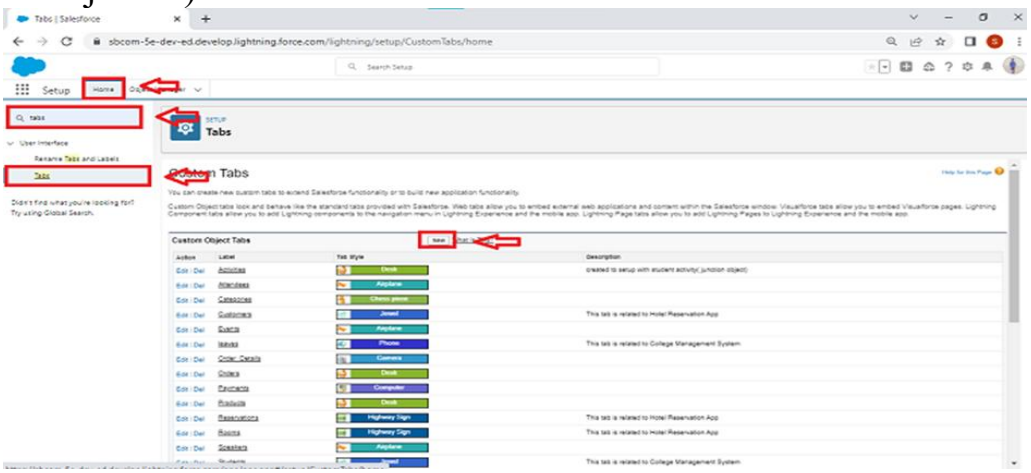
- Record Name >> Billing details and feedback Name
- Data Type >> Auto Number
- Display Format >> bill-{000}
- Starting number >> 1
- Click on Allow reports and Track Field History,
- Allow search >> Save.

TABS:

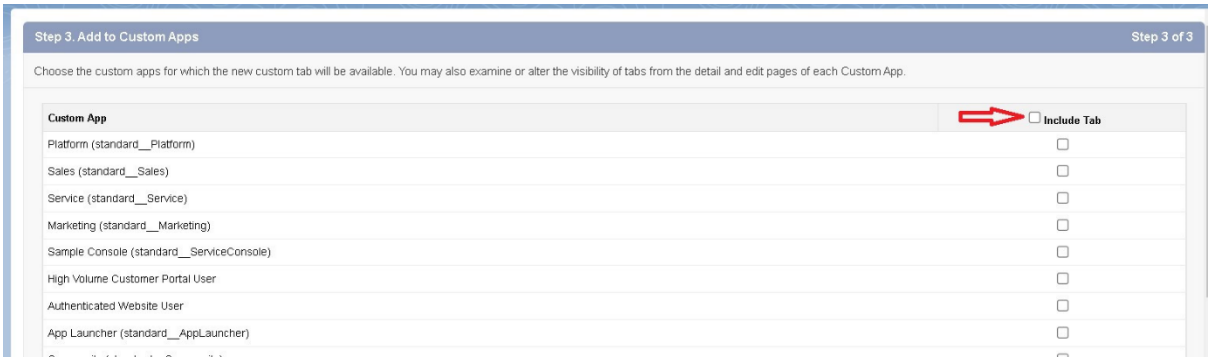
Activity 1: Creating a Custom Tab:

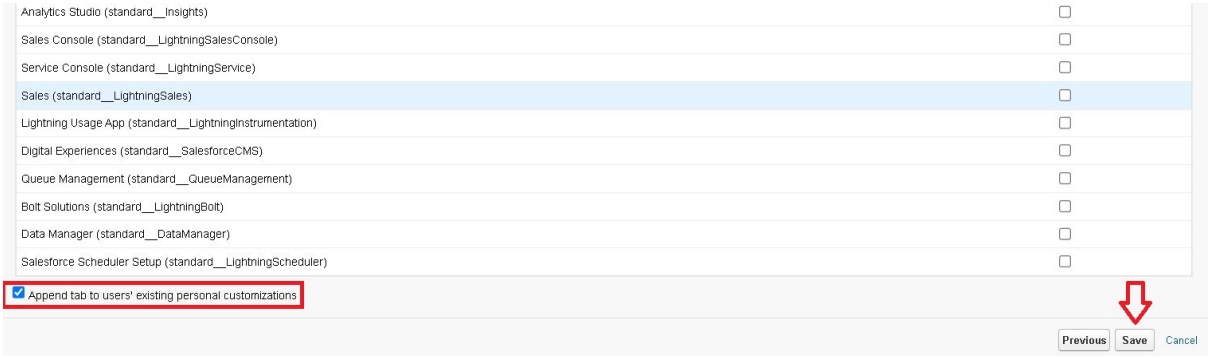
To create a Tab:(Customer Details)

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



2. 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 .
3. Make sure that the Append tab to users' existing personal customizations is checked.
4. Click save.





Activity 2: Creating Remaining Tabs:-

1. Now create the Tabs for the remaining Objects, they are “Drop-Off Point, Task, Volunteer, Execution Details”.
2. Follow the same steps as mentioned in Activity -1.

Phase 3: UI/UX Development & Customization:

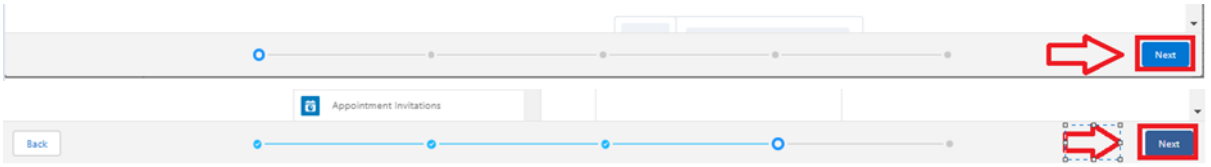
THE LIGHTNING APP:

Create a Lightning App

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

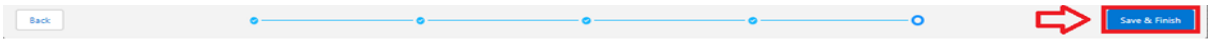


Next.

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:



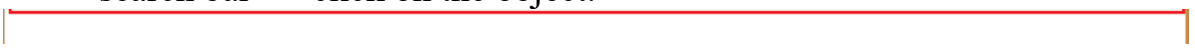
Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.

FIELDS:

Activity 1: Creation of fields for the customer details object:

Creation of fields for the Customer Details object

1. To create fields in an object:
    1. Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.



2. Now click on “Fields & Relationships” >> New

3. Select Data Type as a “Phone”

4. Click on next.

5. Fill the Above as following:
    - Field Label: Phone number
    - Field Name : gets auto generated
    - Click on Next >> Next >> Save and new.

Note: Follow the above steps for the remaining field for the same object.

2. To create another fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Email” and Click on Next
4. Fill the Above as following:

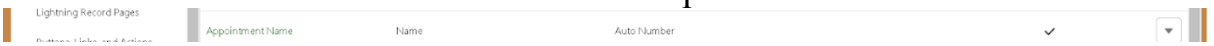
- Field Label : Gmail
- Field Name : gets auto generated
- Click on Next >> Next >> Save and new.

Activity 2: Creation of Lookup Relationship Field on Volunteer Object:

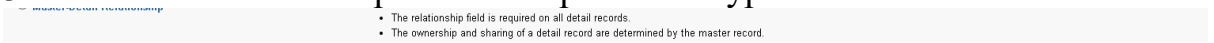
Creation of Lookup Field on Appointment Object :

1. Go to setup >> click on Object Manager >> type object name(Appointment) in the search bar >> click on the object.

2. Now click on “Fields & Relationships” >> New



3. Select “Look-up relationship” as data type and click Next.

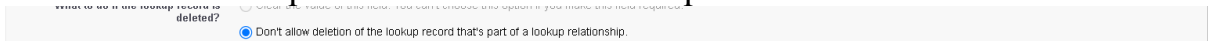


4. Select the related object “ Customer Details” and click next.
5. Next >> Next >> Save.

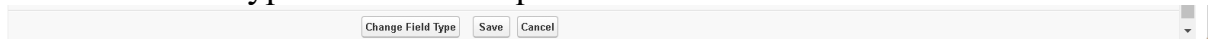
Note: Make sure you complete Activity 4 Before continuing.

Creation of Lookup Field on Service records Object :

1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select “Look-up relationship” as data type and click Next.
4. Select the related object “ Appointment ” and click next.
5. Make it a required field so click on Required.



6. Scroll down for Lookup Filter and click on Show filter settings.
7. Now add the filter criteria.
8. Field : Appointment: Appointment Date >> Operator : less than >> select field >> Appointment: Created Date
9. Filter type should be Required.



10. Error Message : Value does not match the criteria.
11. Enable the filter by click on Active.
12. Next >> Next >> Save.

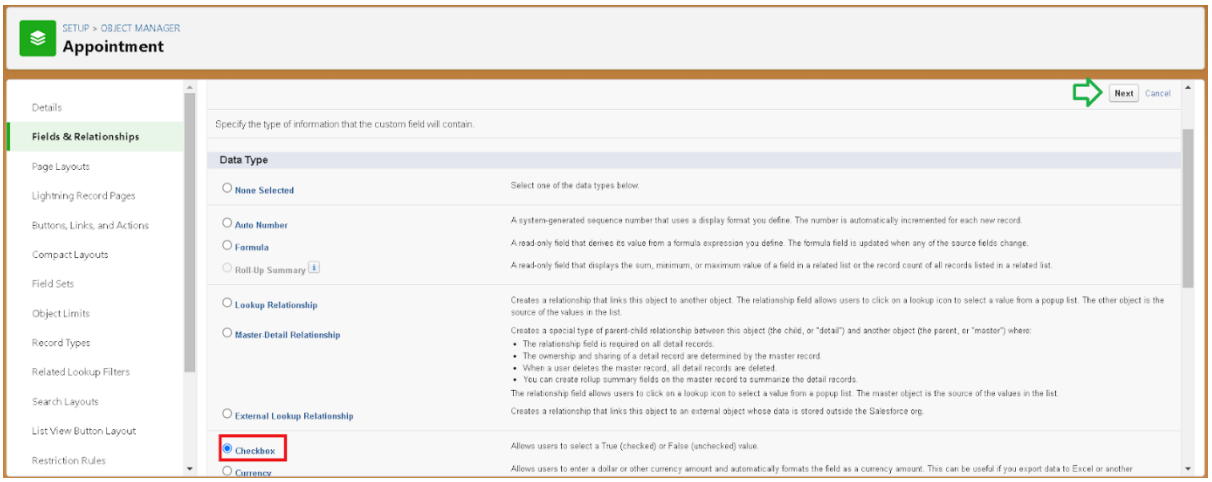
Creation of Lookup Field on Billing details and feedback Object :

1. Go to setup >> click on Object Manager >> type object name(Billing details and feedback) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Look-up relationship” as data type and click Next.
4. Select the related object “ Service records” and click next.
5. Next >> Next >> Save & new.

Activity 3: Creation of Master-Detail Relationship Field on Execution Detail Object:

Creation of Checkbox Field on Appointment Object :

1. Go to setup >> click on Object Manager >> type object name(Appointment) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Check box” as data type and click Next.



4. Give the Field Label : Maintenance service
5. Field Name : is auto populated
6. Default value : unchecked

7. Click on next >> next >> save.

Activity 4: Creation of Date Field on Appointment Object:

Creation of date Fields

Creation of Date Field on Appointment Object :

1. Go to setup >> click on Object Manager >> type object name(Appointment) in the search bar >> click on the object.
2. Now click on "Fields & Relationships" >> New.
3. Select "Date" as data type and click Next.
4. Give the Field Label : Appointment Date
5. Field Nme : is auto populated
6. Make it as a Required field by click on the Required option.
7. Click on next >> next >> save.

Activity 5: Creation of Look-Up Relationship Field on Drop-off Point Object:

Creation of Currency Field on Appointment Object :

1. Go to setup >> click on Object Manager >> type object name(Appointment) in the search bar >> click on the object.
2. Now click on "Fields & Relationships" >> New.
3. Select "Currency" as data type and click Next.
4. Give the Field Label : Service Amount
5. Field Nme : is auto populated
6. Click on next
7. Give read only for all the profiles in field level security for profile.
8. Click on next >> save.

Activity 6: Creation of Lookup Relationship Field on Task Object:

Creation of Text Fields

1. Go to setup >> click on Object Manager >> type object name(Appointment) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Text” as data type and click Next.
4. Give the Field Label : Vehicle number plate
5. Field Name : is auto populated
6. Length : 10
7. Make field as Required and Unique.
8. Click on next >> next >> save.

Activity 7: Creation of Lookup Relationship Field on Task Object:

Creation of Picklist Fields in Service records object :

1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Picklist” and click Next.
4. Enter Field Label as “Service Status”, under values select “Enter values, with each value separated by a new line" and enter values as shown below.
5. The values are: Started, Completed.
6. Click Next.
7. Next >> Next >> Save.

Activity 8: Creation of Fields for Venue Object:

Creating Formula Field in Service records Object

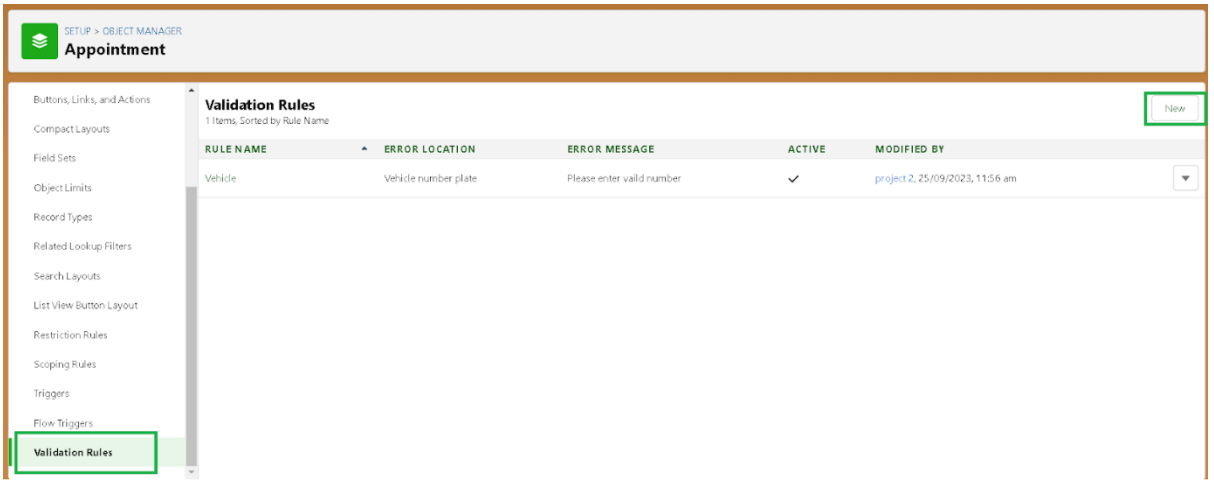
1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Formula” and click Next.
4. Give Field Label and Field Name as “service date” and select formula return type as “Date” and click next.
5. Insert field formula should be : CreatedDate

6. click “Check Syntax” .
7. Click next >> next >> Save.

VALIDATION RULE:

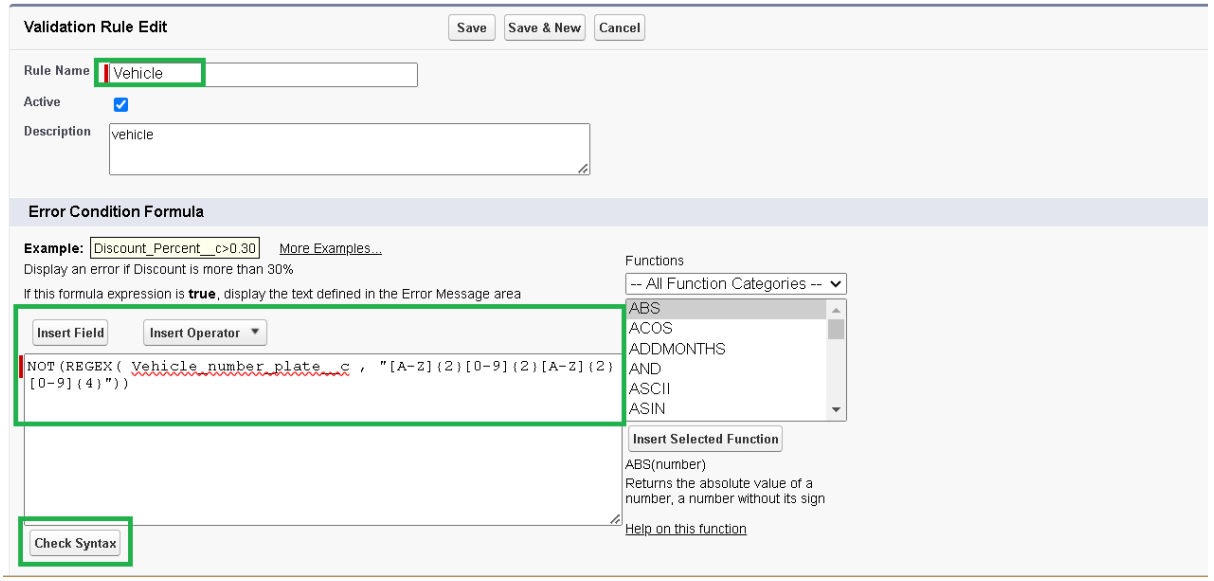
Activity 2: Create a validation rule to an appointment object:

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.

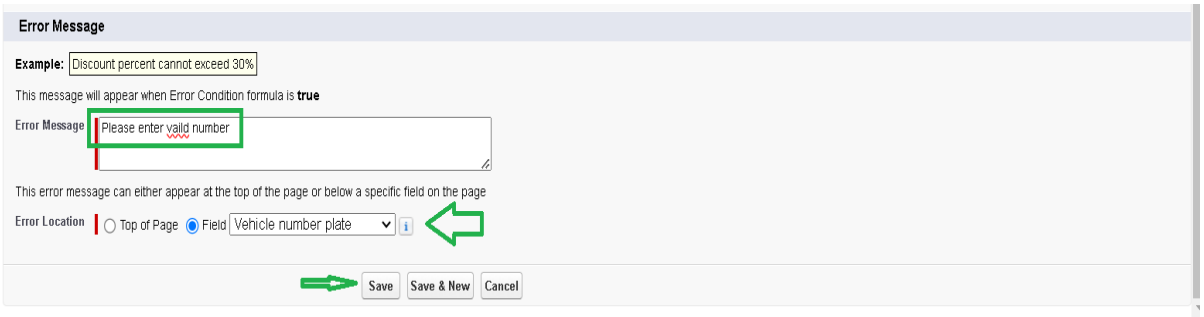


3. Enter the Rule name as “ Vehicle ”.
4. Insert the Error Condition Formula as : -

NOT(REGEX(Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))



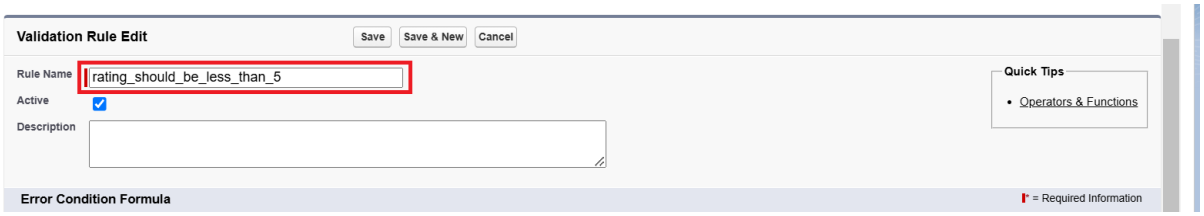
5. Enter the Error Message as “Please enter vaild number ”, select the Error location as Field and select the field as “Vehicle number plate”, and click Save.



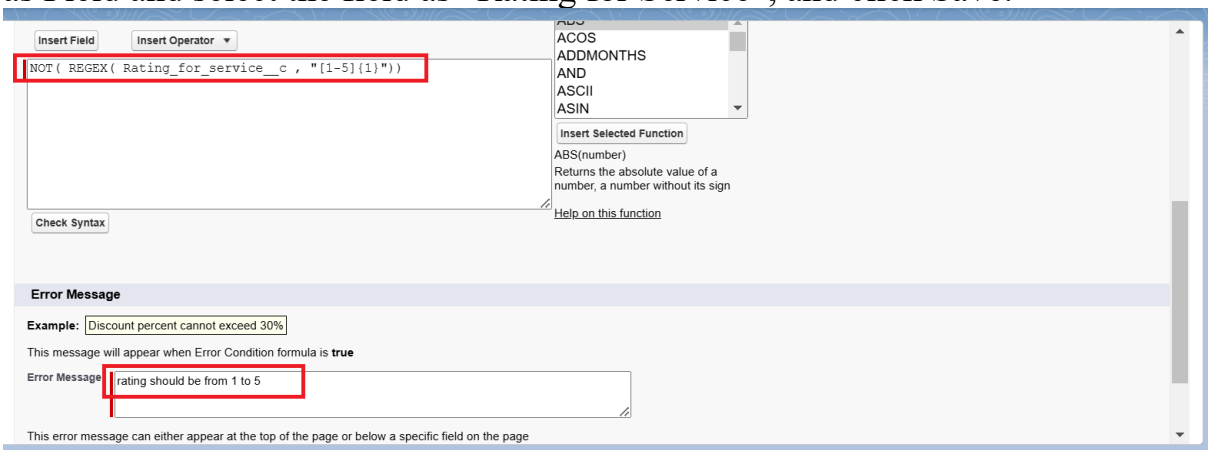
Activity 2: Create a validation to an billing details and feedback object:

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.

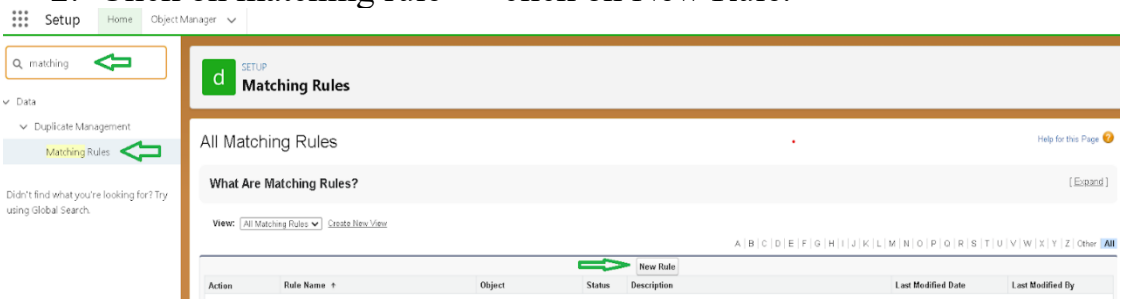


DUPLICATE RULE:

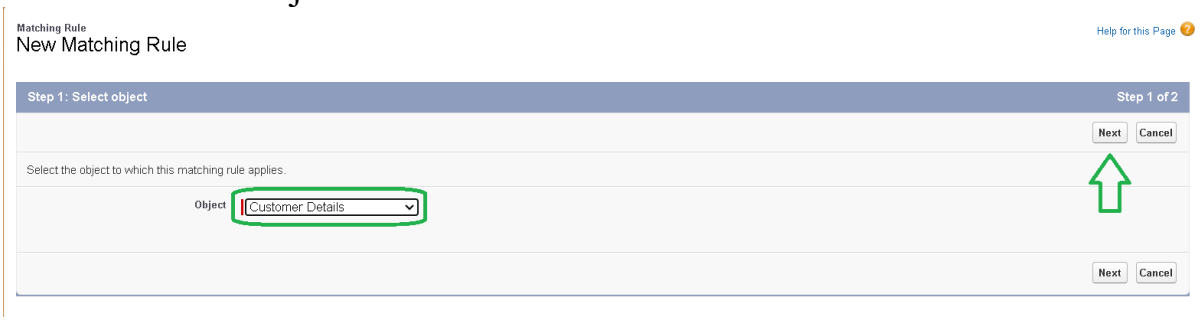
Activity 1: Create a matching rule to an customer details object:

To create a matching rule to an Customer details Object

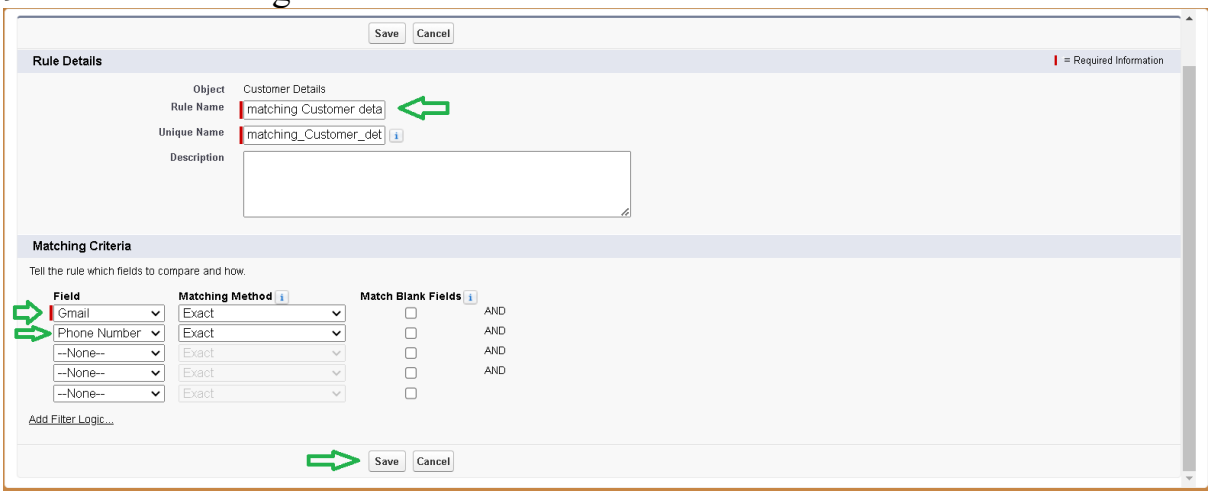
1. Go to quick find box in setup and search for matching Rule.
2. Click on matching rule >> click on New Rule.



3. Select the object as Customer details and click Next.



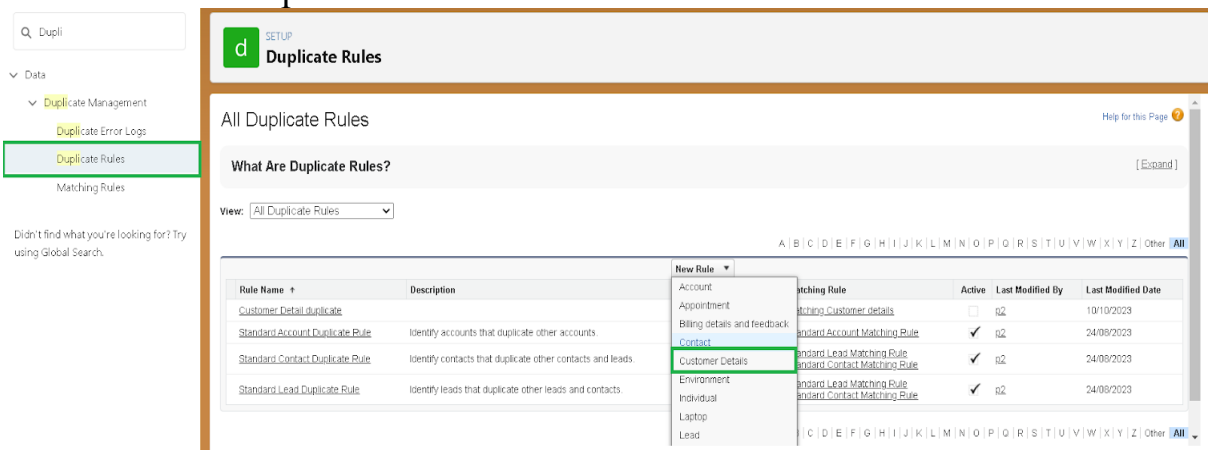
4. Give the Rule name : Matching customer details
5. Unique name : is auto populated
6. Define the matching criteria as
7. Field Matching Method
   1. Gmail Exact
   2. Phone Number Exact
8. Click save.
9. After Saving Click on Activate.



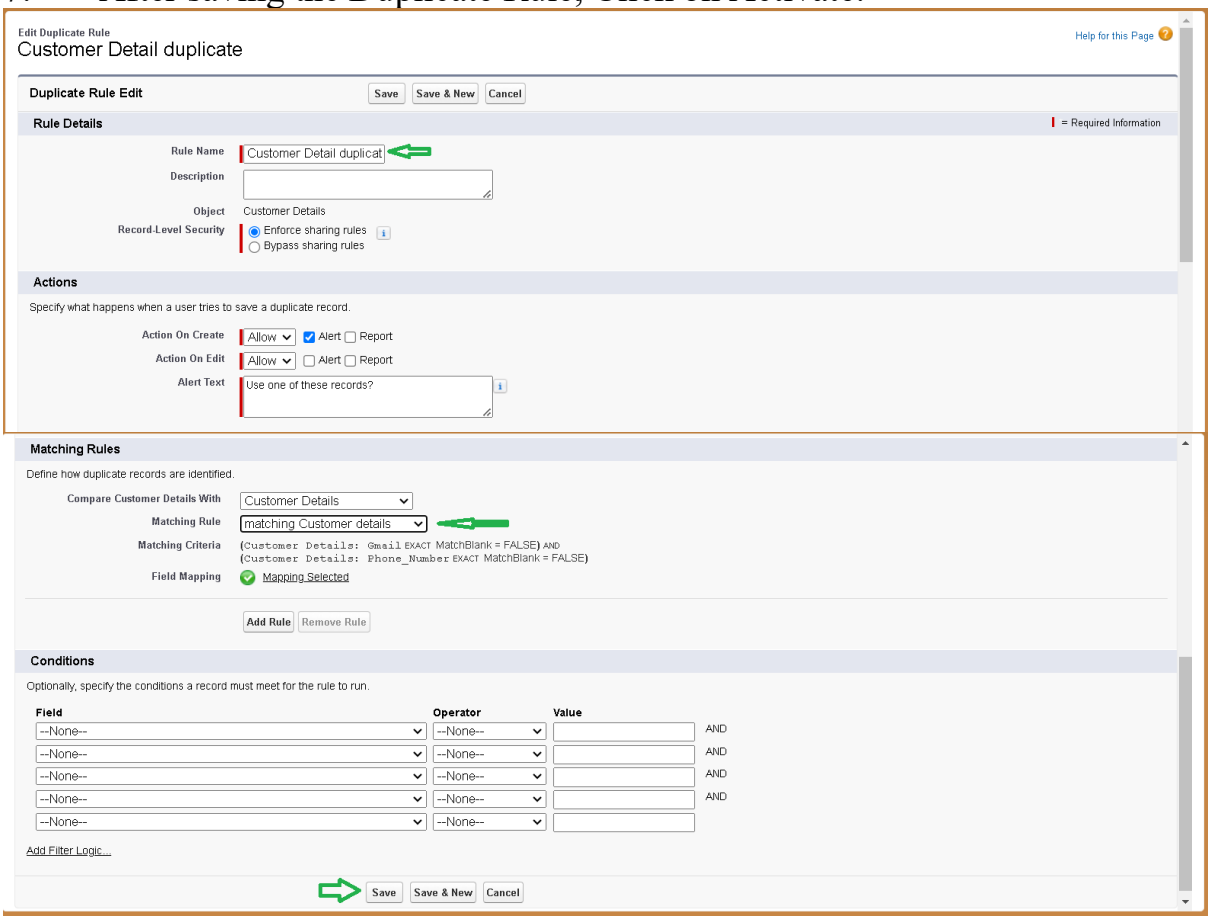
Activity 2: Create a duplicate rule to an customer details object:

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.

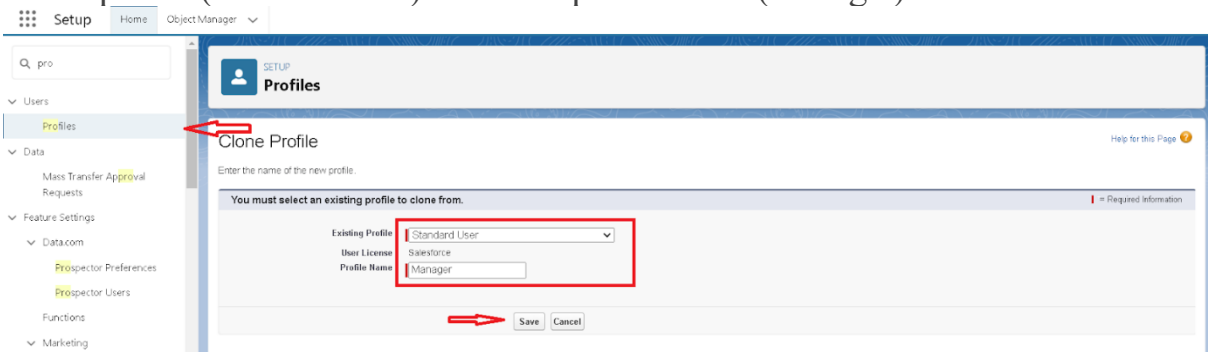


PROFILES:

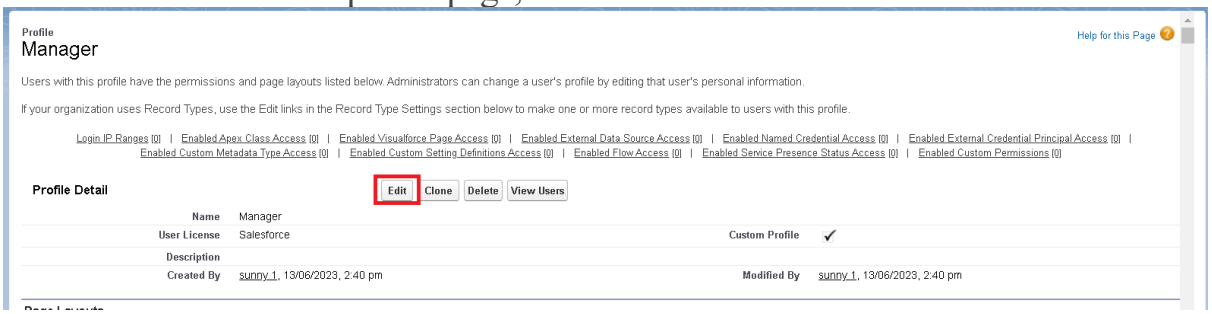
Activity 1: Create a manager profile:

To create a new profile:

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Manager) >> Save.

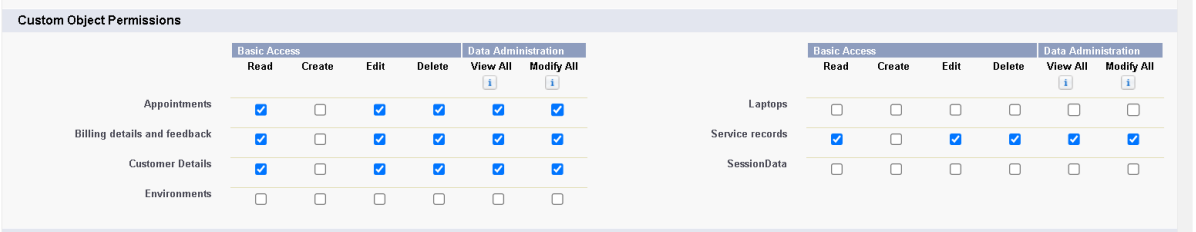


2. While still on the profile page, then click Edit.



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

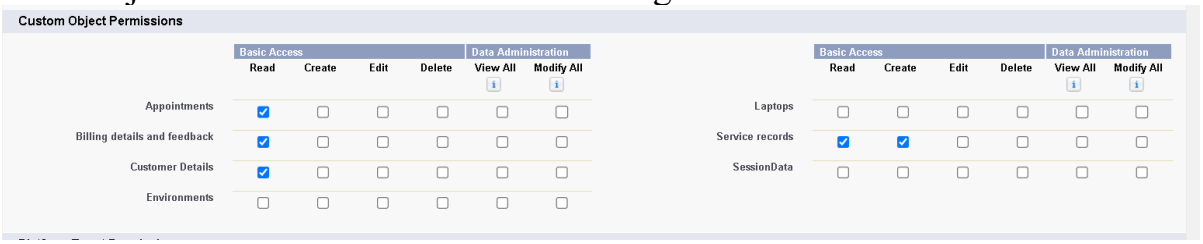
4. Scroll down to Custom Object Permissions and Give access permissions for Appointments,Billing details and feedback , service records and customer details objects as mentioned in the below diagram.



5. Changing the session times out after should be " 8 hours of inactivity".
6. Change the password policies as mentioned :
7. User passwords expire in should be " never expires ".
8. Minimum password length should be " 8 ", and click save.

Activity 2: Create a salesperson profile:

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Salesforce Platform User) >> enter profile name (sales person) >> Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the GArage management.
4. 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.



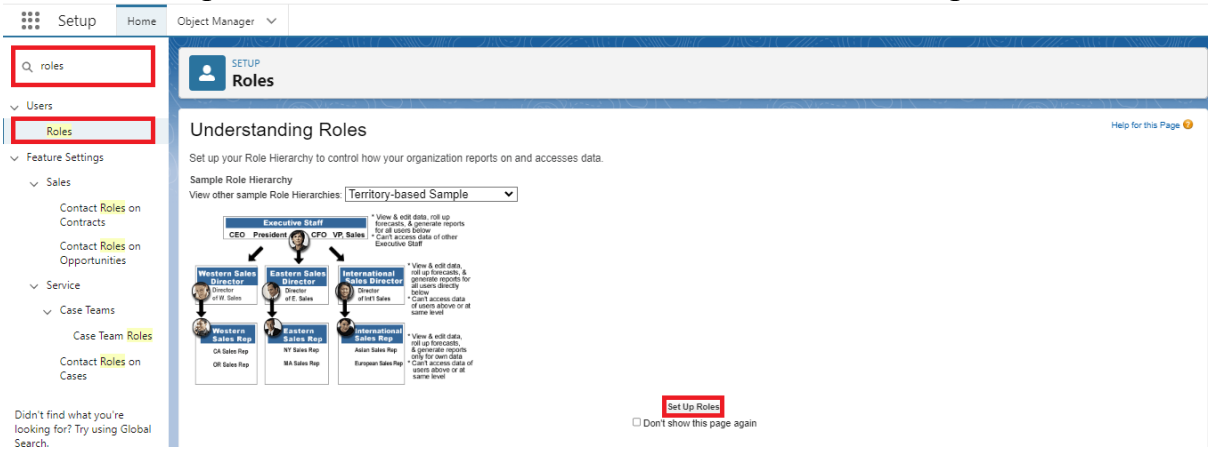
5. And click save.

ROLE & Role HIERARCHY:

Activity 1: Creating Manager Role:

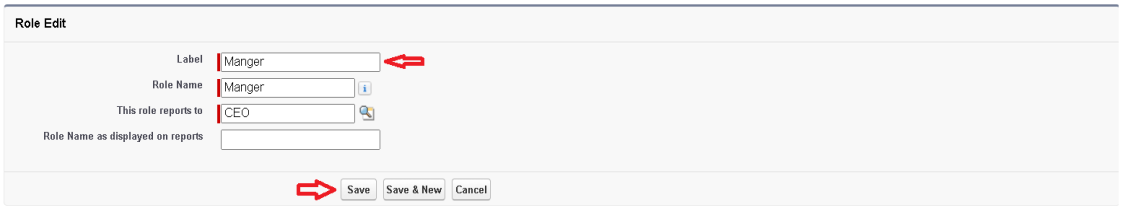
Creating Manager Role:

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.

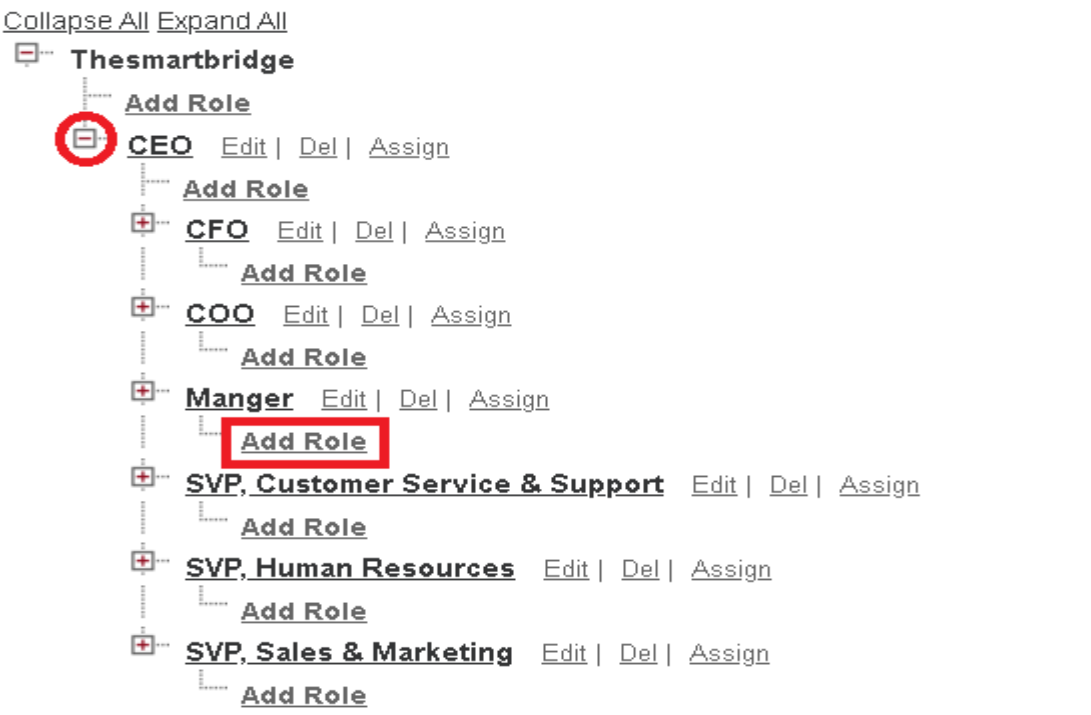
3. Give Label as “Manager” and Role name gets auto populated. Then click on Save.



Activity 2: Creating another roles:

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.

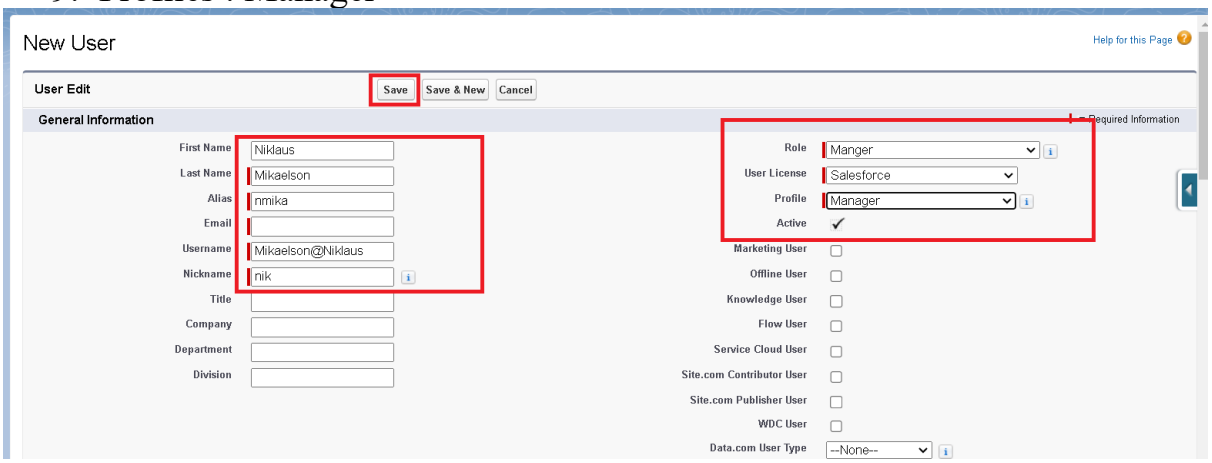


3. Give Label as “sales person” and Role name gets auto populated. Then click on Save.

USERS:

Activity 1: Create Users:

1. Go to setup >> type users in quick find box >> select users >> click New user.
2. 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



3. Save.

Activity 2: Creating Another User:

Repeat the steps and create another user using

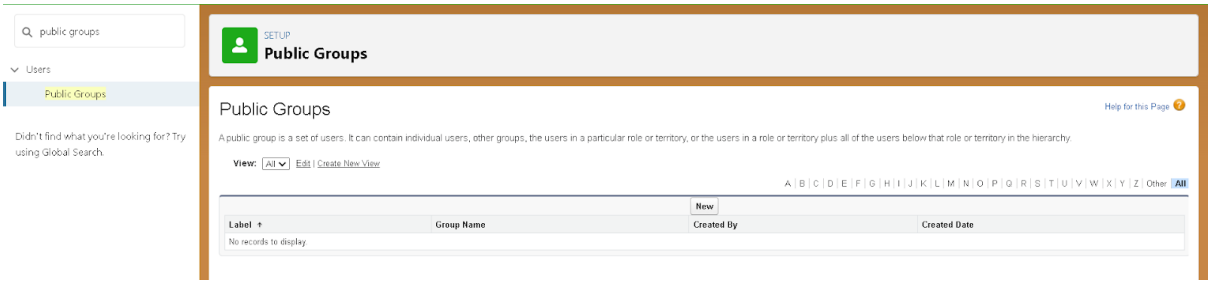
a. Role : sales person
b. User licence : Salesforce Platform
c. Profile : sales person

Note : create atleast 3 users with these permissions.

PUBLIC GROUPS:

Activity 1: Creating new public groups:

1. Go to setup >> type users in quick find box >> select public groups >> click New.

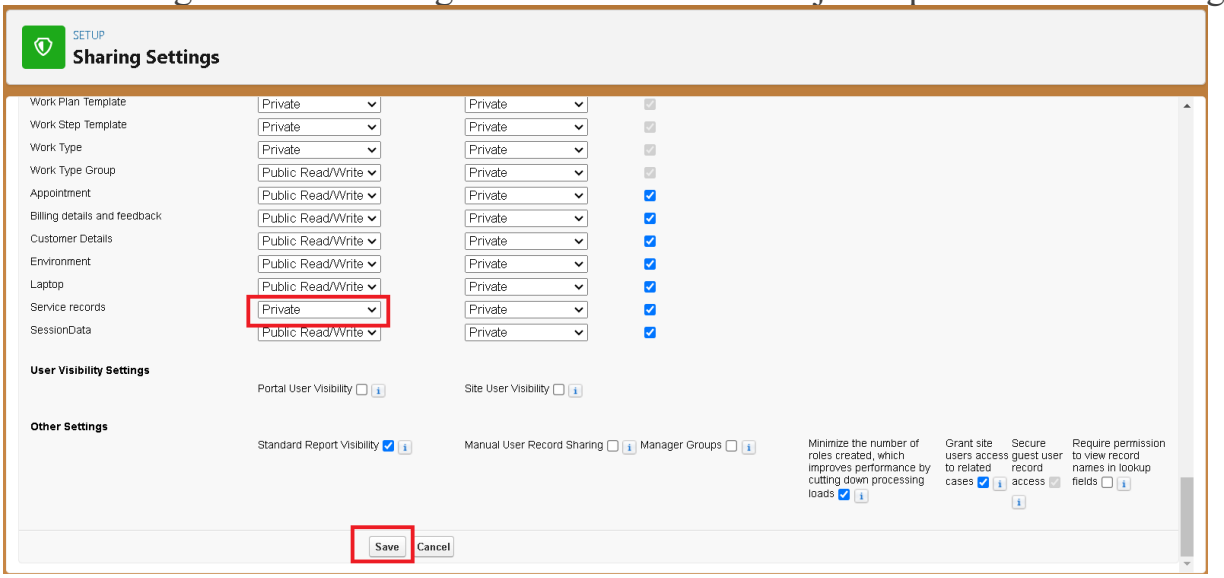


2. Give the Label as “sales team”.
3. Group name is autopopulated.
4. Search for Roles.
5. In Available Members select Sales person and click on add it will be moved to selected member.
6. Click on save.

SHARING SETTINGS:

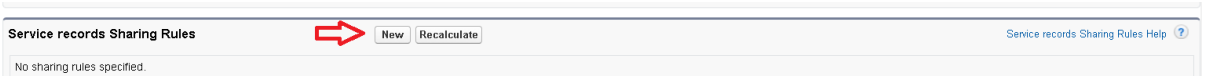
Activity 1: Create the sharing settings:

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.

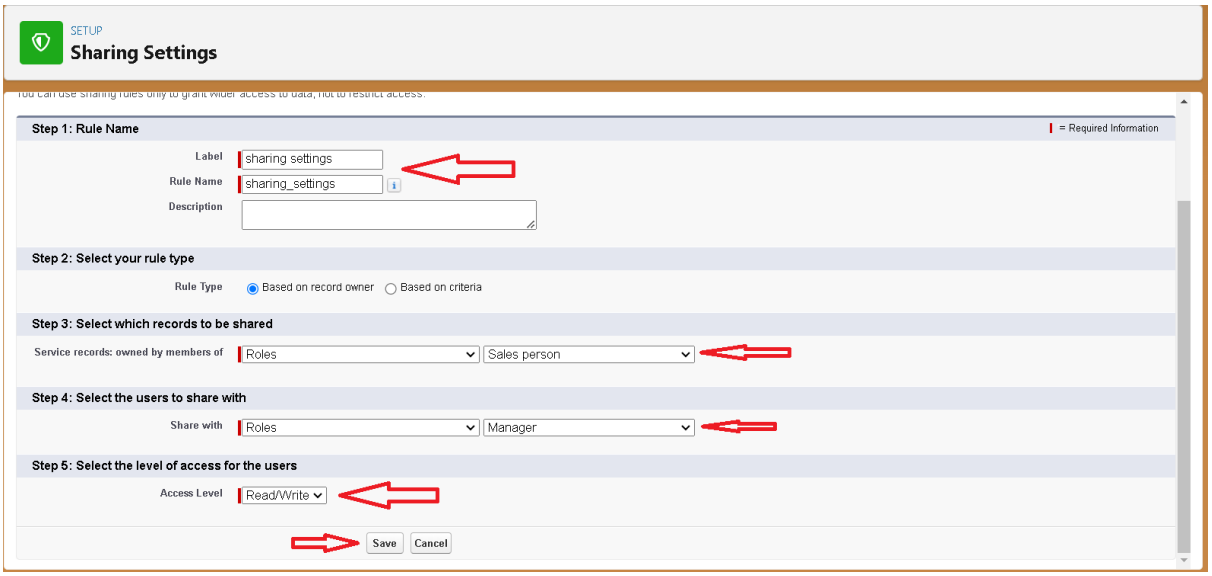


3. Click on save and refresh.

4. Scroll down a bit, Click new on Service records sharing Rules.
5.



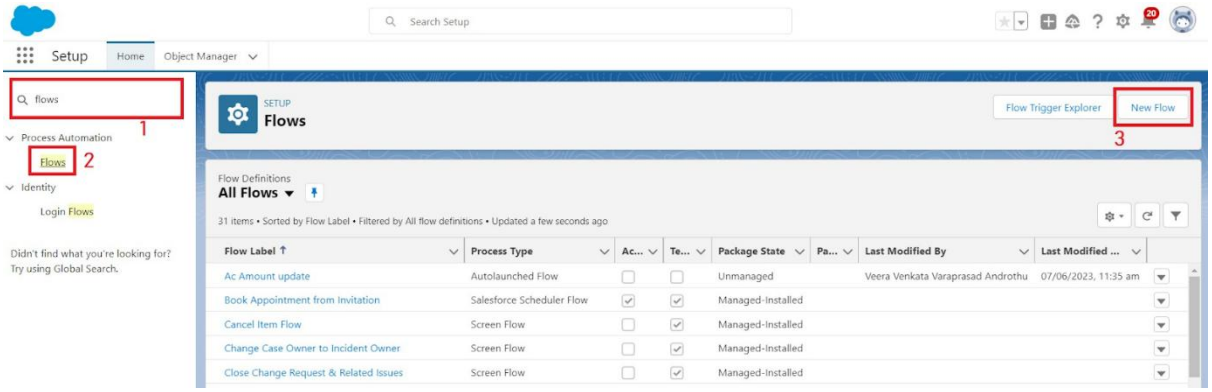
6. Give the Label name as " Sharing setting"
7. Rule name is auto populated.
8. In step 3 : Select which records to be shared, members of " Roles " >> " Sales person"
9. In step 4: share with, select " Roles " >> " Manager "
10. In step 5 : Change the access level to " Read / write ".
11. Click on save.



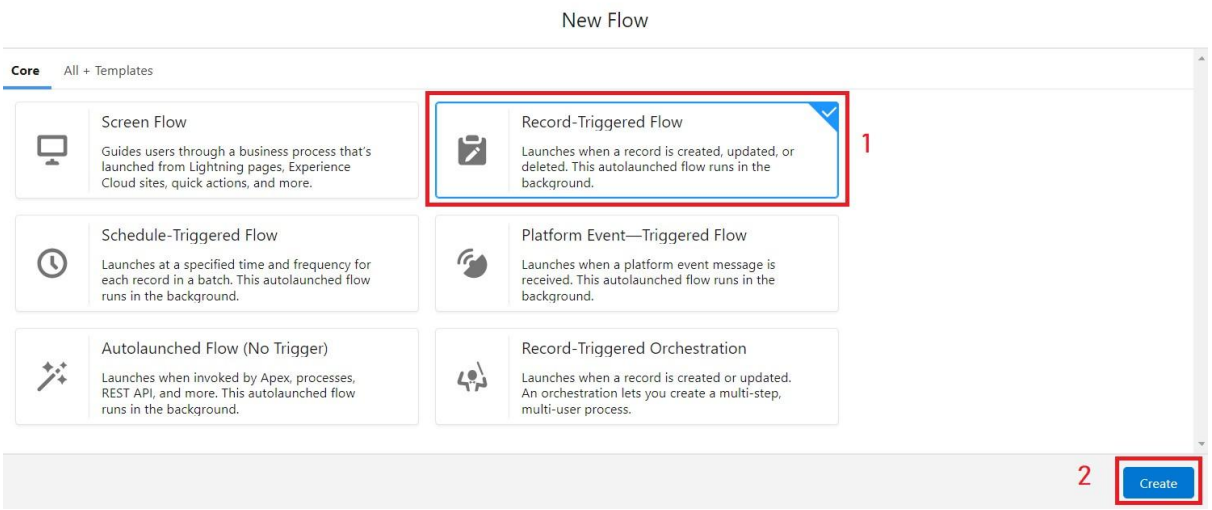
FLOWS:

Activity 1: Create a flow:

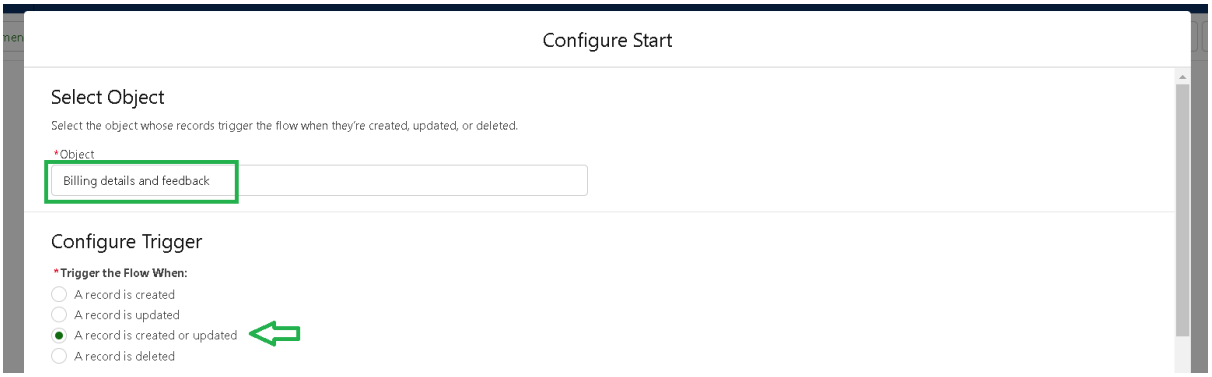
1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.

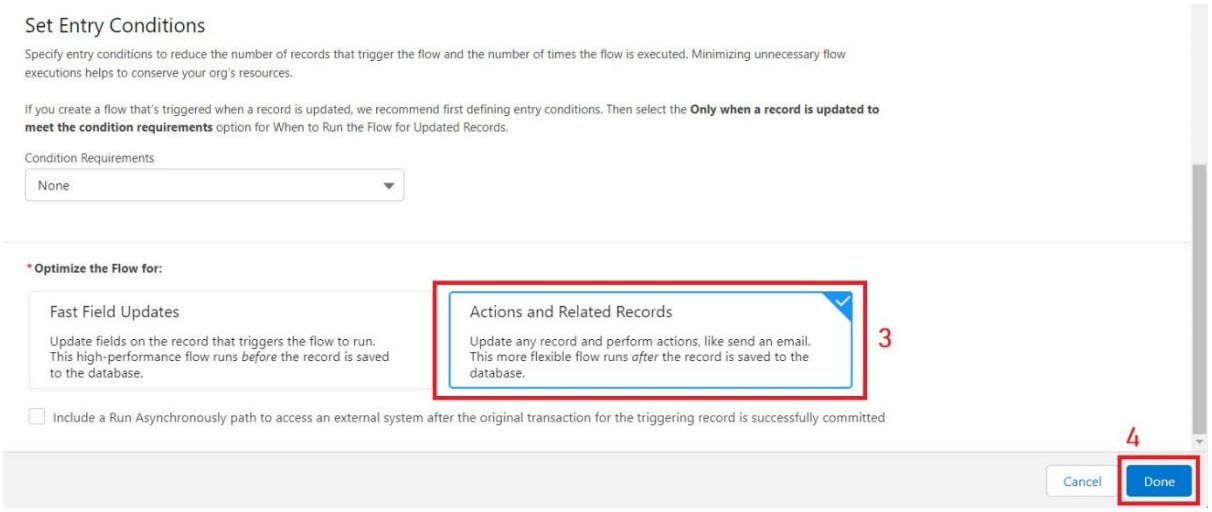


2. Select the Record-triggered flow and Click on Create.
3.

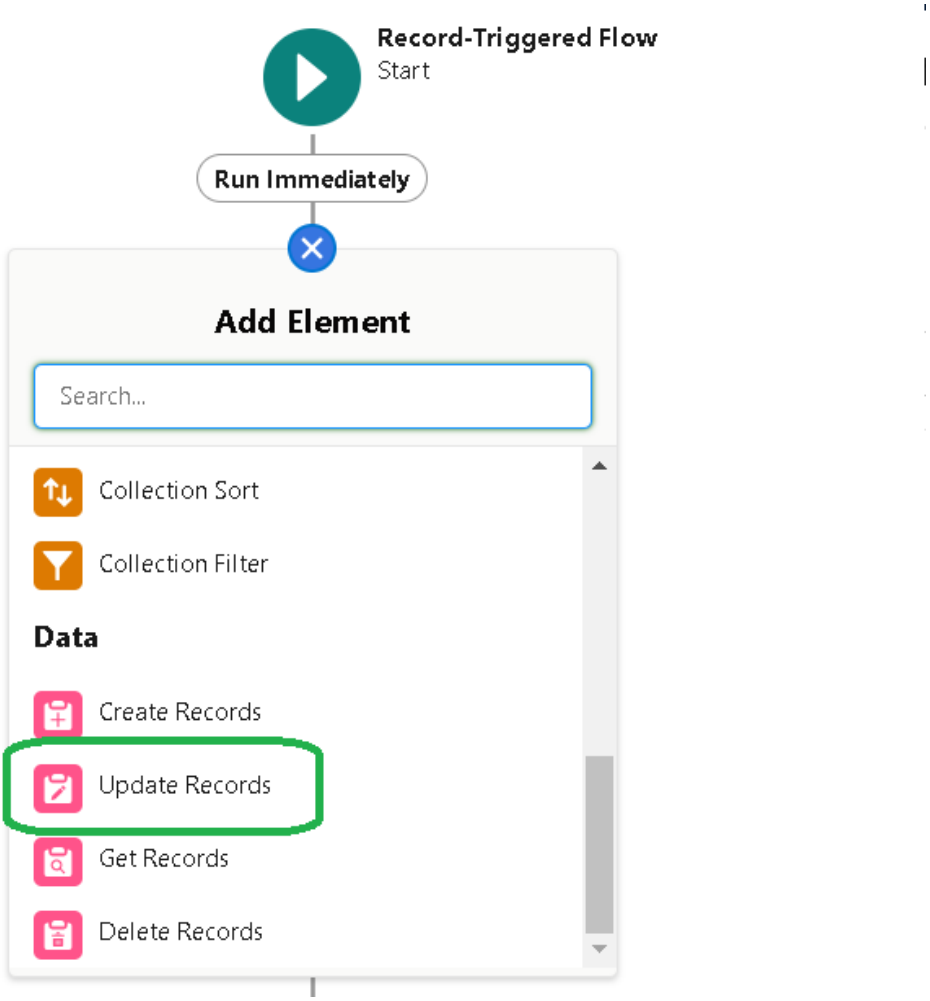


3. Select the Object as "Billing details and feedback"in the Drop down list.
4. Select the Trigger Flow when: "A record is Created or Updated".
5. Select the Optimize the flow for: "Actions and Related Records" and Click on Done.
6.

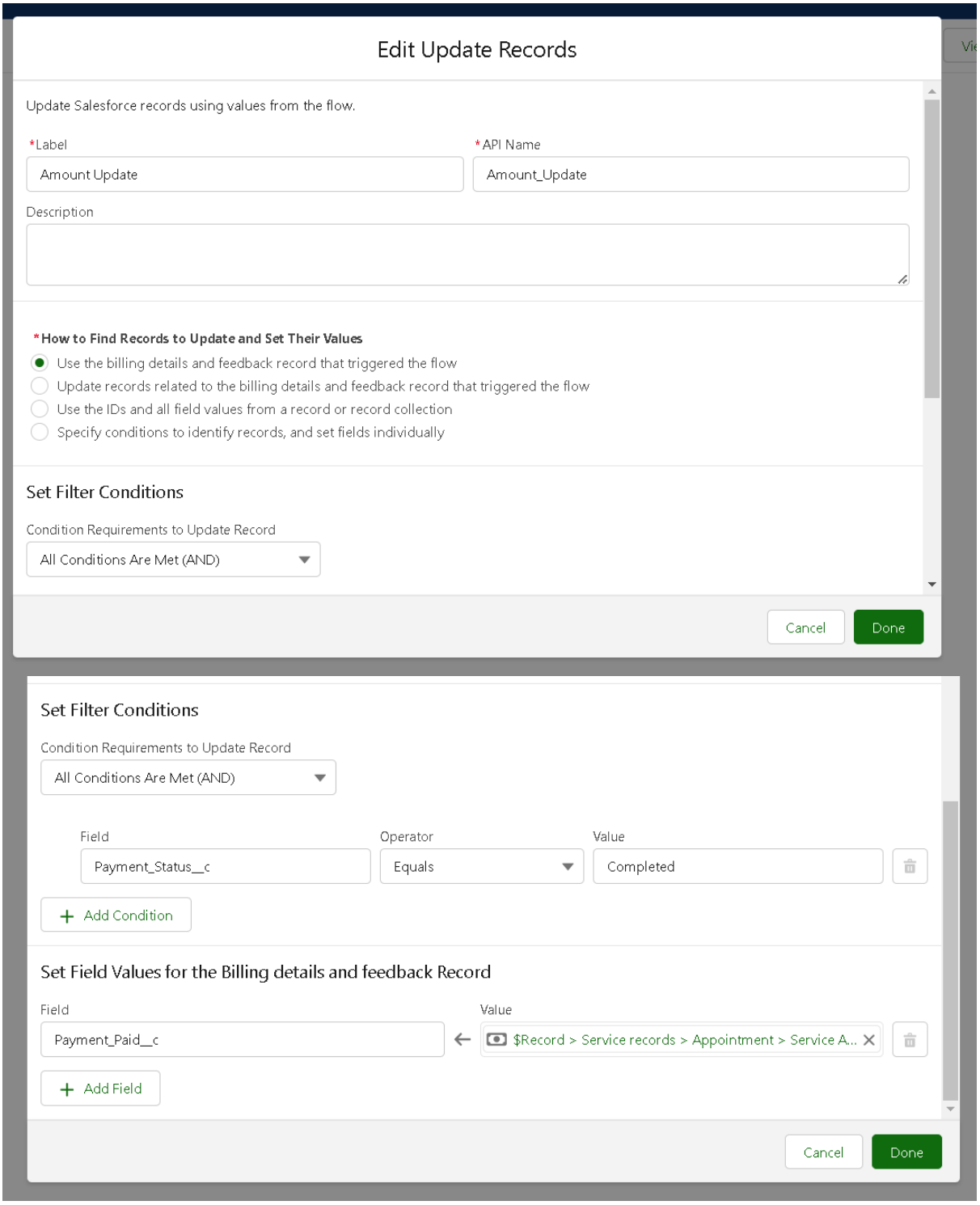




6. Under the Record-triggered Flow Click on "+" Symbol and In the Drop down List select the "Update records Element".

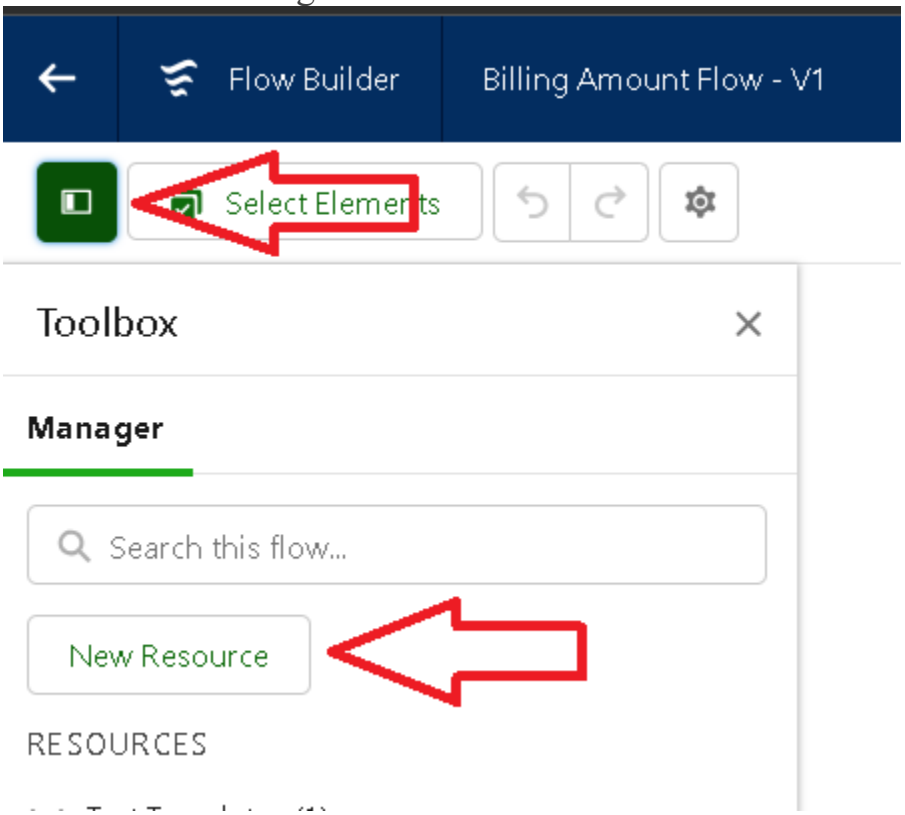


7. Give the Label Name : Amount Update
8. Api name : is auto populated



9. Set a filter condition : All Conditions are met(AND)
10. Field : Payment_Status__c
11. Operator : Equals
12. Value : Completed
13. And Set Field Values for the Billing details and feedback Record
14. Field : Payment_Paid__c
15. Value : {!$Record.Service_records__r.Appointment__r.Service_Amount__c}
16. Click On Done.

17. Before creating another Element. Create a New Resource form Toolbox form top left.



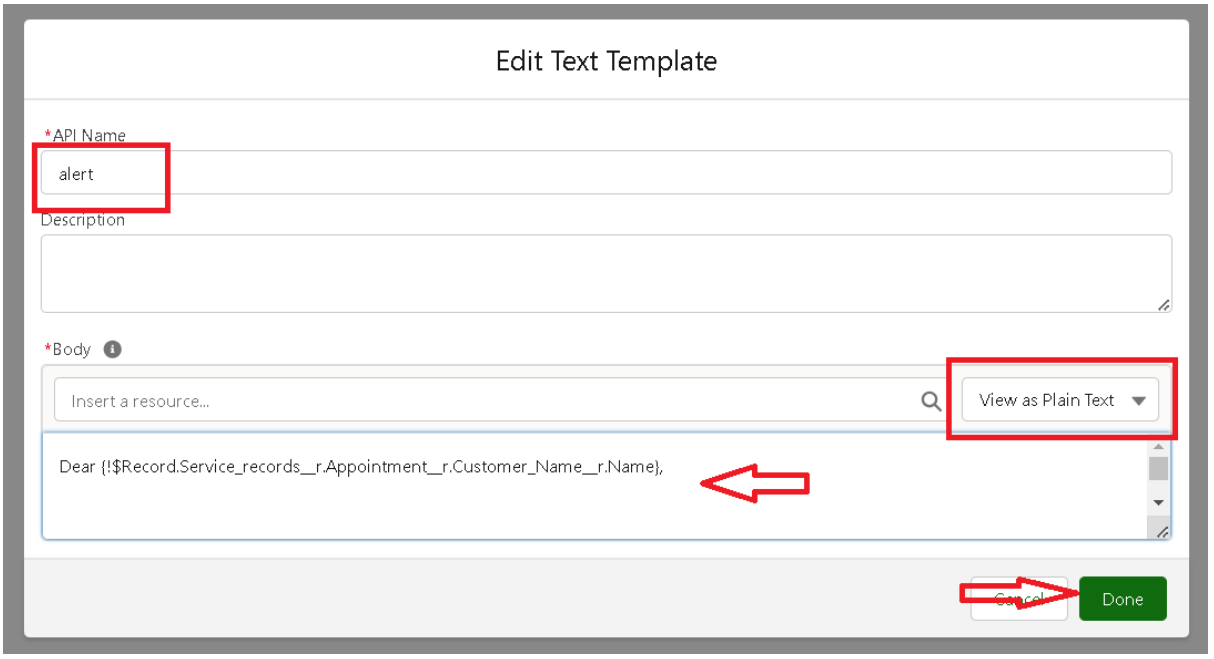
18. Click on the New Resource, And select Variable.
19. Select the resource type as text template.
20. Enter the API name as " alert".
21. Change the view as Rich Text ? View to Plain Text.
22. In body field paste the syntax that given below.

Dear {!$Record.Service_records__r.Appointment__r.Customer_Name__r.Name},

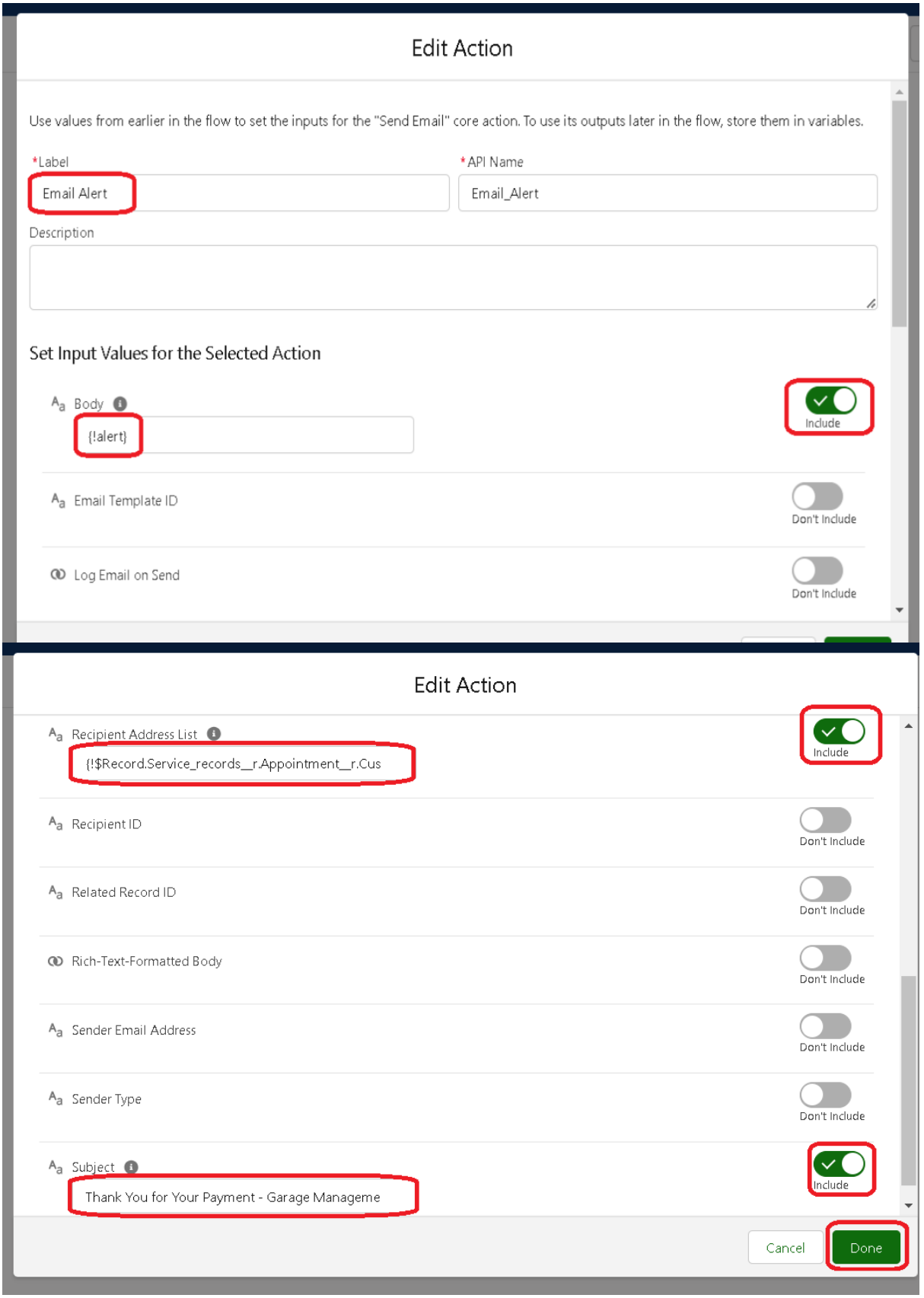
I hope this message finds you well. I wanted to take a moment to express my sincere gratitude for your recent payment for the services provided by our garage management team. Your prompt payment is greatly appreciated, and it helps us continue to provide top-notch services to you and all our valued customers.

Amount paid : {!$Record.Payment_Paid__c}
Thank you for Coming .

23. Click done.

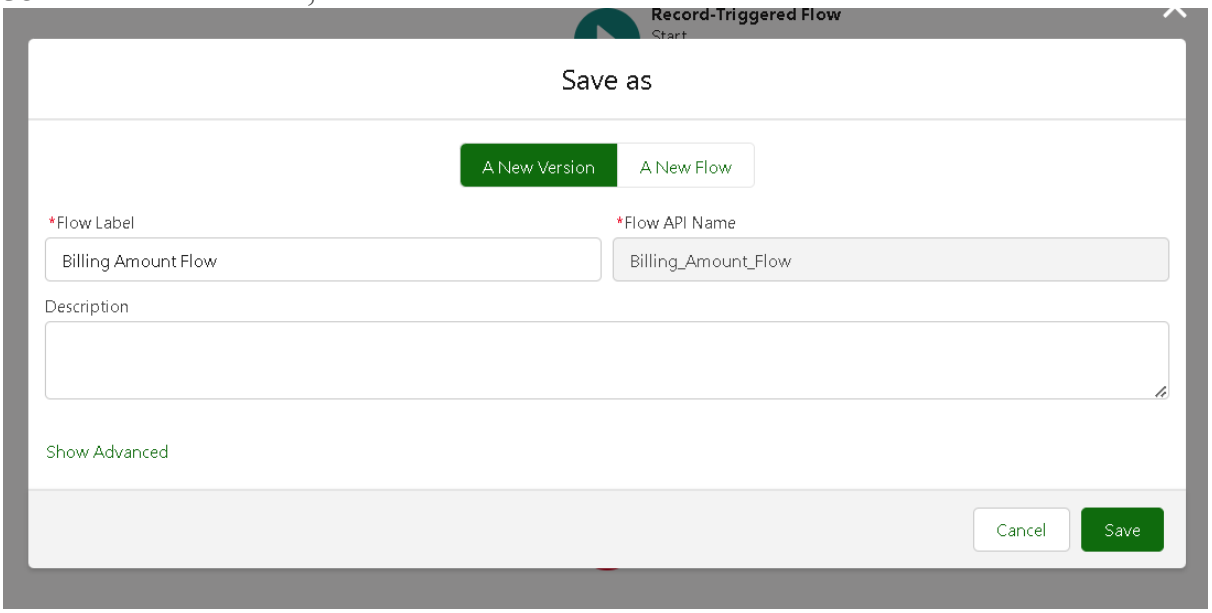


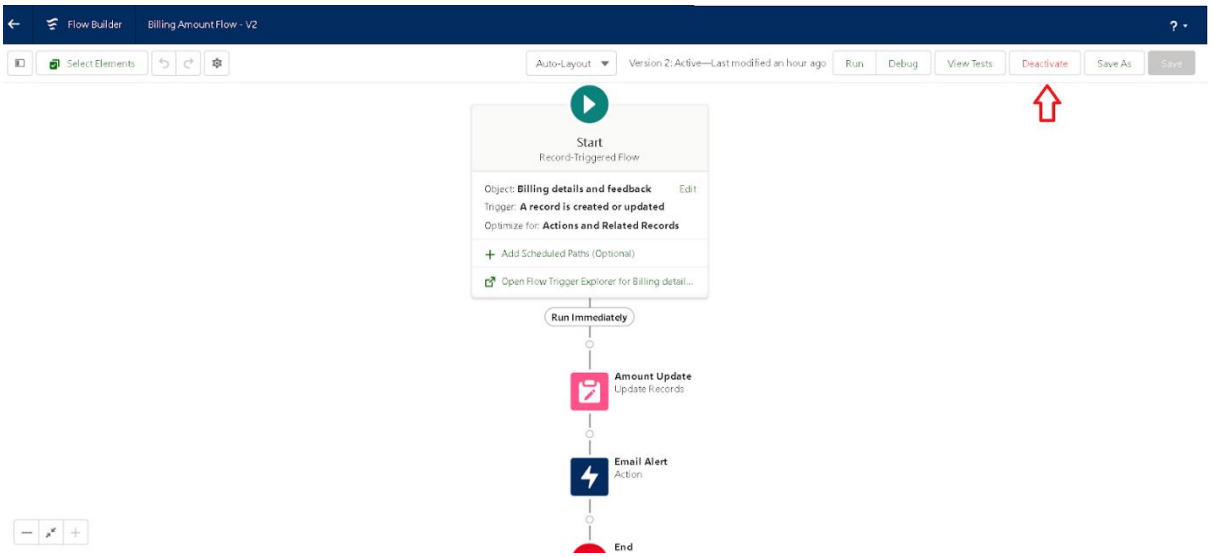
24. Now Click on Add Element , select Action.
25. Their action bar will be opened in that search for " send email " and click on it.
26. Give the label name as " Email Alert"
27. API name will be auto populated.
28. Enable the body in set input values for the selected action.
29. Select the text template that created , Body : {!alert}
30. Include recipient address list select the email form the record.
31. RecipientAddressList: {!$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail__c}
32. Include subject as " Thank You for Your Payment - Garage Management".
33. Click done.



34. Click on save. Give the Flow label , Flow Api name will be autopopulated.

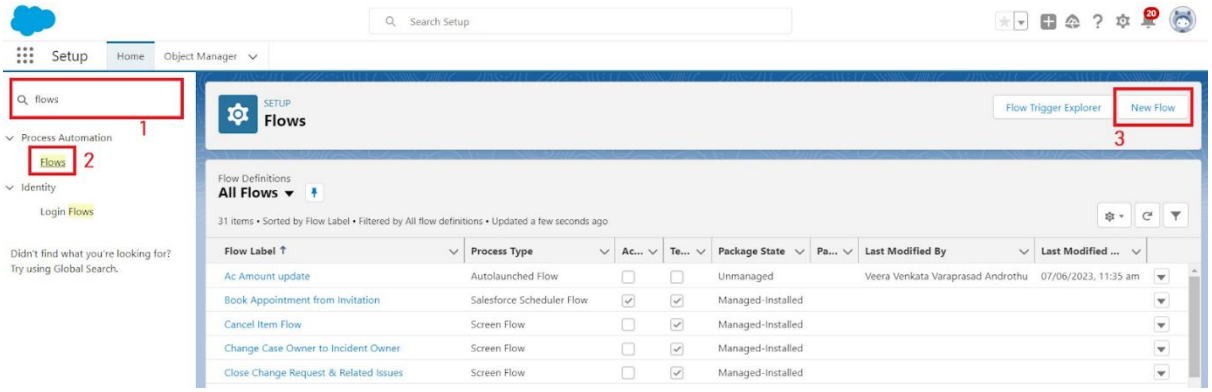
35. And click save, and click on activate.



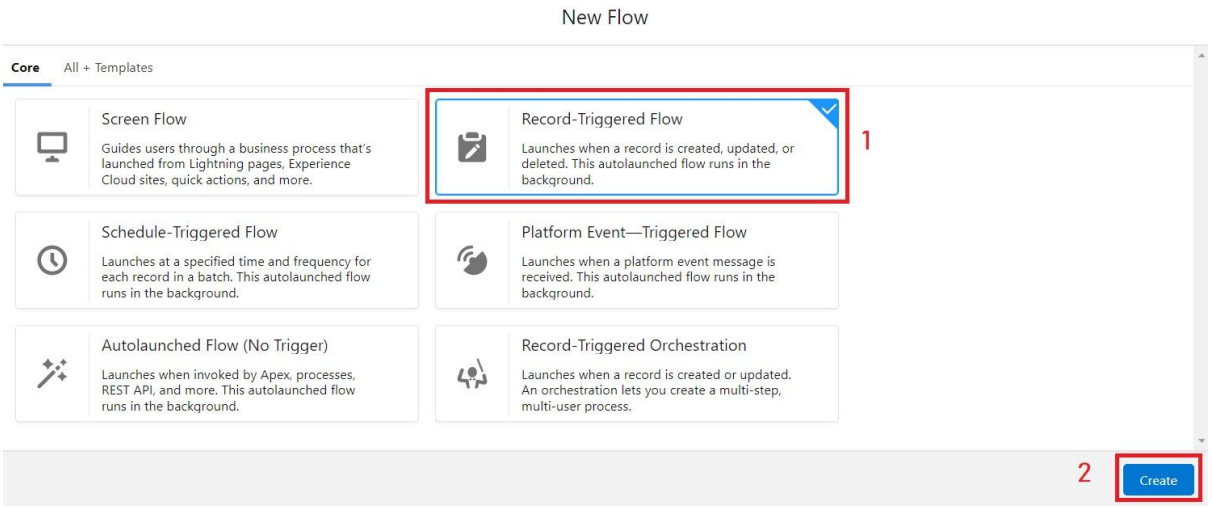


Activity 2: Create another flow:

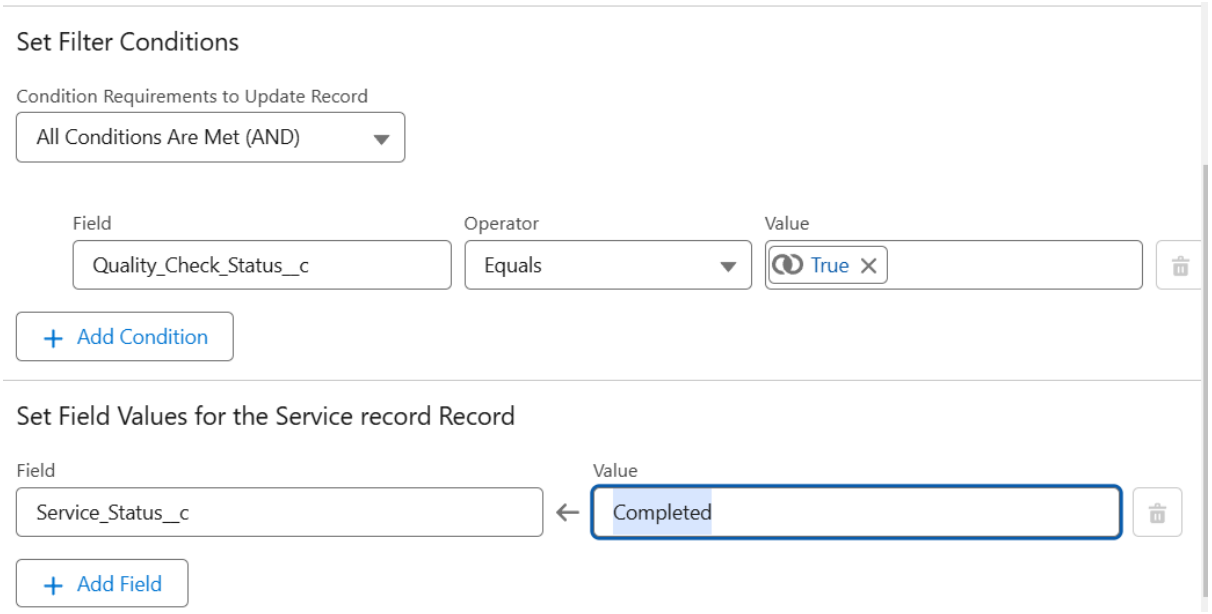
1. Go to setup ? type Flow in quick find box ? Click on the Flow and Select the New Flow.



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



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" and Click on Done.
6. Under the Record-triggered Flow Click on "+" Symbol and In the Drop down List select the "Update records Element".
7. Set a filter condition : All Conditions are met(AND)
8. Field : Quality_Check_Status__c
9. Operator : Equals
10. Value : True
11. And Set Field Values for the Billing details and feedback Record
12. Field : Service_Status__c
13. Value : Completed

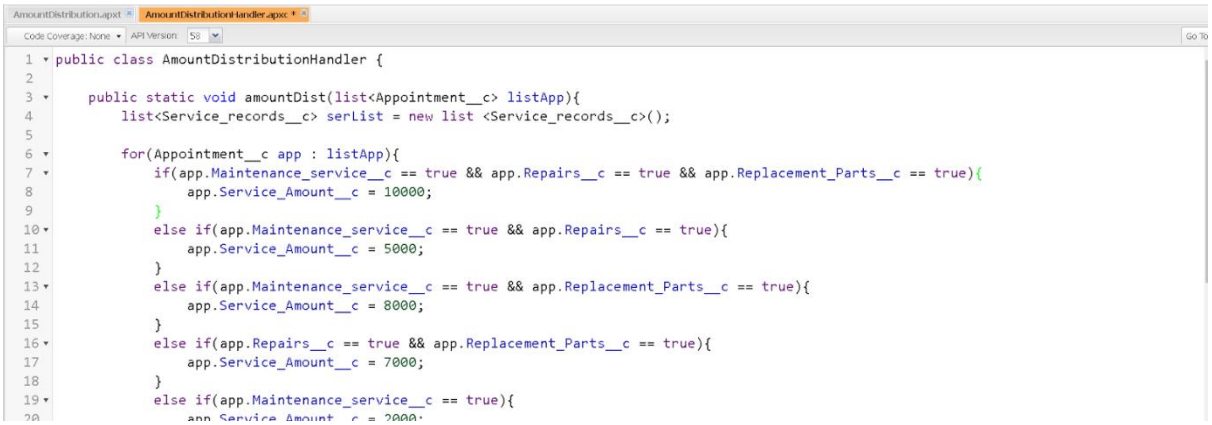


14. Click On Done.
15. Click on save
16. Given the Flow label as Update Service Status , Flow Api name will be auto populated.
17. And click save, and click on activate

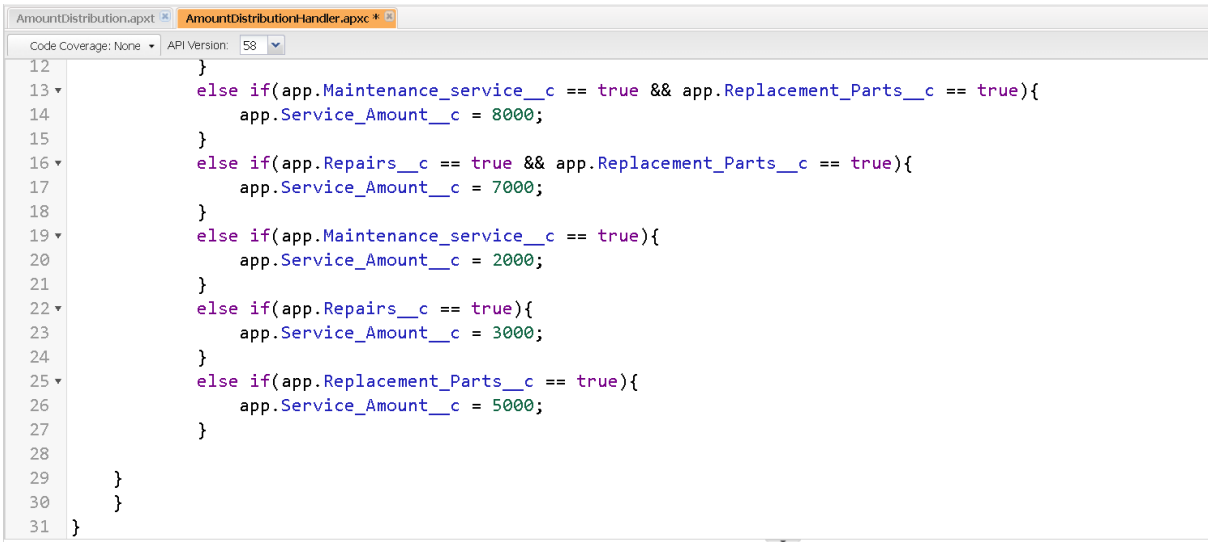
APEX TRIGGER:

Activity 1: Create a apex handler:

1. Login to the respective trailhead account and navigate to the gear icon in the top right corner.
2. Click on the Developer console. Now you will see a new console window.
3. In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.
4. Name the class as "AmountDistributionHandler ".



```
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;
```



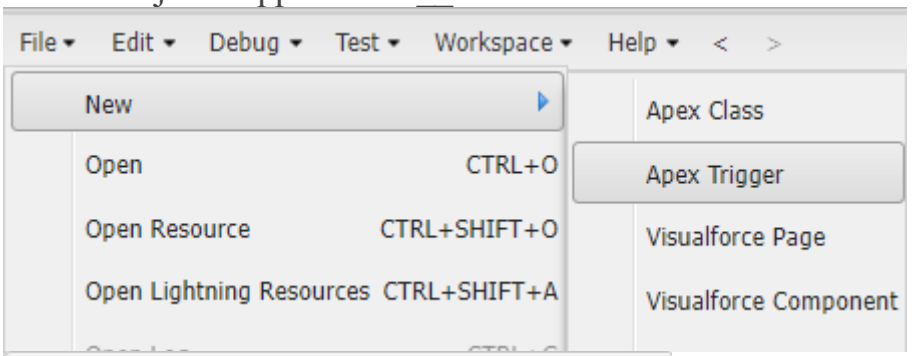
Code:

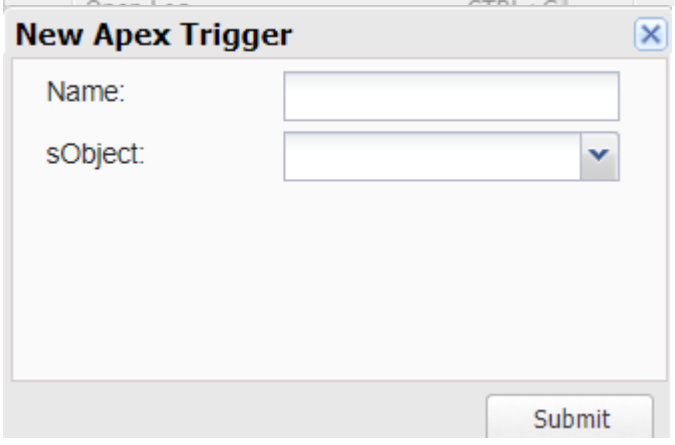
```
public class AmountDistributionHandler {

   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 :
How to create a new trigger :

1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on File menu in the tool bar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.
5. Name : AmountDistribution
6. sObject : Appointment__c

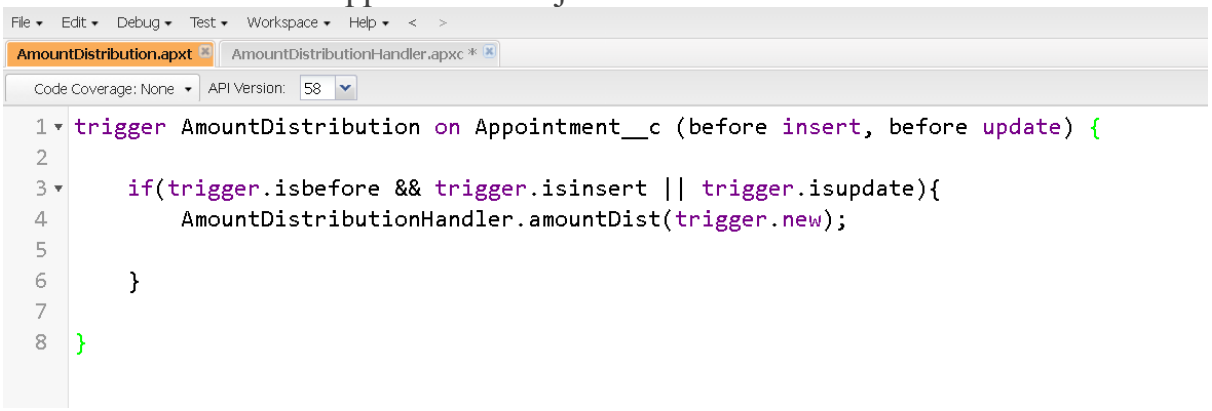




Syntax For creating trigger :
The syntax for creating trigger is :
Trigger [trigger name] on [object name](Before/After event)
{
}

In this project , trigger is called whenever the particular records sum exceed the threshold i.e minimum business requirement value. Then the code in the trigger will get executed.

1. Handler for the Appointment Object



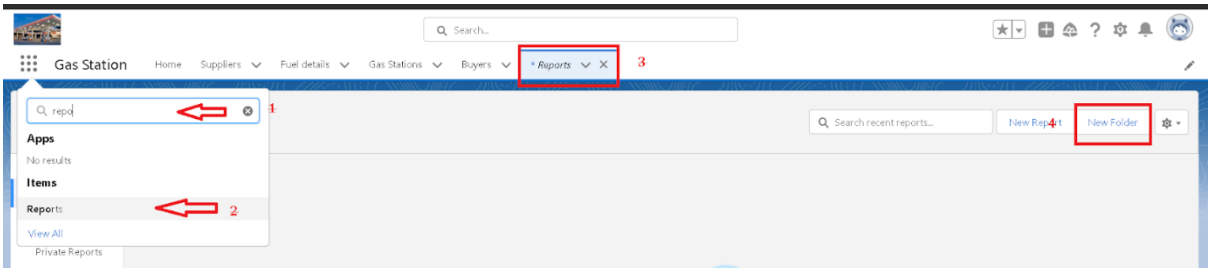
Code:

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

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

   }

}
```

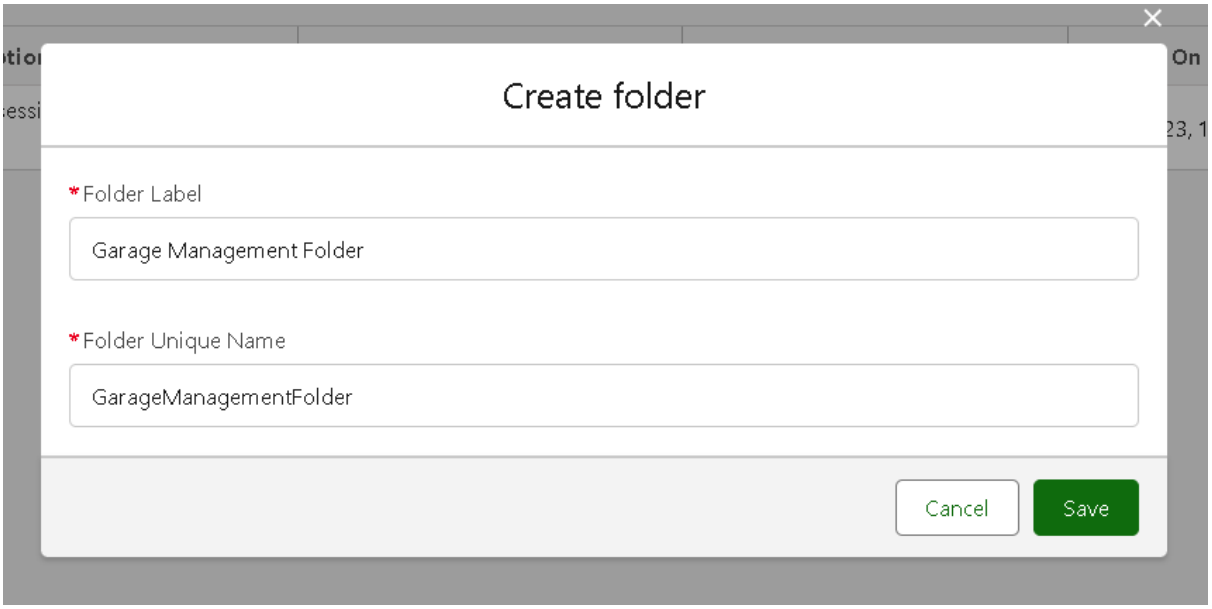
REPORTS:

Activity 1: Create a report folder:

1. Click on the app launcher and search for reports.
2. Click on the report tab, click on new folder.

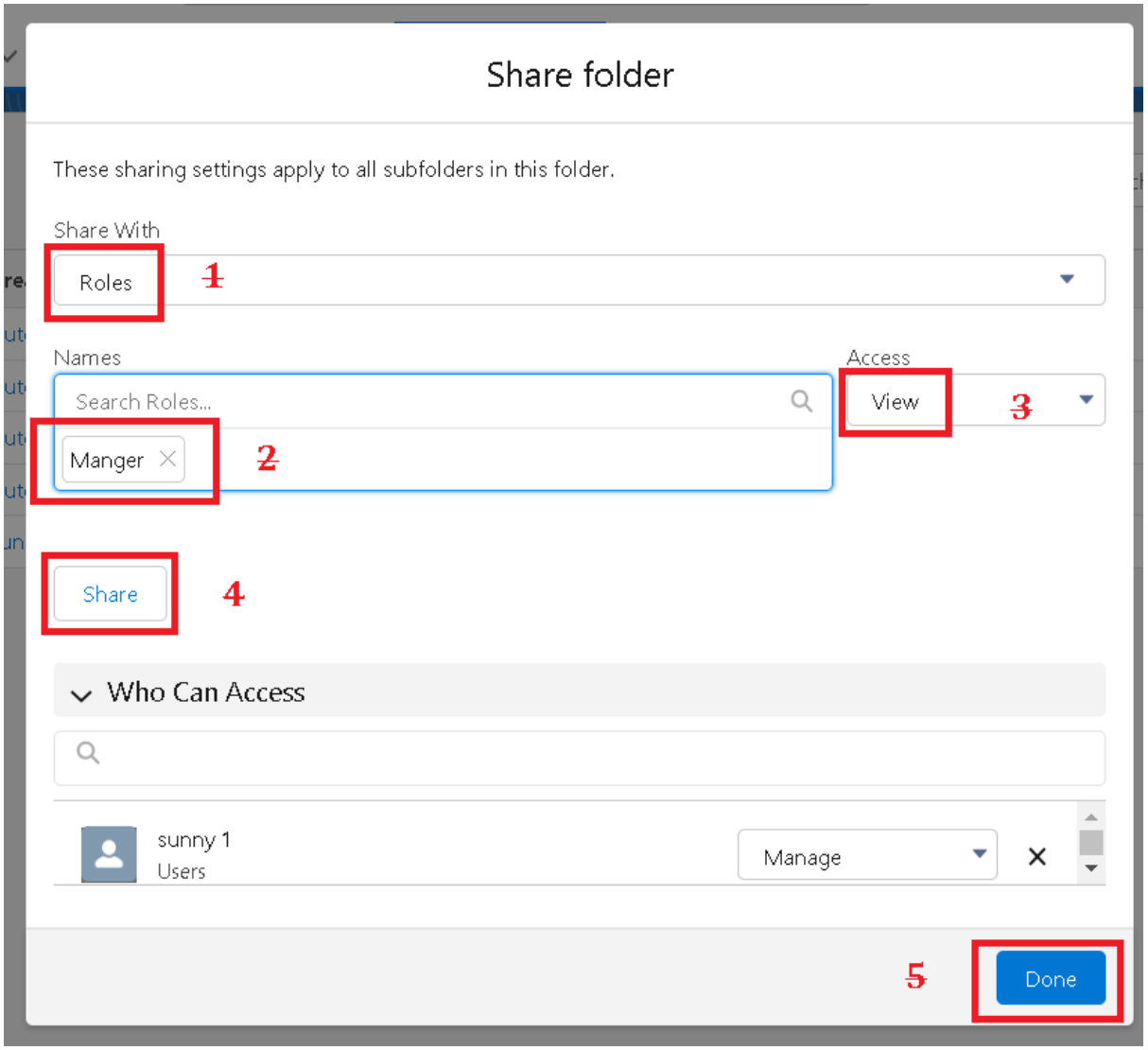


3. Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.
4. Click save.



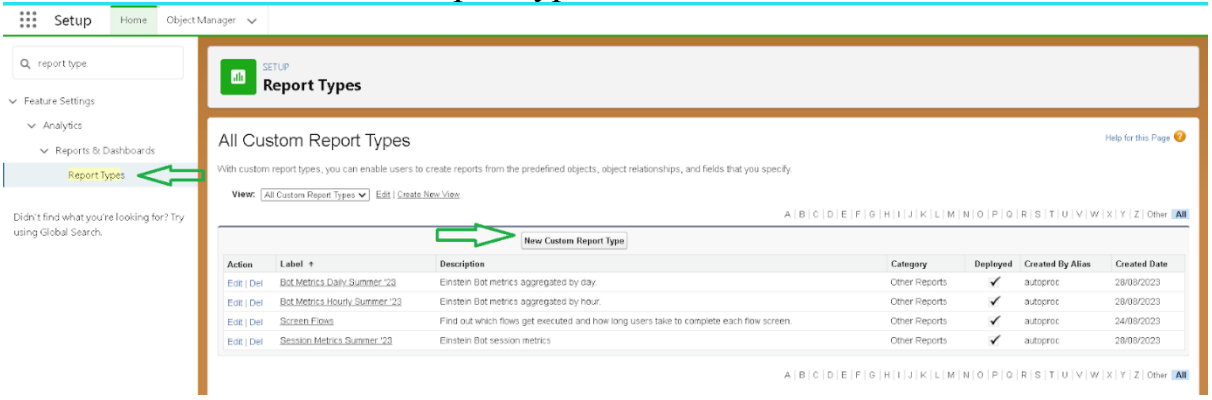
Activity 2: Sharing a report folder:

1. Go to the app >> click on the reports tab.
2. Click on the All folder , click on the Drop down arrow for Garage Management folder, and Click on share.
3. Select the share with as “roles”, in name field search for “manager”, give “view” as access for that role.
4. Then click share, and click on Done.



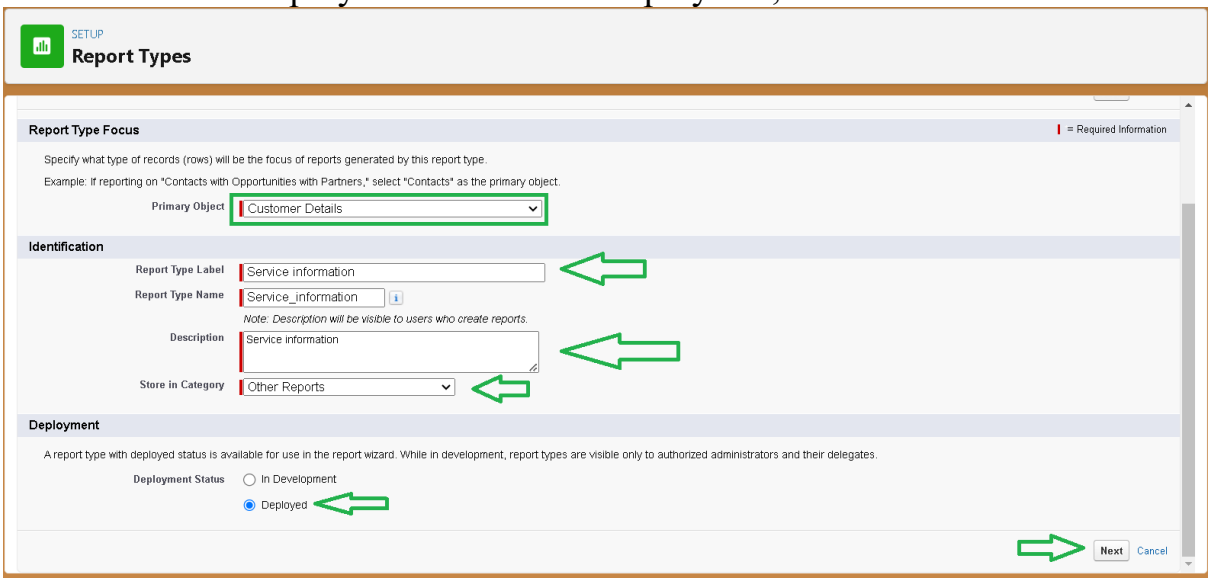
Activity 3: Create a report type:

1. Go to setup >> type users in quick find box >> select Report Type >> click on Continue.
2. Click on new custom report type.

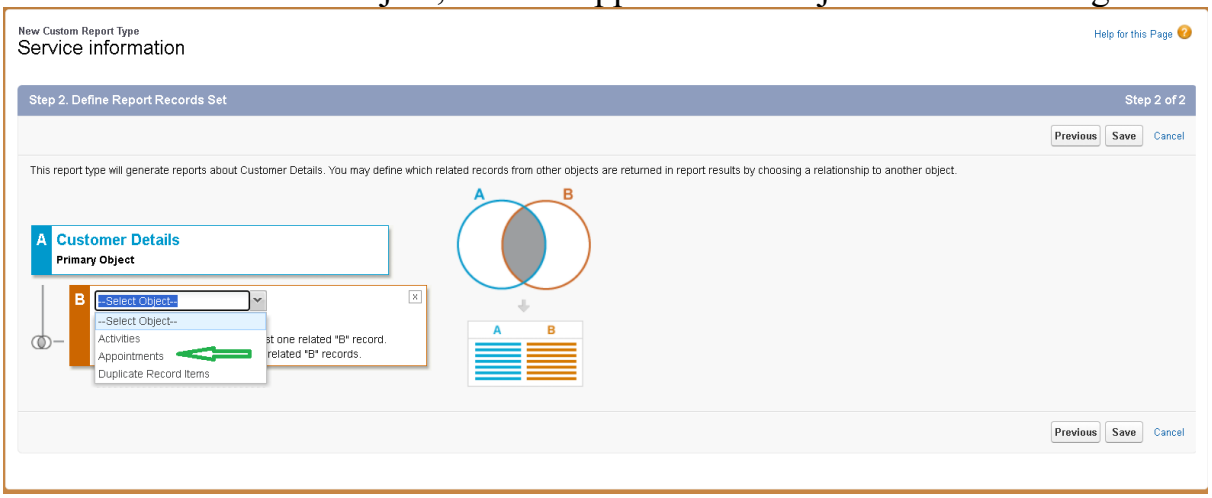


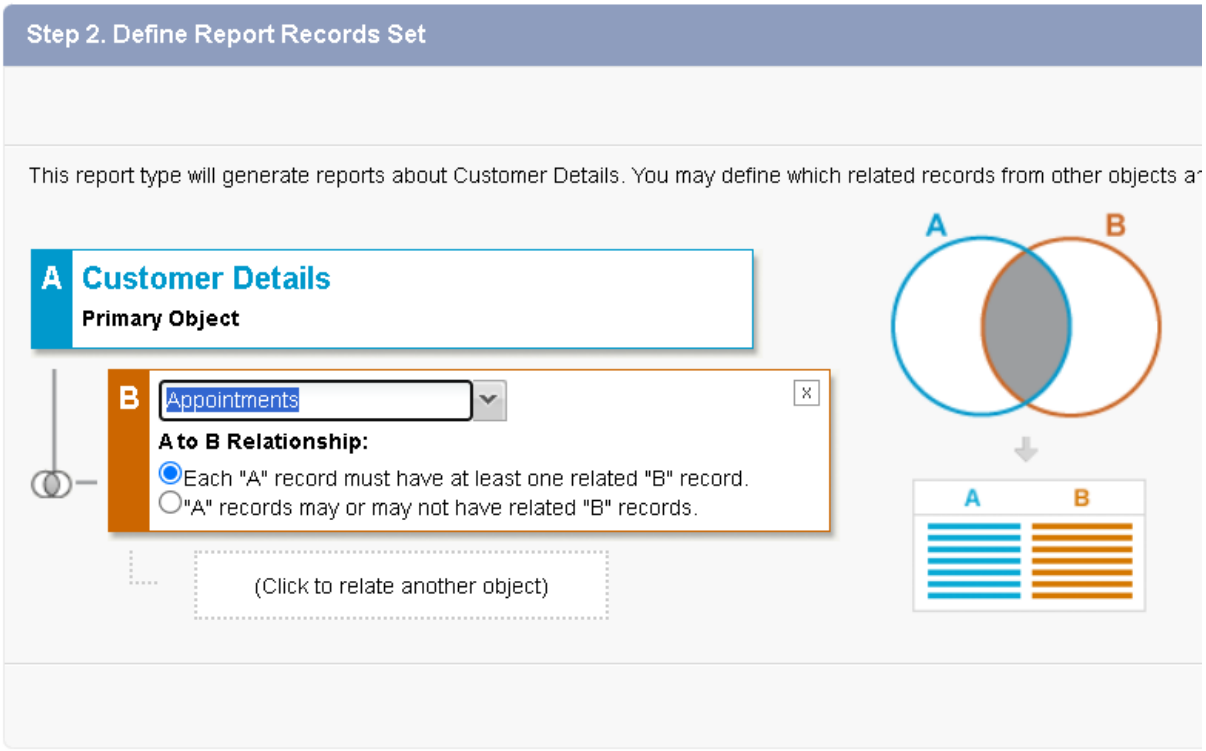
3. Select the Primary object as " Customer details" .

4. Give the Report type Label as " Service information "
5. Report type Name is autopopulated.
6. Keep the Description as same.
7. Select Store in Category as " other Reports "
8. Select the deployment status as " Depolyed ", click on Next.

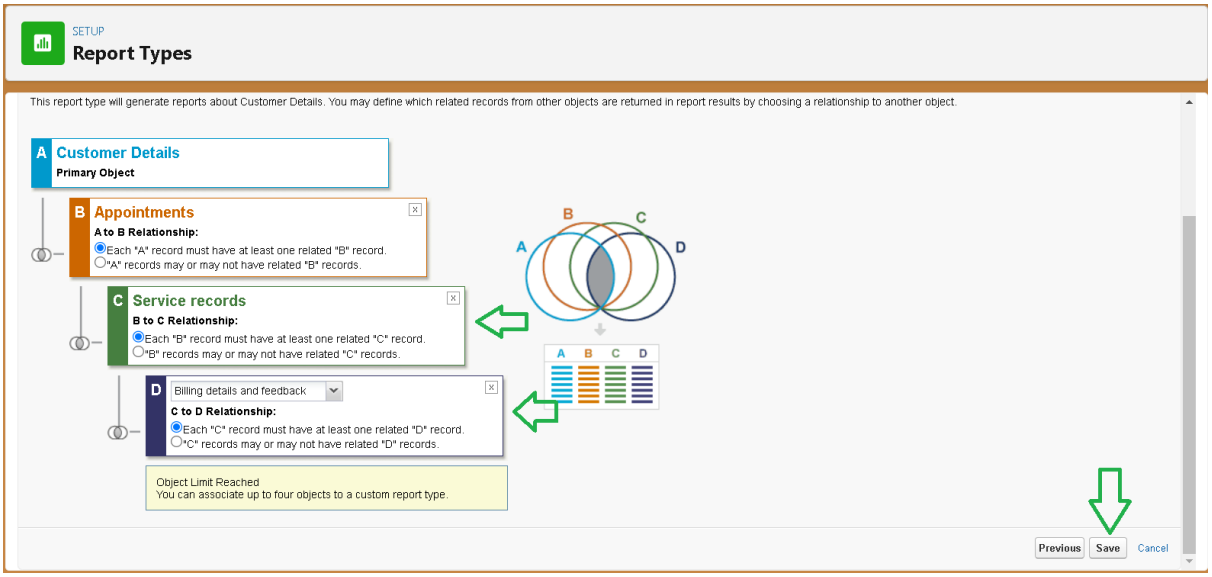


9. now , Click on Related object box.
10. Click on Select Object, choose Appointment Object as shown in fig.



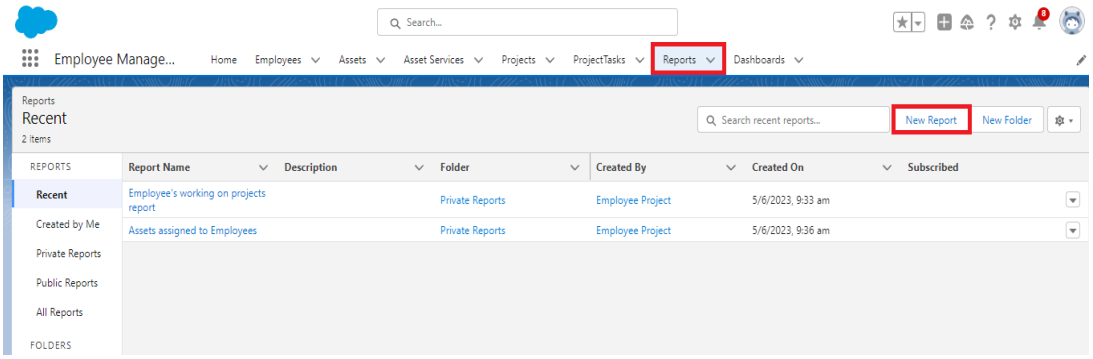


11. Again Click to relate another object.
12. And select the related object as " service records".
13. Repeat the process and select the related object as " Billing details and feedback".
14. And click on save.

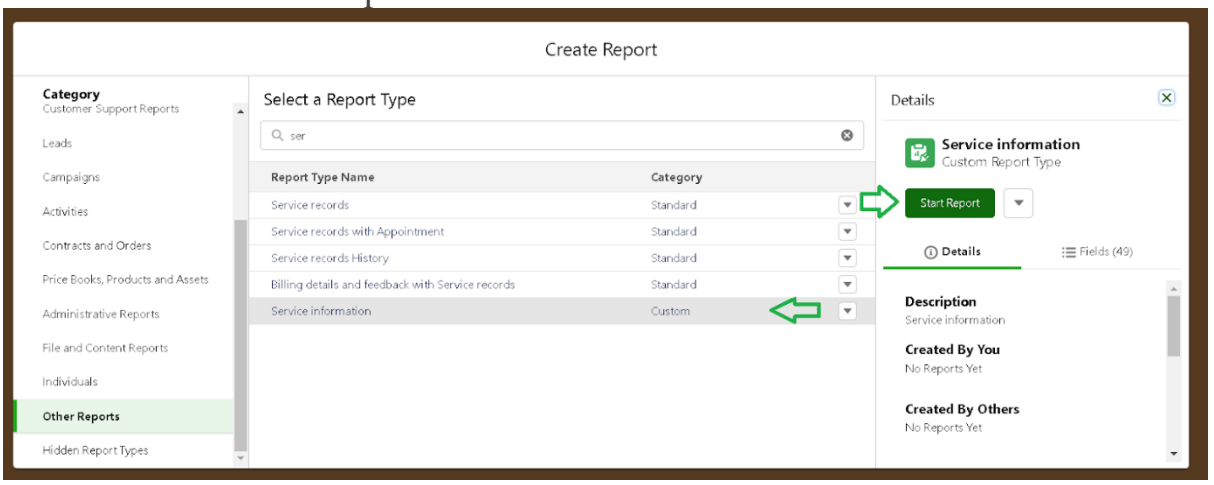


Activity 4: Create a report:

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.

a. Customer name
b. Appointment Date
c. Service Status
d. Payment paid

5. Remove the unnecessary fields.
6. Select the fields that mentioned below in GROUP ROWS section.

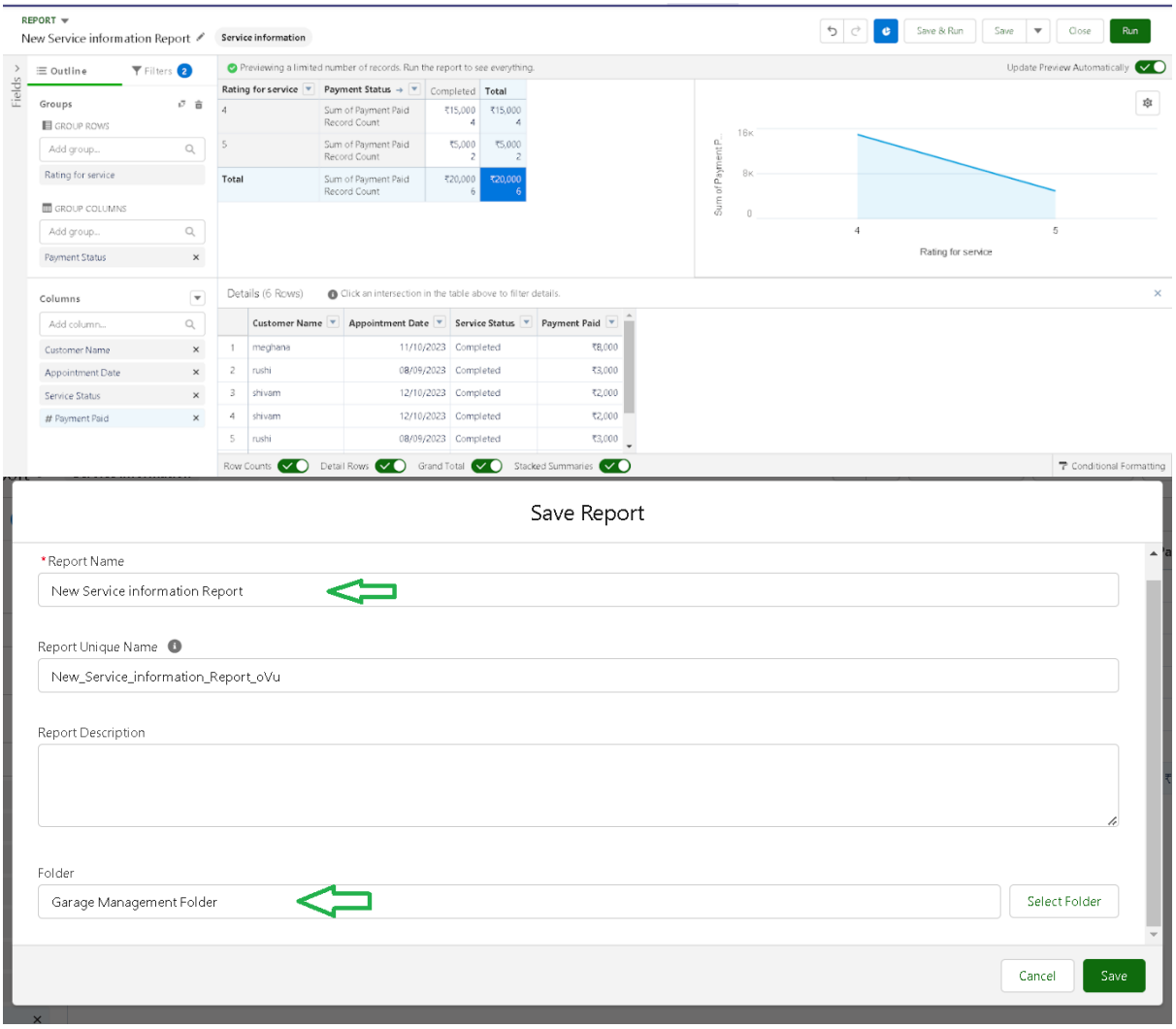
a. Rating for Service

7. Select the fields that mentioned below in GROUP ROWS section.

a. Payment Status

8. Click on Add Chart , Select the Line Chart.

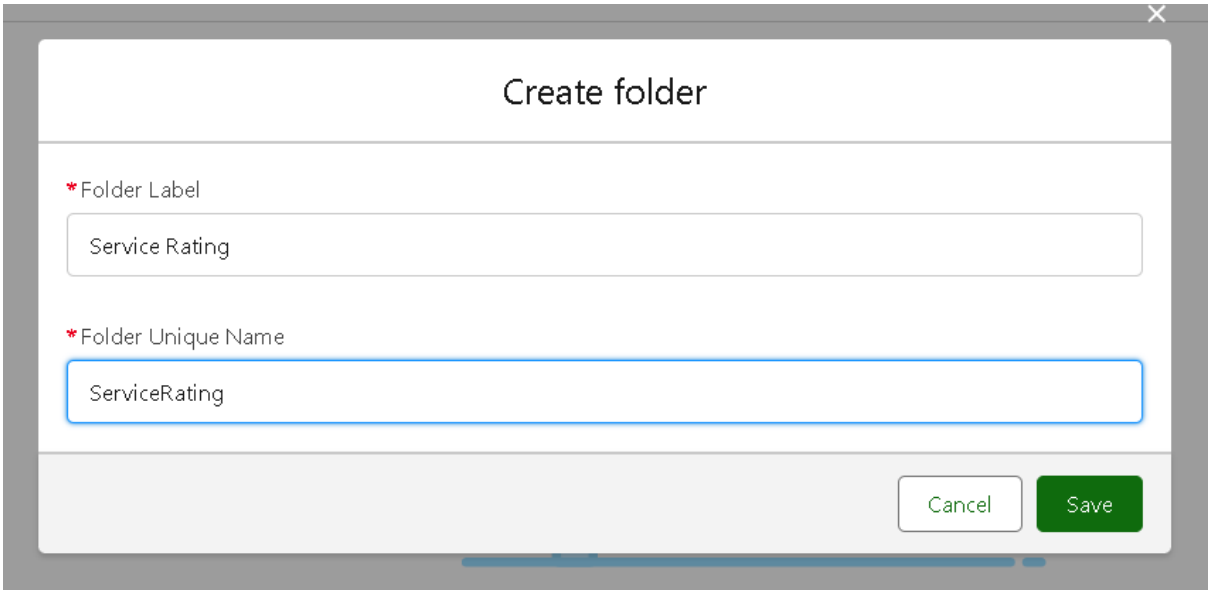
9. Click on save, Give the report Name : New Service information Report
10. Report unique Name is auto populated.
11. Select the folder the created and Click on save.



DASHBOARDS:

Activity 1: Create dashboard folder:

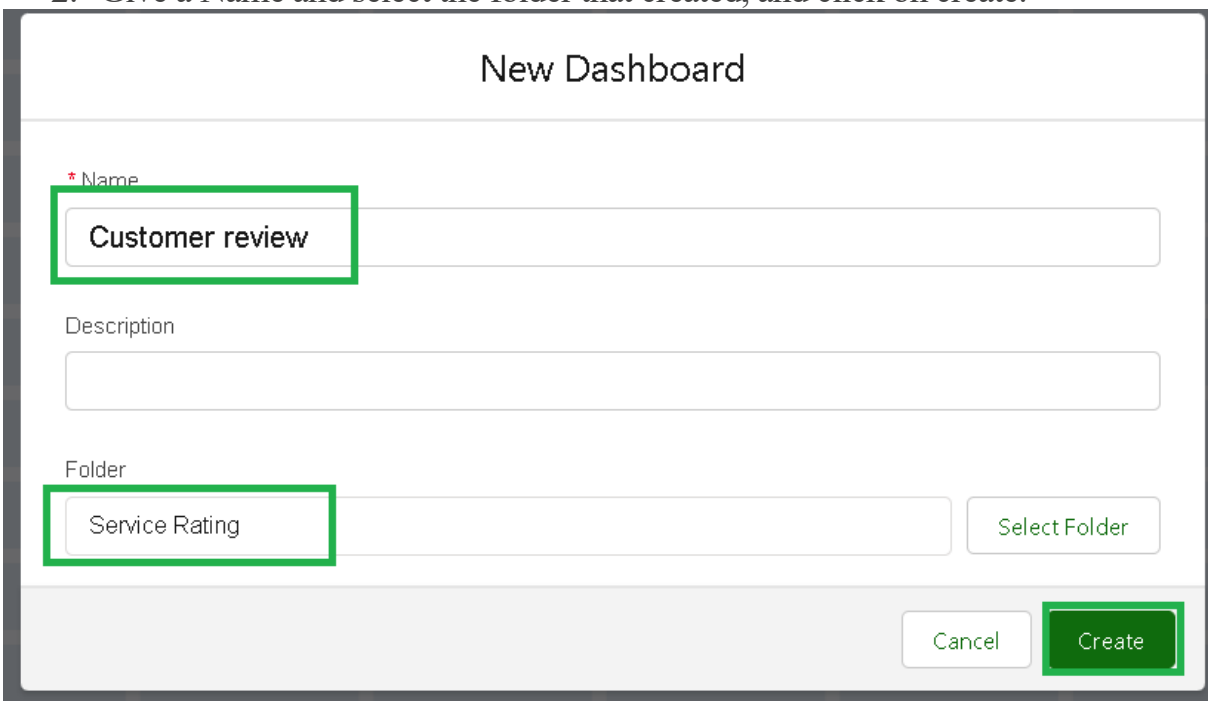
1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as " Service Rating dashboard".
4. Folder unique name will be auto populated.
5. Click save.



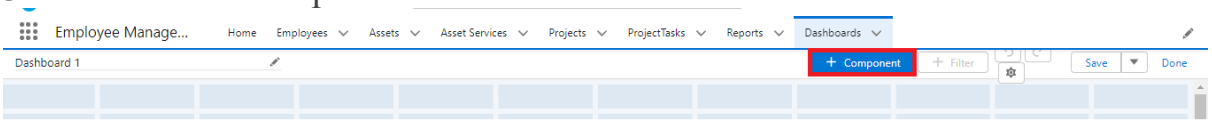
6. Follow the same steps, from Reports Milestone and Activity 2, and provide the sharing settings for the folder that was just created.

Activity 1: Create a dashboard:

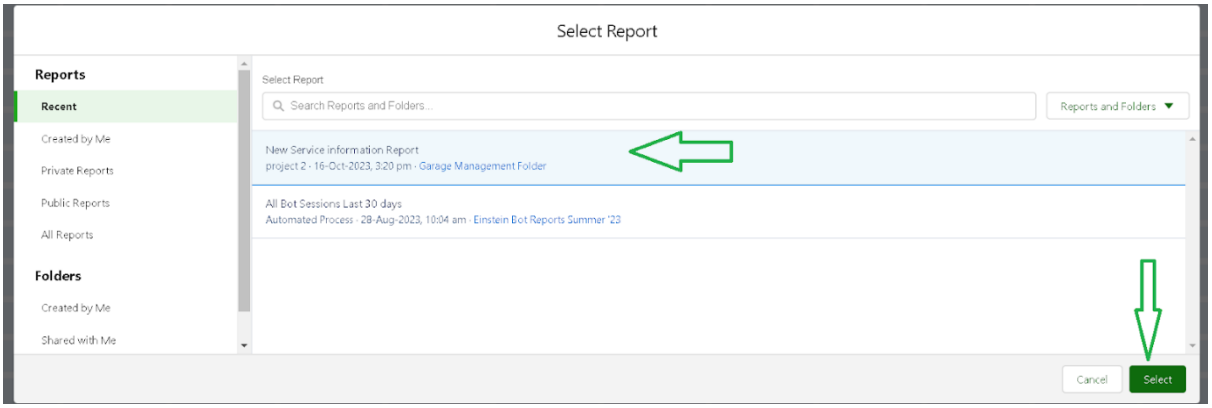
1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.



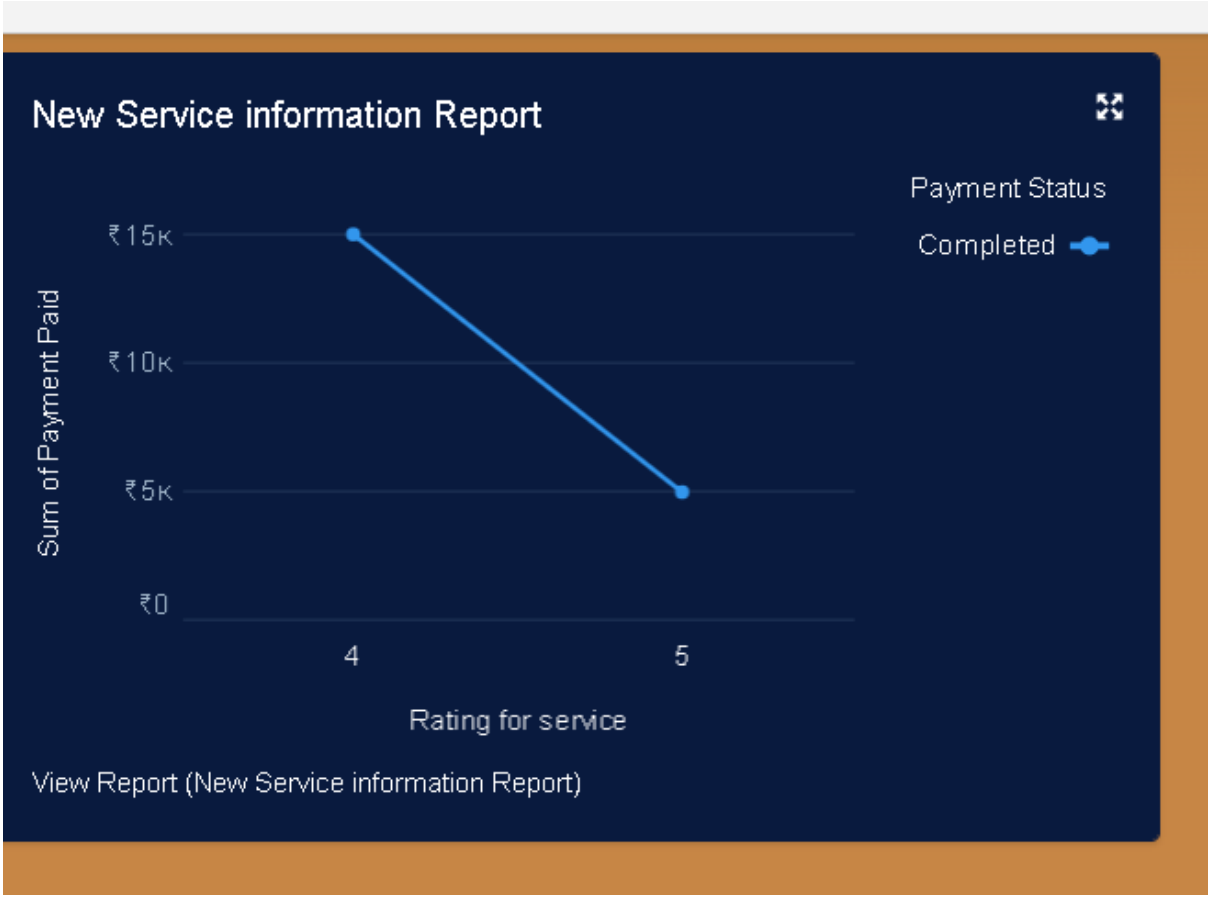
3. Select add component.



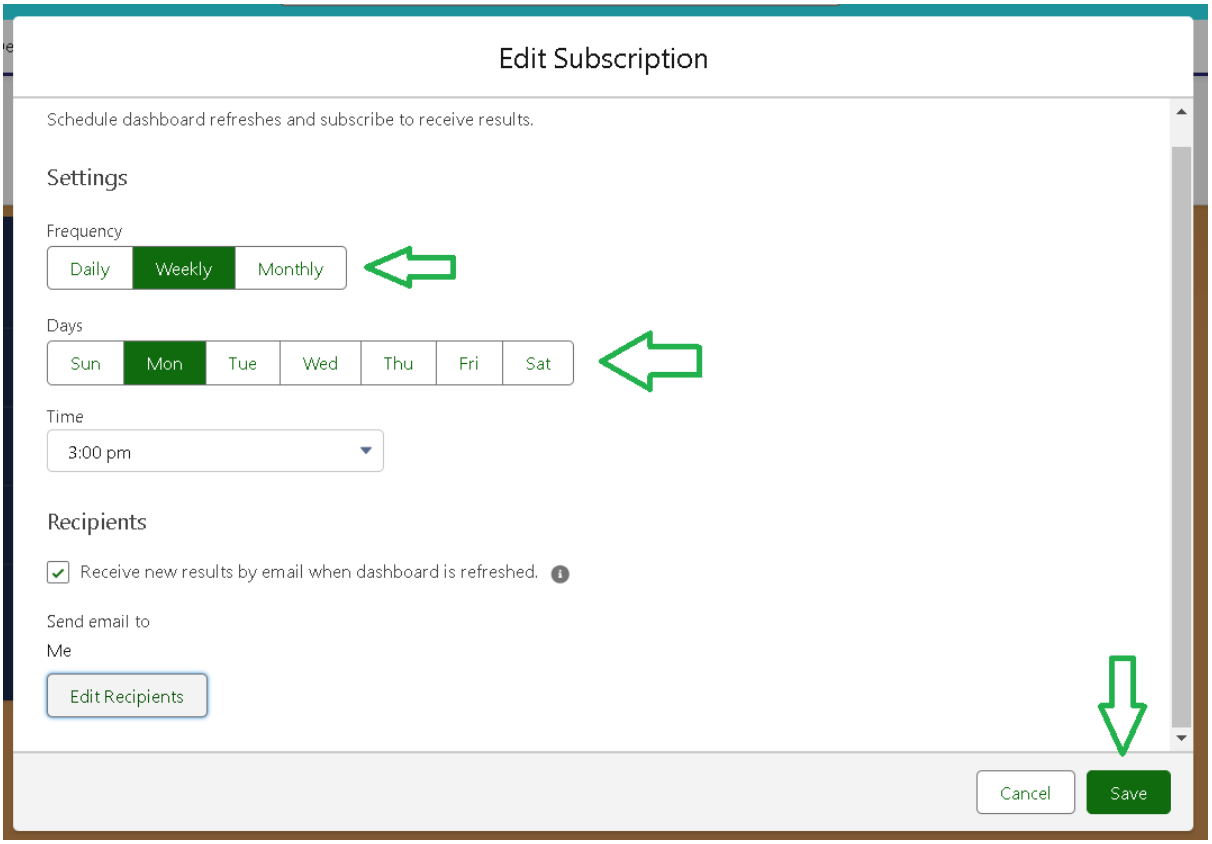
4. Select a Report and click on select.



5. Select the Line Chart. Change the theme.
6. Click Add then click on Save and then click on Done.
7. Preview is shown below.



8. After that Click on Subcribe on top right.
9. Set the Frequency as “ weekly ”.
10. Set a day as monday.
11. And Click on save.

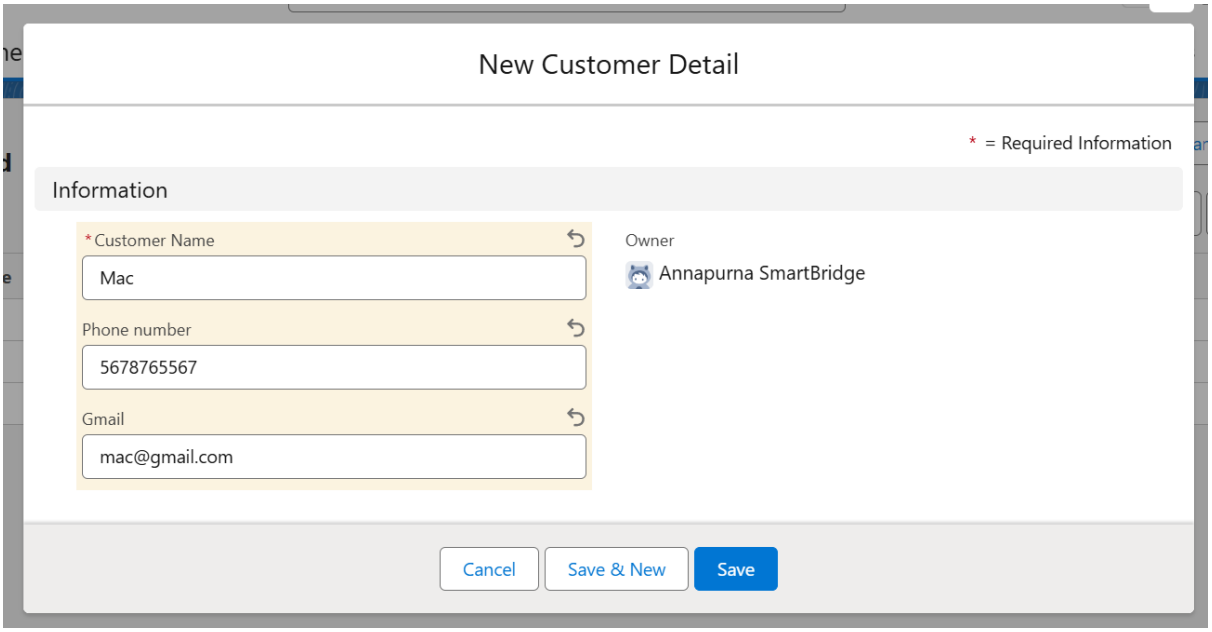


USER ADOPTION:

Activity 1: Creating Records:

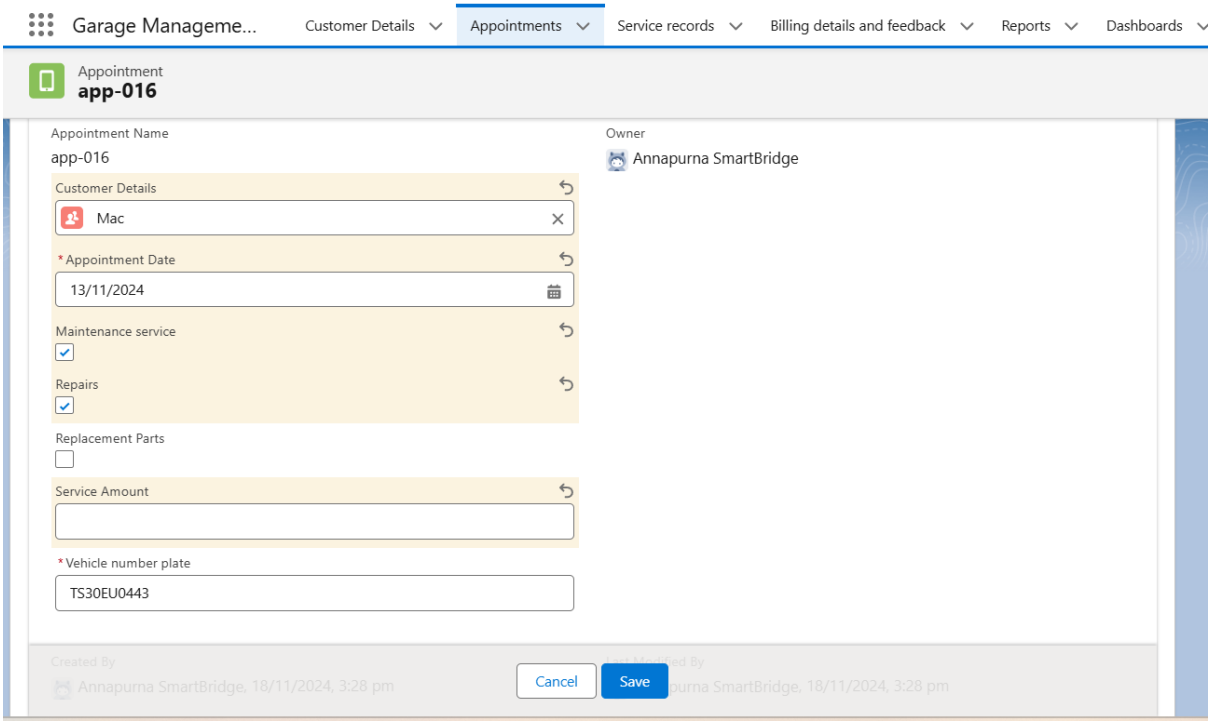
To create a record in the follow objects follow these steps

1. Click on the app launcher located at the left side of the screen.
2. Search for " **Garage Management**" and click on it.
3. Click on the " **Consumer details** tab".
4. Click on new and fill the details as shown below figs, and click save.



Now, Create the Appointment Record

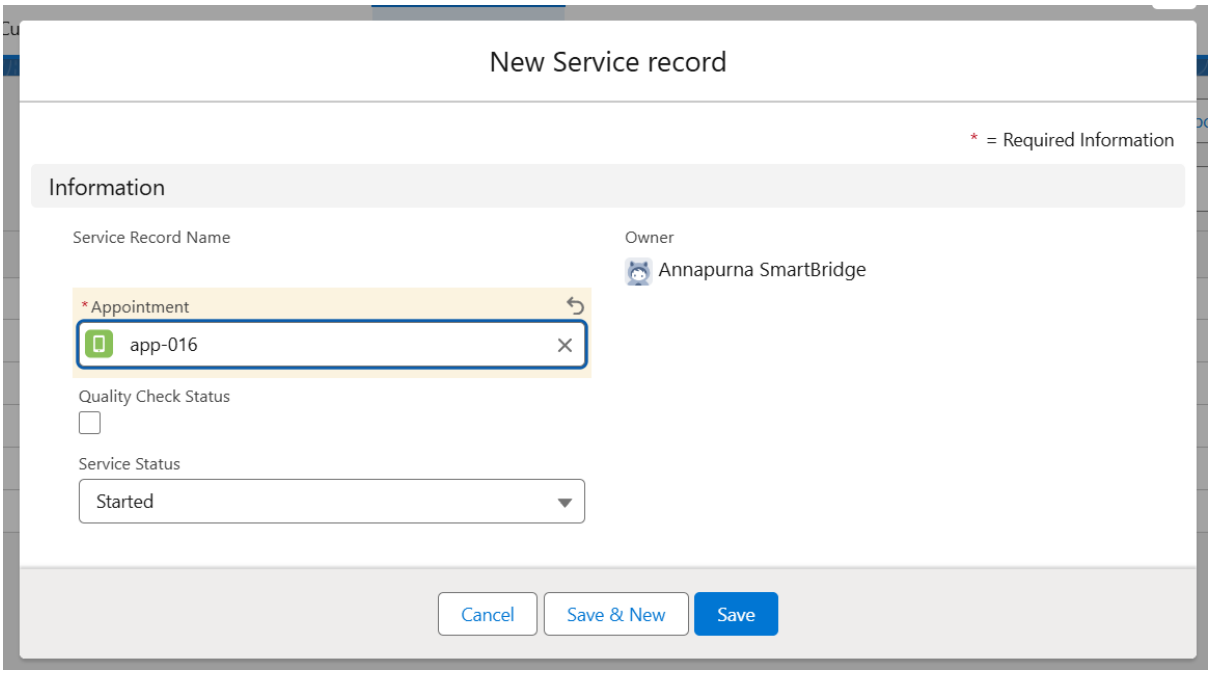
1. Click on the "**Appointment** tab".
2. Enter the customer details as created, while entering Appointment Date enter the date less than the created date.
3. Match the validation while entering the vehicle number plate.
4. Select the services you need.
5. Click on save to see the Service Amount.



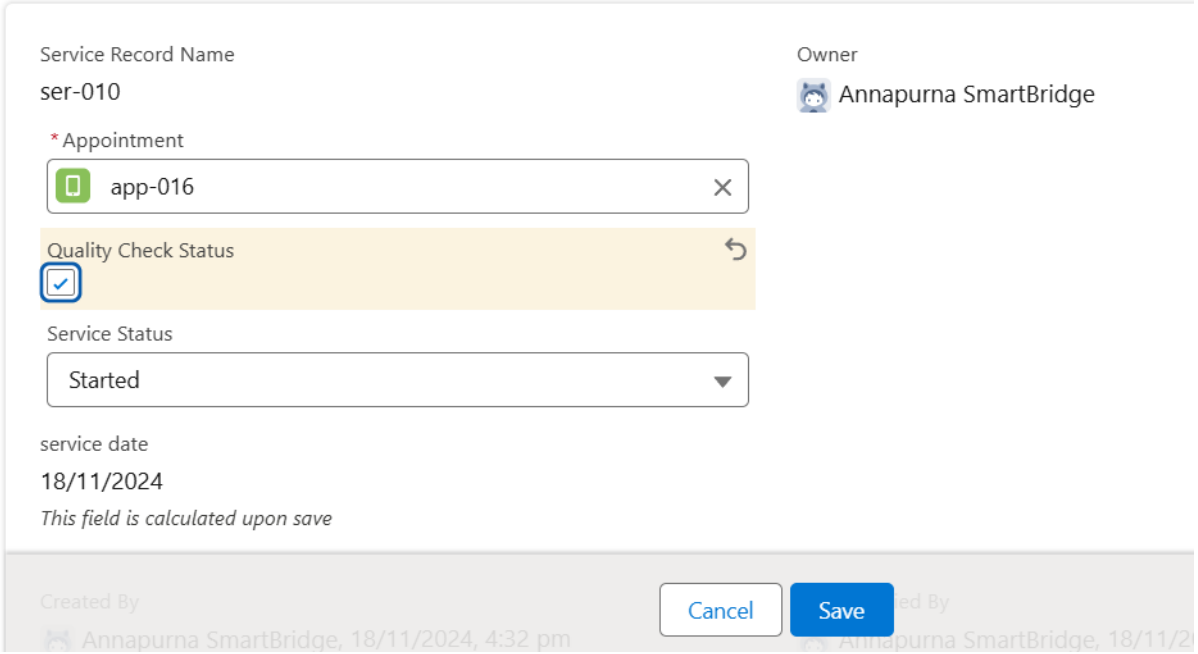
Now, Create a service Record

1. Click on the "**Service record** tab".

2. Enter the Appointment, and started is selected as default.
3. Click on save.



4. Open the record and click on Quality check status as true.
5. Click on save.



6. Now automatically Service status will be moved to completed.

Conclusion:

By leveraging the Salesforce platform, the project successfully established a streamlined and transparent system for managing garage operations. Through efficient coordination between customers, service advisors, mechanics, and inventory managers, the project effectively addressed common challenges such as job tracking, spare parts shortages, and billing delays.

The project "Garage Management System using Salesforce" has been successfully implemented and demonstrates the practical application of Salesforce CRM in the automotive service industry.

Project Achievements:

- Streamlined the process of vehicle service booking, job allocation, and tracking.
- Ensured real-time coordination among customers, mechanics, and service advisors.
- Automated workflows for service requests, approvals, and billing through custom objects, fields, Flows, and Apex triggers.
- Improved transparency with service history, invoices, and payment records using reports and dashboards.
- Enhanced usability with Lightning Apps, role-based security, and customized home pages for different stakeholders.

Student Learning Outcomes:

- Gained hands-on skills in Salesforce development, CRM customization, and automation.
- Developed problem-solving abilities by implementing real-time use cases in service management.
- Improved team collaboration in requirement gathering, development, and testing.
- Acquired industry-relevant exposure to CRM applications in the automotive sector, including project lifecycle management.

Future Scope:

- Mobile Integration: Enable customers to book services and track status via mobile apps.
- AI and Analytics: Use predictive analytics to forecast demand for spare parts and optimize resource allocation.
- Integration with IoT: Connect with vehicle diagnostic systems for automated service recommendations.
- Multi-Branch Expansion: Extend the system to manage operations across multiple garage locations.
- Customer Engagement: Introduce loyalty programs, feedback systems, and personalized offers to enhance customer satisfaction.