Garage Management System Using Salesforce

College Name: Sri Krishna Adithya College of Arts and Science

College Code: bruag

Team ID: NM2025TMID22754

Team Members: 4

Team Leader Name: Hemanth Raj P

Email: 23bsds222hemanthrajp@skacas.ac.in

Team Member 1: Pragadeeshwaran R

Email: 23bsds243pragadeeshwaranr@skacas.ac.in

Team Member 2: Rishikesh P

Email: 23bsds248rishikeshp@skacas.ac.in

Team Member 3: Hari Murugan U

Email: 23bsds218harimuruganu@skacas.ac.in

Garage Management System Using Salesforce

Project Overview: The Garage Management System is a valuable tool designed for automotive repair facilities to help them deliver top-notch service, increase operational efficiency, and build lasting customer relationships. This project uses the Salesforce platform to create a comprehensive application that manages the entire customer and vehicle service lifecycle. The system connects the garage's staff with customer details, appointments, service records, and billing information in one centralized location. Through the use of custom objects, automated flows, Apex triggers, and insightful reports, the process of managing a garage is streamlined and made more efficient, ensuring a seamless experience for both customers and staff.

Objectives:

- **Increase Operational Efficiency:** Streamline garage operations from appointment booking to final billing.
- **Enhance Service Delivery:** Provide staff with the tools and information needed to deliver high-quality, consistent service to every customer.
- **Build Lasting Customer Relationships:** Maintain a comprehensive record of customer interactions and service history to foster loyalty and trust.
- **Ensure Data Accuracy and Integrity:** Use validation and duplicate rules to maintain clean and reliable data for customers and their vehicles.

- **Provide Actionable Insights:** Leverage reports and dashboards to track service performance, revenue, and customer feedback for informed decision-making.
- **Automate Key Processes:** Reduce manual work by automating service amount calculation, status updates, and customer notifications.

Student Outcomes:

- **Hands-on Experience with Salesforce Platform:** Gained practical skills in building a complete application by creating custom objects, fields, and relationships.
- **Proficiency in Business Process Automation:** Learned to automate complex logic using record-triggered Flows and Apex triggers to solve real-world business problems.
- **Understanding of Salesforce Security Model:** Developed a strong understanding of data security by configuring Profiles, Roles, and Sharing Rules to control user access.
- **Enhanced Analytical and Reporting Skills:** Gained experience in creating custom report types, building detailed reports, and visualizing data with dashboards to provide business insights.
- **End-to-End Project Lifecycle Experience:** Learned the complete process from data model design and UI configuration to testing and deployment of a functional Salesforce application.

System Requirements

- **Hardware Requirements:**
    - Computer with a modern processor
    - Minimum 4 GB RAM
    - A stable internet connection

- **Software Requirements:**
  - Salesforce Developer Edition Org
  - A modern web browser (e.g., Google Chrome, Firefox)

Project Phases Overview:

Phase No.	Phase Name	Description	Page Numbers
1	**Requirement Analysis & Planning**	Defining project goals, identifying key features for a garage, and planning the data model and user interactions.	5
2	**Salesforce Development - Backend & Configurations**	Creating the foundational data structure including custom objects, fields, and relationships to store all garage-related data.	5-13
3	**UI/UX Development & Customization**	Building the user-facing components, including creating tabs and a custom Lightning App to ensure an intuitive user experience.	13-15
4	**Automation, Security & Analytics**	Implementing business logic with Flows and Apex, securing data with profiles and sharing rules, and creating reports and dashboards for analysis.	15-59
5	**Deployment, Documentation & Maintenance**	Finalizing the application, performing end-user testing by creating records, and documenting the project for future maintenance and scalability.	60-62

Phase 1: Requirement Analysis & Planning

This initial phase involved defining the core needs of a modern auto garage. The primary requirement was to create a unified system to manage customer information, vehicle appointments, the services performed on vehicles, and the associated billing and feedback processes. The plan focused on creating a relational data model that logically connects these entities for efficient data management and reporting.

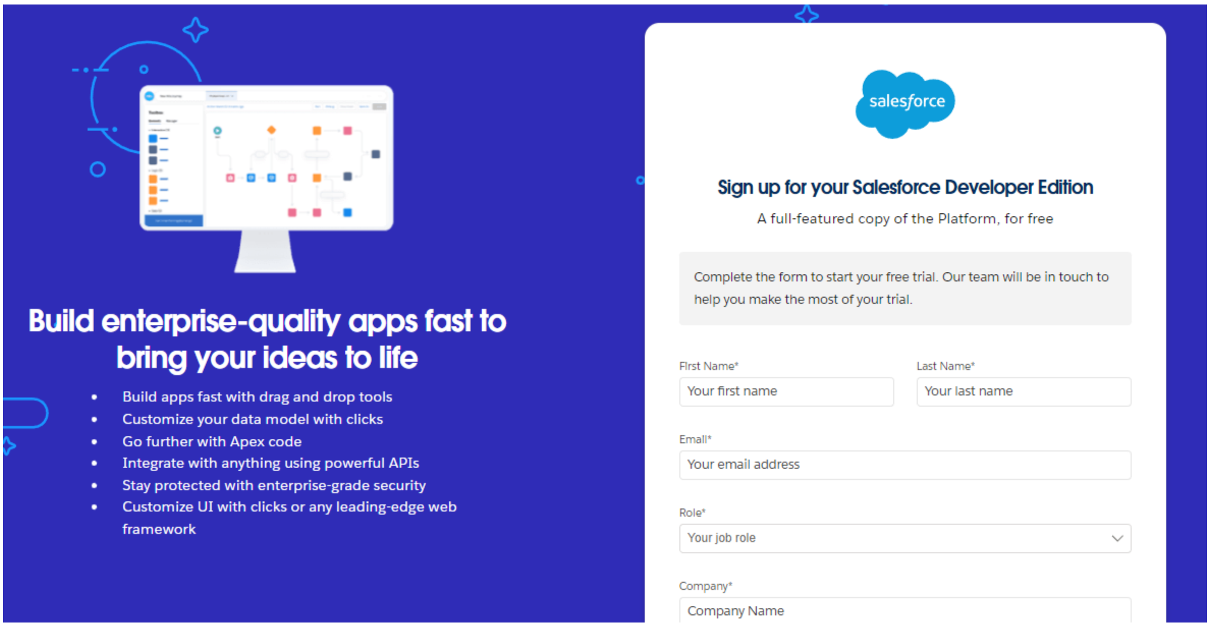
Phase 2: Salesforce Development – Backend & Configurations:-

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 :

i. First name & Last name
ii. Email
iii. Role : Developer

iv. Company : College or Company Name
v. County : India
vi. Postal Code : pin code
vii. Username : should be a combination of your name and company

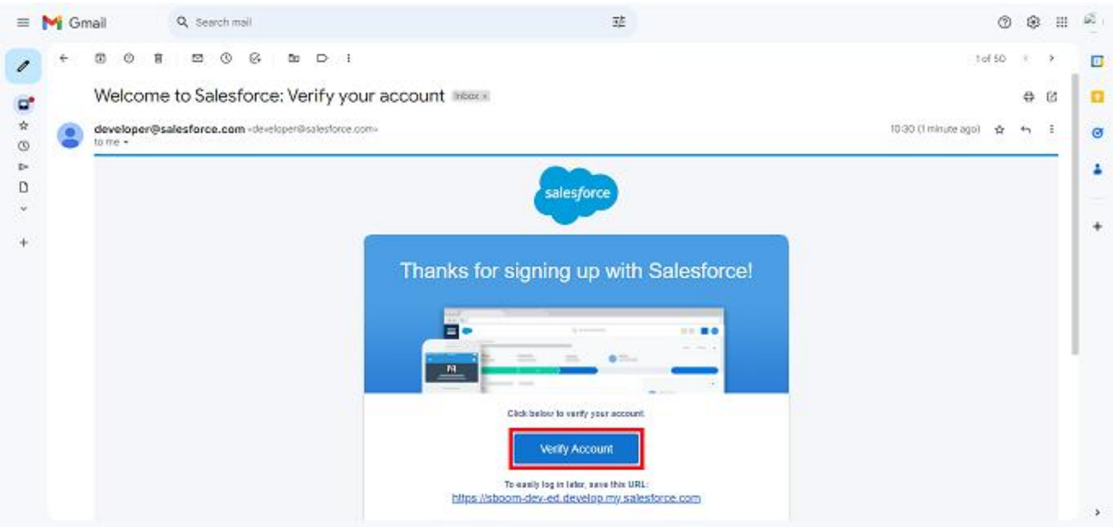
This need not be an actual email id, you can give anything in the format :

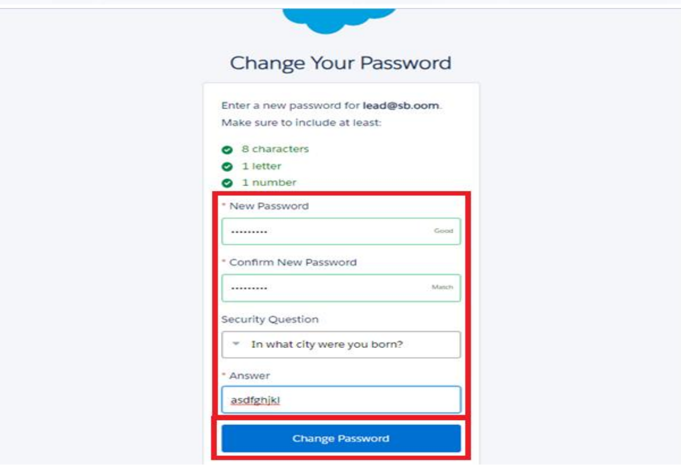
username@organization.com

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.



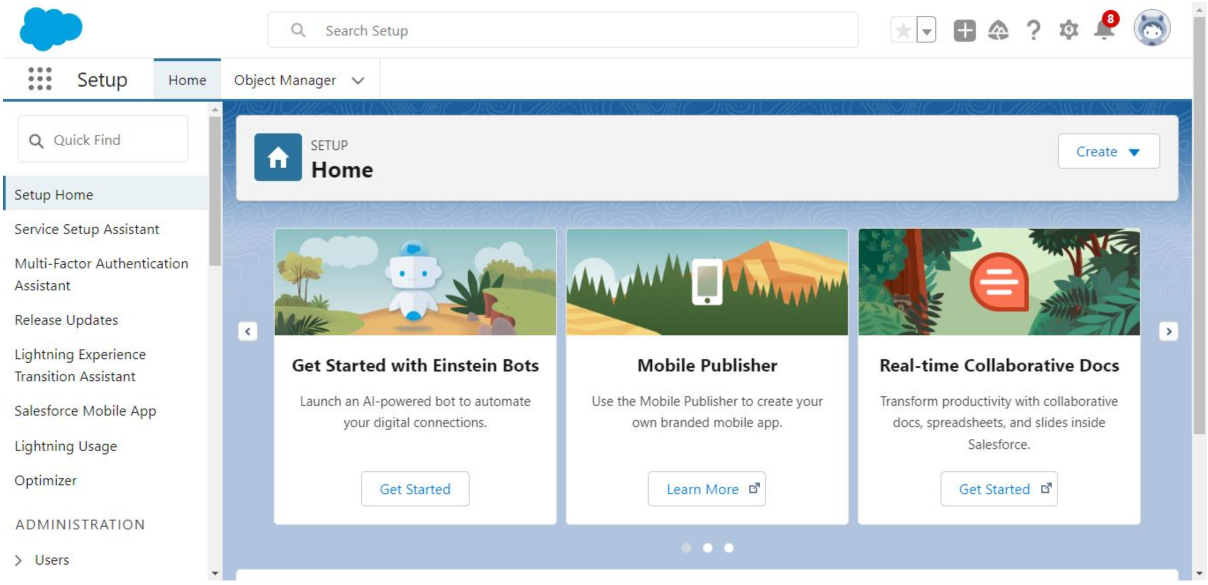


1. Click on Verify Account

2. Give a password and answer a security question and click on change

password.

3. Then you will redirect to your salesforce setup page.



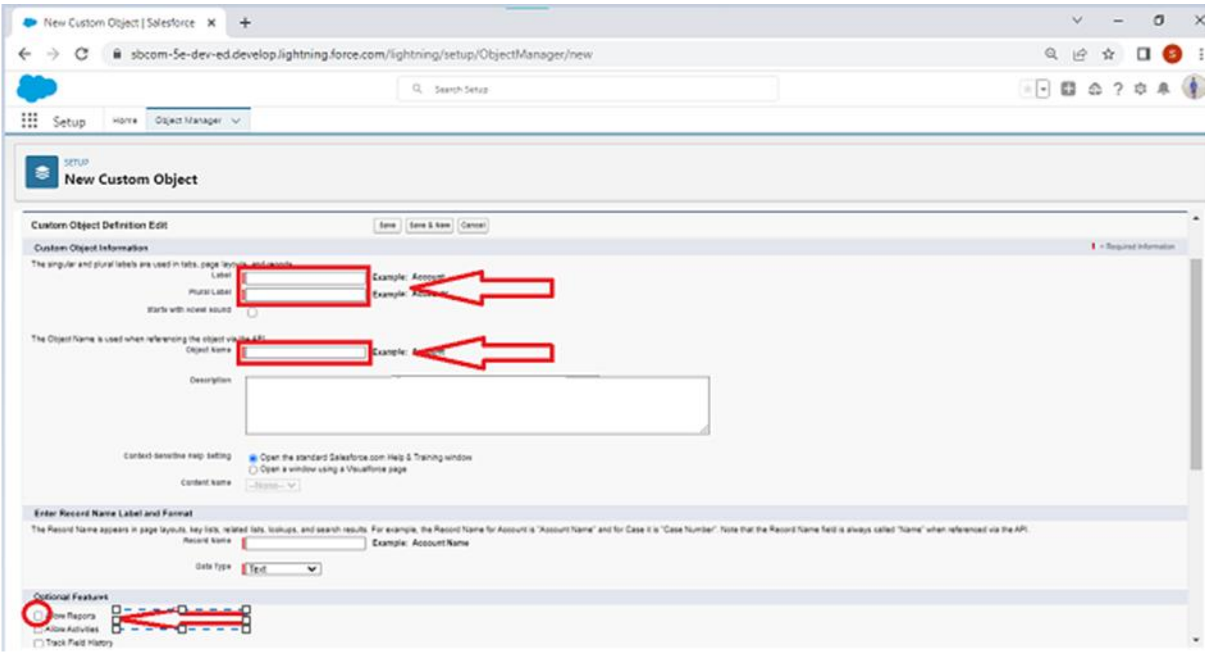
Milestone 2: OBJECTS

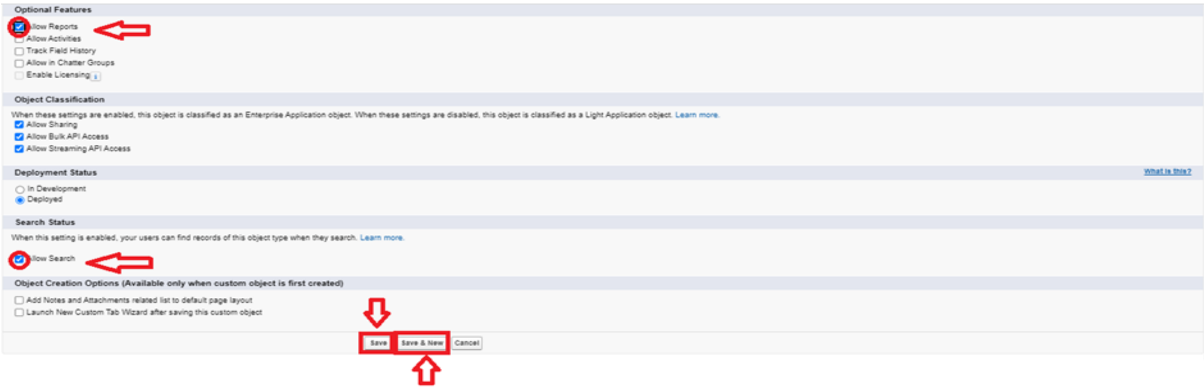
To create an object:

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

2. On Custom object defining page:

3. Enter the label name, plural label name, click on Allow reports, Allow search.





4. Click on Save.

Activity 1: Create Customer Details Object

Create Customer DetailsObject:

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

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

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

Activity 4: Create Billing details and feedback Object

To create an object:

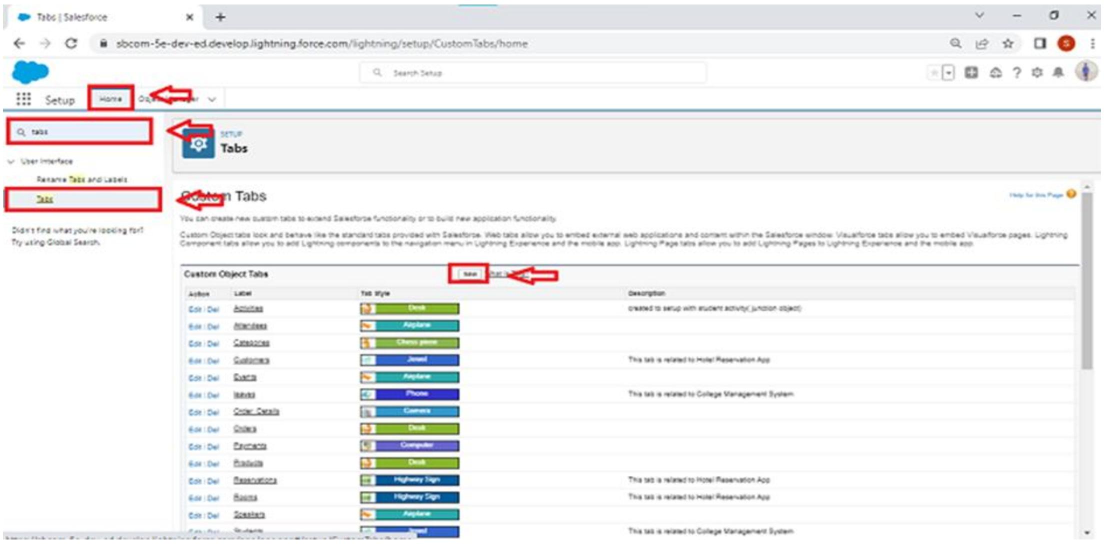
1. From the setup page>>Click on Object Manager>>Click on Create>>Click on Custom Object.
2. Enter the label name>>Billing details and feedback
3. Plural label name>>Billing details and feedback
4. Enter Record Name Label and Format
5. Record Name>>Billing details and feedback Name
6. Data Type>>Auto Number
7. Display Format>>bill-{000}
8. Starting number>>1
9. Click on Allow reports and Track Field History,
10. Allow search>>Save.

Milestone 3: TABS

Activity 1: Creating Custom Tabs

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)



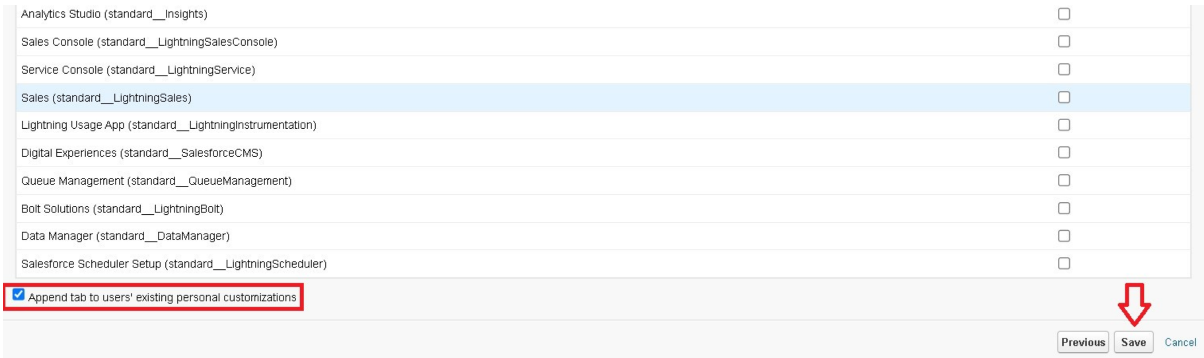
2. Select Object(Customer Details) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .

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

4. Click save.


Tab Style Selector
Create your own style
Hide styles which are used on other tabs
Airplane
Alarm clock
Apple
Balls
Bank[1]
Bell
Big top
Boat[1]
Books
Bottle
Box
Bridge
Building
Building Block
Caduceus
Camera
Can
Car
Castle
CD/DVD
Cell phone
Chalkboard
Chess piece
Chip
Circle
Compass
Computer
Credit card
CRT TV
Cup
Desk[1]
Diamond
Dice
Factory
Fan
Flag
Form
Gears
Globe
Guitar
Hammer
Hands
Handsaw
Headset
Heart[1]
Helicopter
Hexagon
Highway Sign
Hot Air Balloon
Insect
IP Phone
Jewel
Keys
Laptop
Leaf
Lightning
Save
Cancel


Step 3. Add to Custom Apps
Step 3 of 3
Choose the custom apps for which the new custom tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each Custom App.
Custom App
Include Tab
Platform (standard__Platform)
Sales (standard__Sales)
Service (standard__Service)
Marketing (standard__Marketing)
Sample Console (standard__ServiceConsole)
High Volume Customer Portal User
Authenticated Website User
App Launcher (standard__AppLauncher)


Analytics Studio (standard__Insights)
Sales Console (standard__LightningSalesConsole)
Service Console (standard__LightningService)
Sales (standard__LightningSales)
Lightning Usage App (standard__LightningInstrumentation)
Digital Experiences (standard__SalesforceCMS)
Queue Management (standard__QueueManagement)
Bolt Solutions (standard__LightningBolt)
Data Manager (standard__DataManager)
Salesforce Scheduler Setup (standard__LightningScheduler)
Append tab to users' existing personal customizations
Previous
Save
Cancel

Activity 2: Creating Remaining Tabs

Creating Remaining Tabs

1. Now create the Tabs for the remaining Objects, they are “Appointments, Service records,Billing details and feedback”.

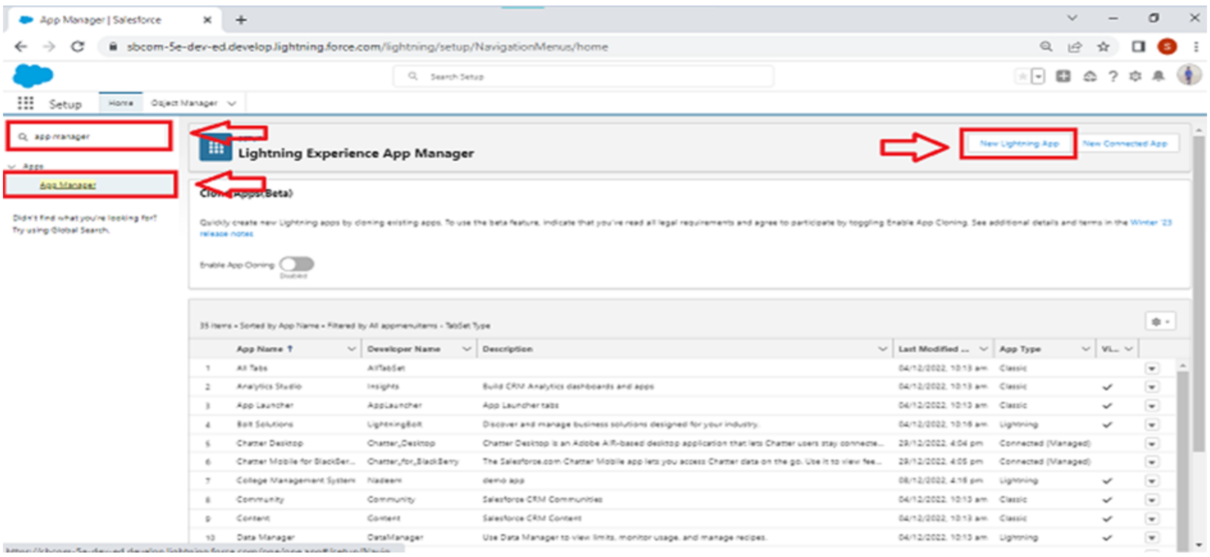
2. Follow the same steps as mentioned in Activity -1.

Phase 3: UI/UX Development & Customization

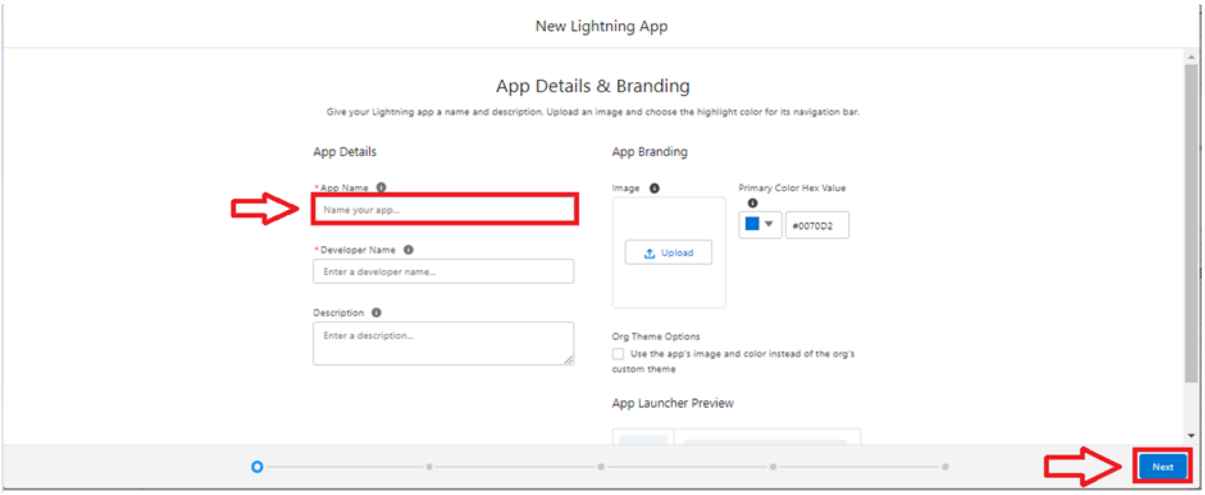
Activity 1: Create a Lightning App

To create a lightning app page:

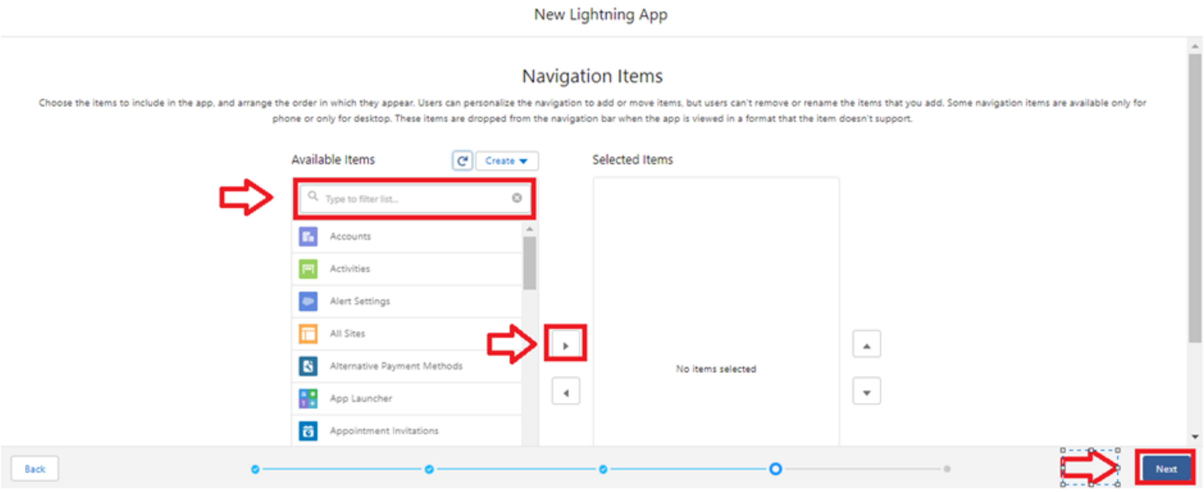
1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on new lightning App.



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 default >> Next.

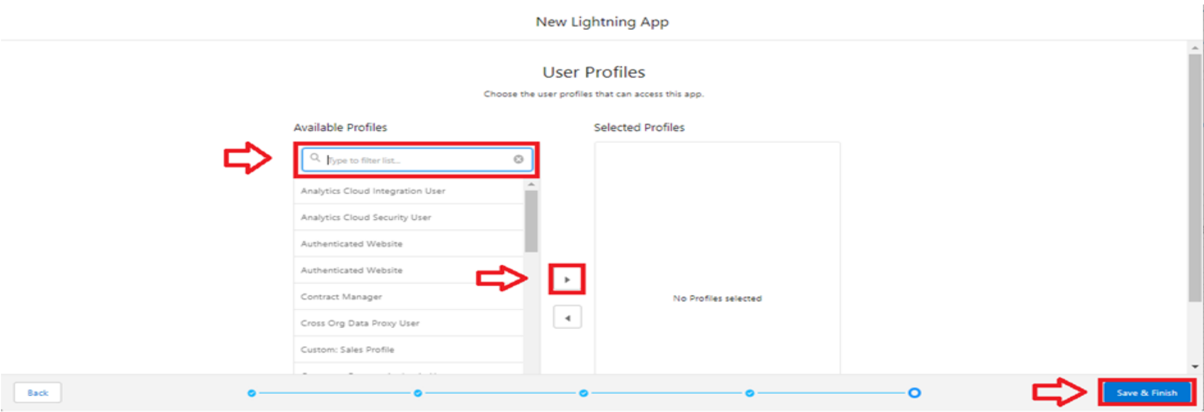


3. To Add Navigation Items:



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.

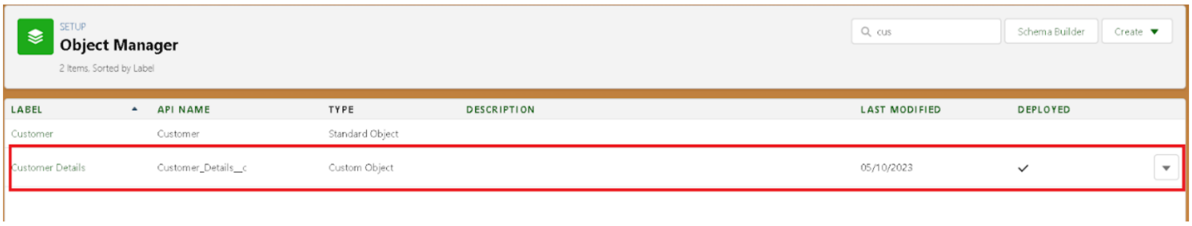
Phase 4: Automation, Security & Analytics

Milestone 5: FIELDS

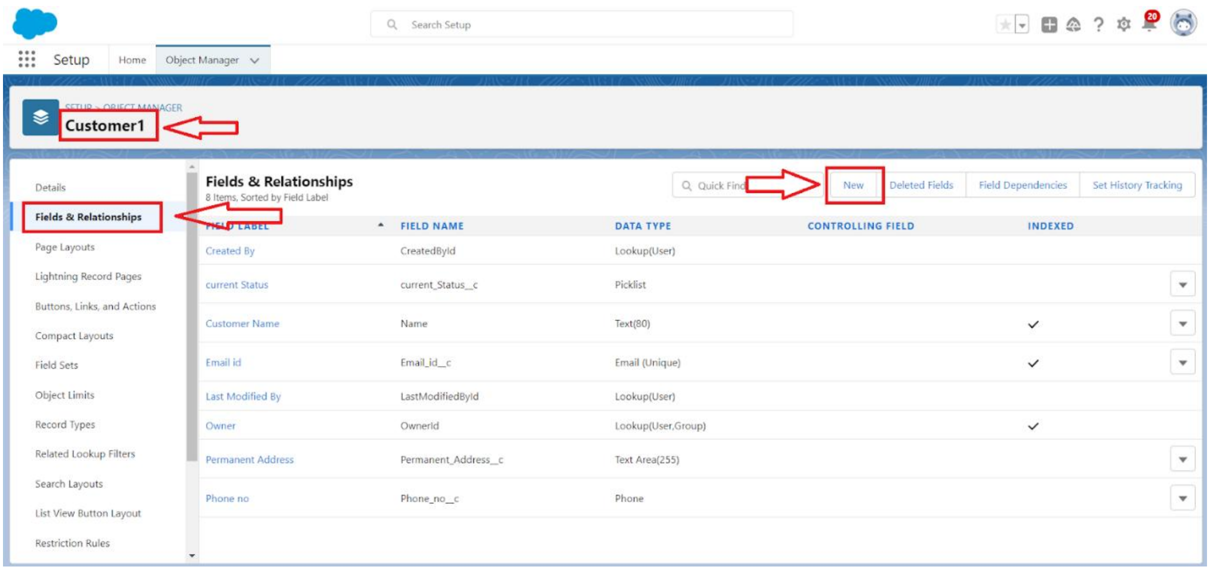
Activity 1: Creation of fields for the Customer Details object

To create fields in an object:

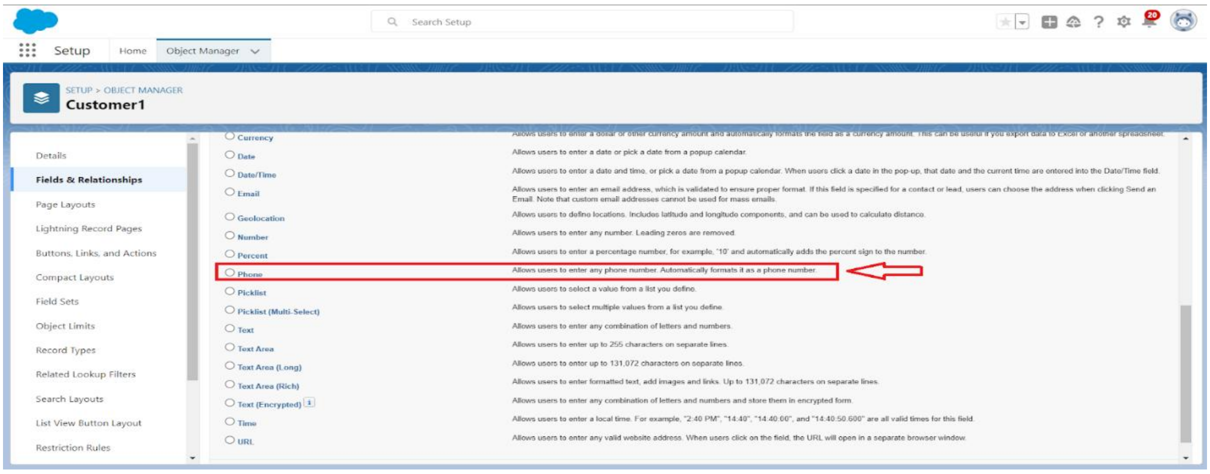
1. Go to setup >> click on Object Manager >> type object name (Customer Details) in search bar >> click on the object.



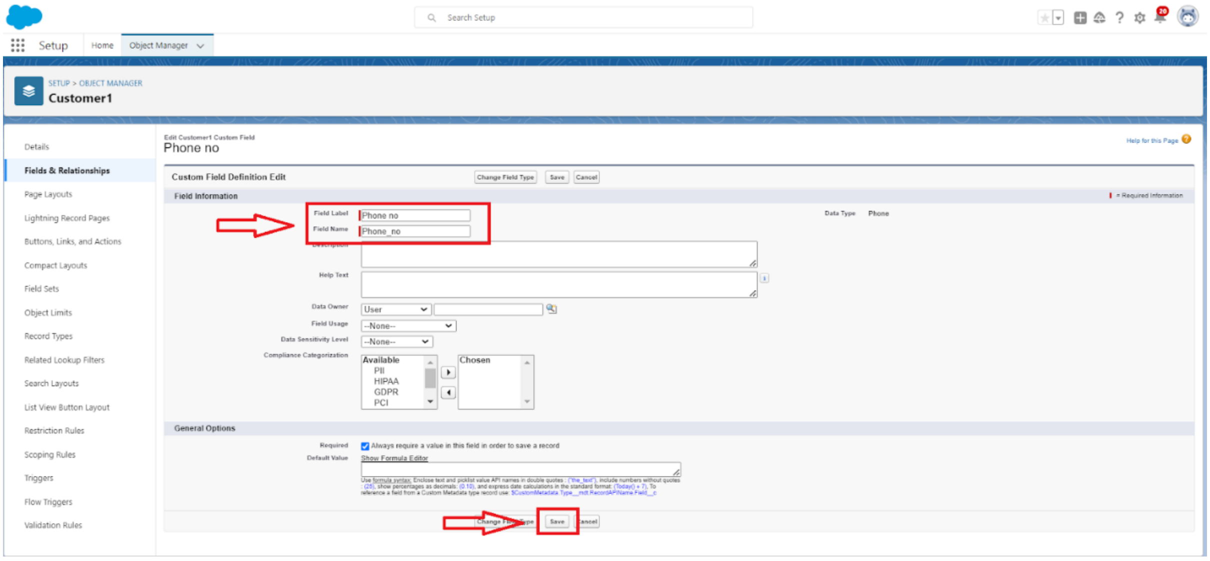
2. Now click on "Fields & Relationships" >> New



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.

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

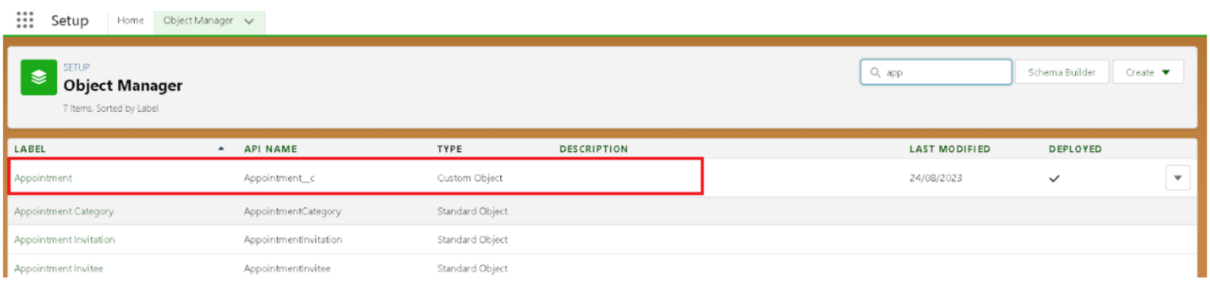
4. Fill the Above as following:

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

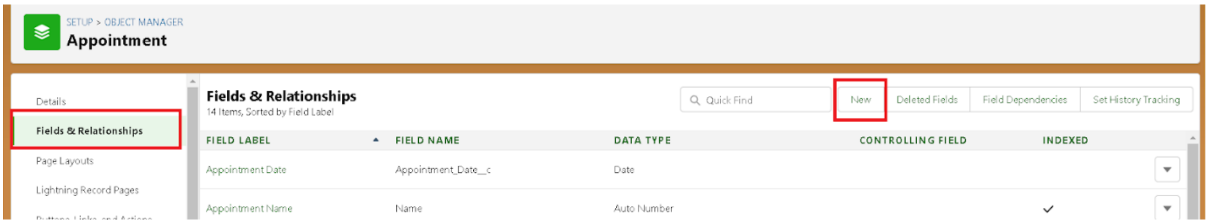
Activity 2: Creation of Lookup Fields

Creation of Lookup Field on Appointment Object :

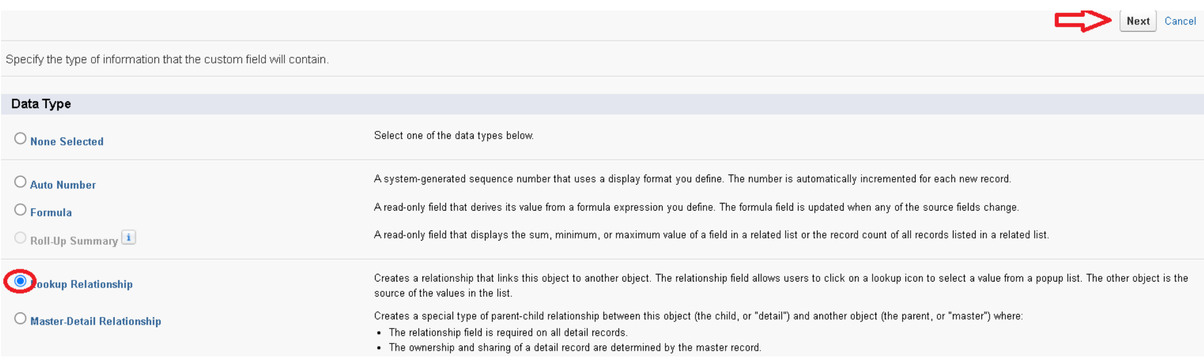
1. Go to setup >> click on Object Manager >> type object name(Appointment) in the search bar >> click on the object.



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



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

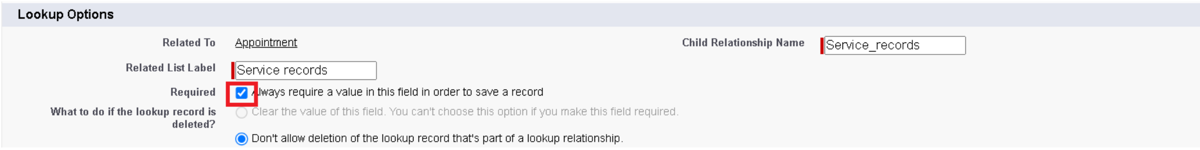


4. Select the related object “ Customer Details” and click next.

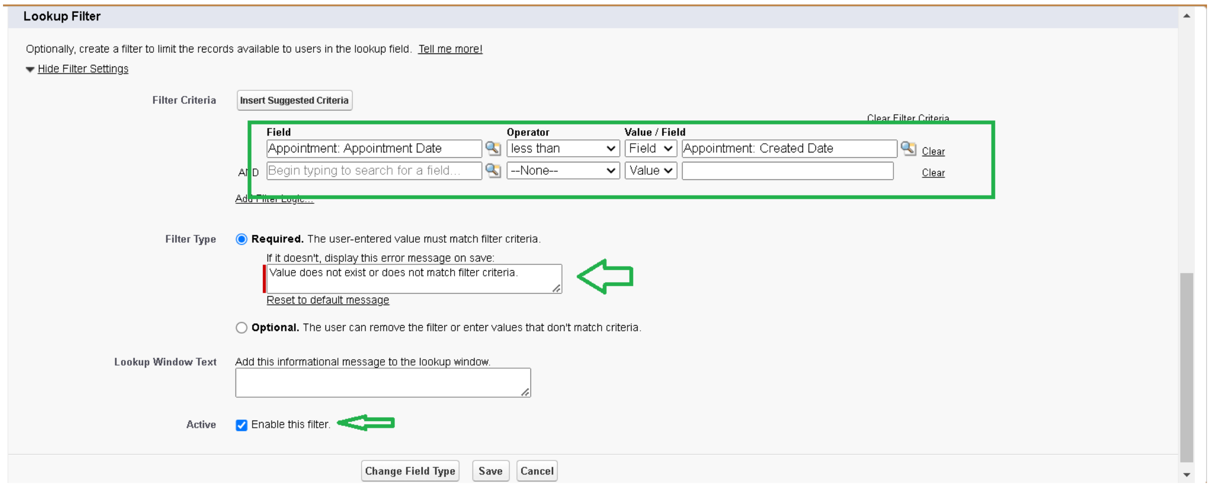
5. Next >> Next >> Save.

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.

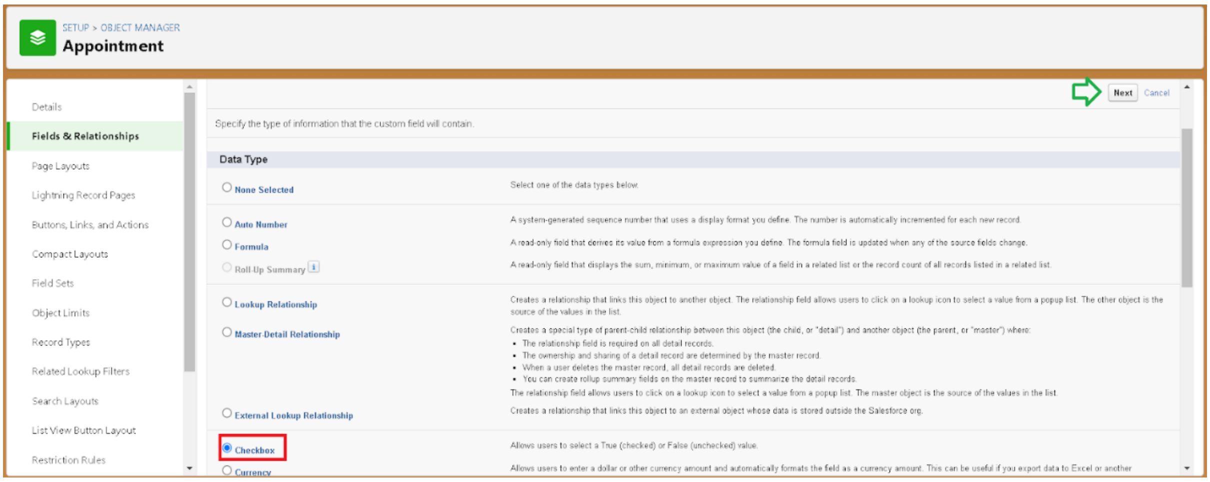
Creation of Lookup Field on Billing details and feedback Object :

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 Checkbox Fields

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.

Creation of Another Checkbox Field on Appointment Object :

1. Repeat the steps form 1 to 3.
2. Give the Field Label : Repairs
3. Field Name : is auto populated
4. Default value : unchecked
5. Click on next >> next >> save.
6. Follow the same and create another checkbox with given names
7. Give the Field Label : Replacement Parts
8. Field Name : is auto populated
9. Default value : unchecked
10. Click on next >> next >> save.

Creation of Checkbox Field on Service records Object :

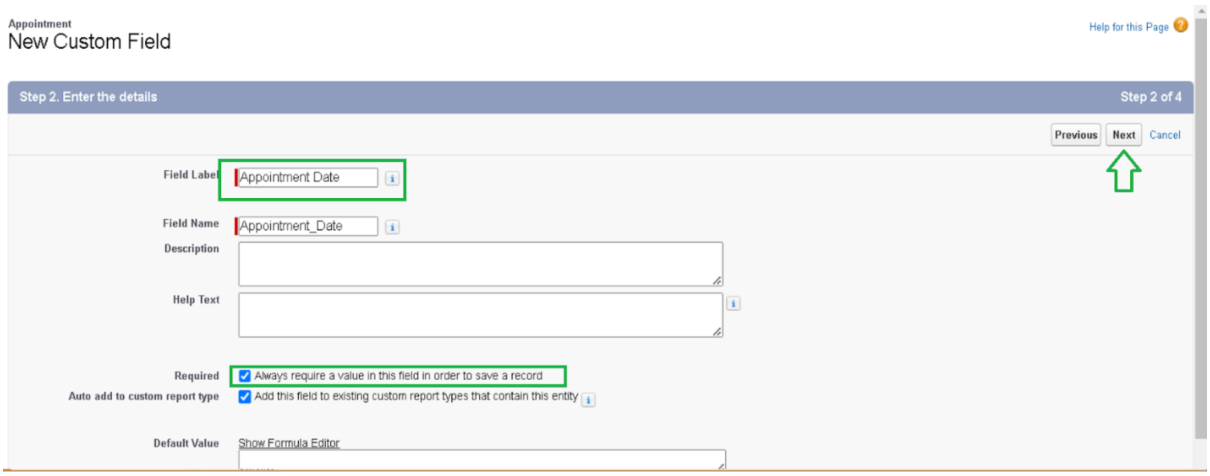
1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Check box” as data type and click Next.

4. Give the Field Label : Quality Check Status
5. Field Name : is auto populated
6. Default value : unchecked
7. Click on next >> next >> save

Activity 4: Creation of date Fields

Creation of Date Field on Appointment Object :

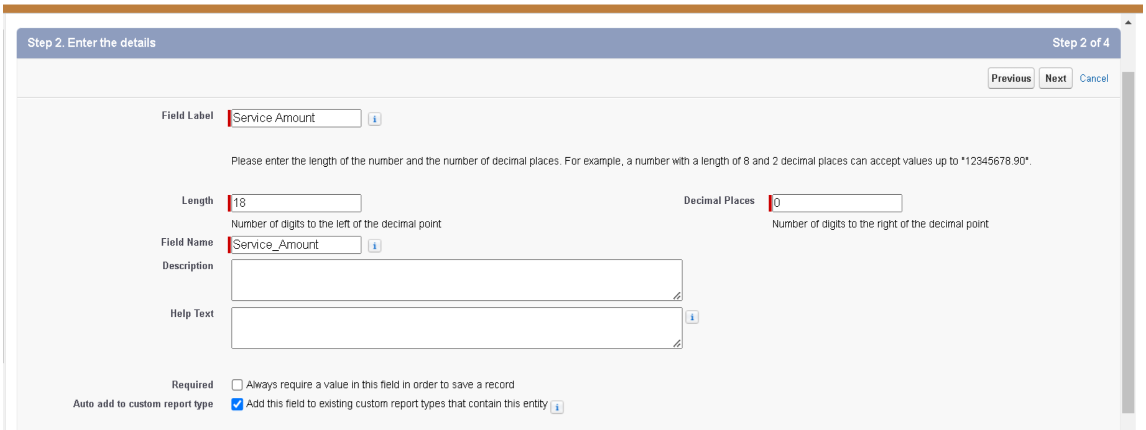
1. Go to setup >> click on Object Manager >> type object name(Appointment) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Date” as data type and click Next.
4. Give the Field Label : Appointment Date
5. Field Nme : is auto populated
6. Make it as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



Activity 5: Creation of Currency Fields

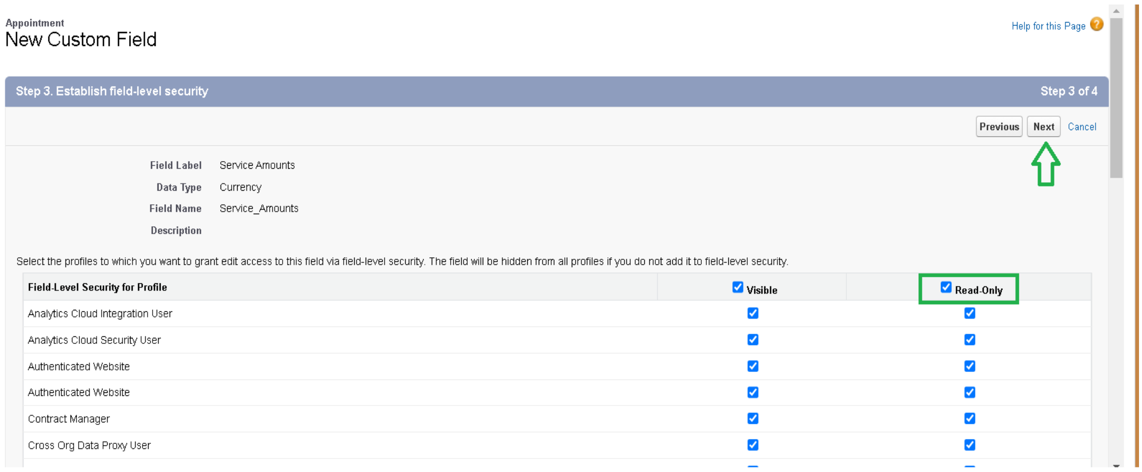
Creation of Currency Field on Appointment Object :

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 Name : is auto populated



6. Click on next

7. Give read only for all the profiles in field level security for profile.



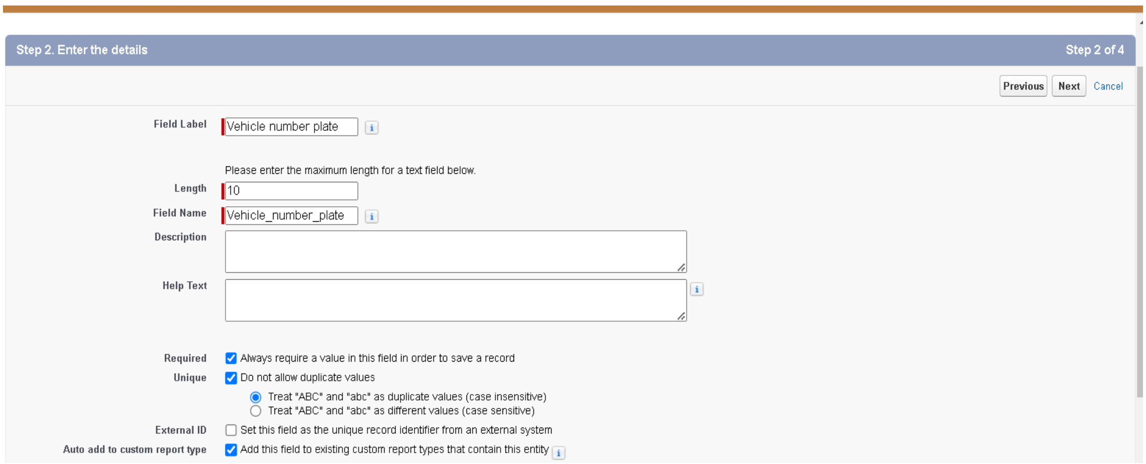
8. Click on next > > save.

Creation of Currency Field on Billing details and feedback Object :

1. Follow the same steps as mentioned above in Billing details and feedback Object.
2. Change the label name as mentioned.
3. Give the Field Label : Payment Paid
4. Field Nme : is auto populated

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.

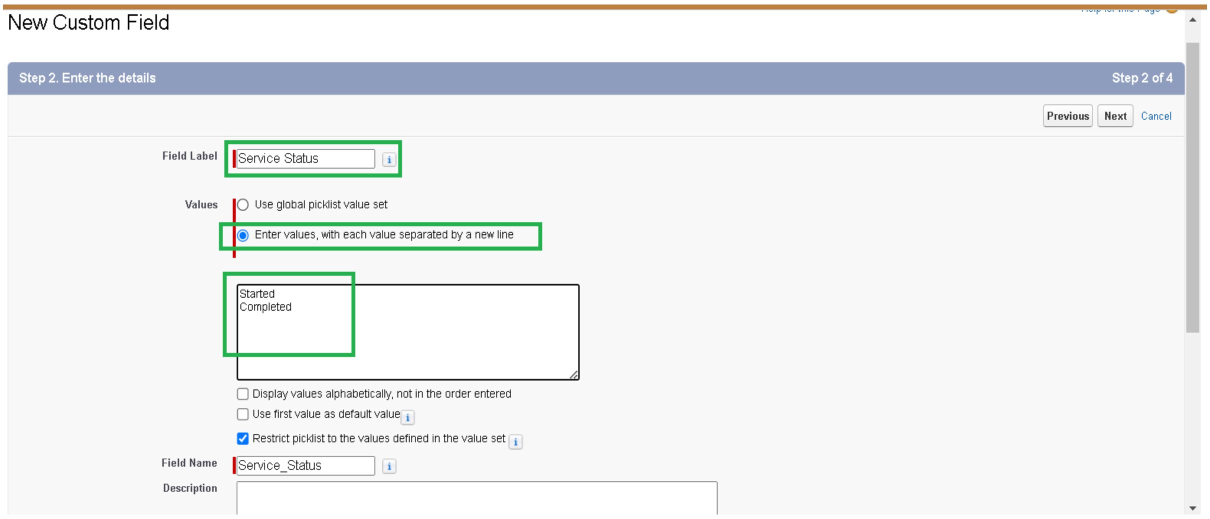
Creation of Text Fields in Billing details and feedback object :

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 “text” as data type and click Next.
4. Give the Field Label : Rating for service
5. Field Name : is auto populated
6. Length : 1
7. Make field as Required.
8. Click on next >> next >> save

Activity 6: Creation of Picklist Fields

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.

Creation of Picklist Fields in Billing details and feedback object :

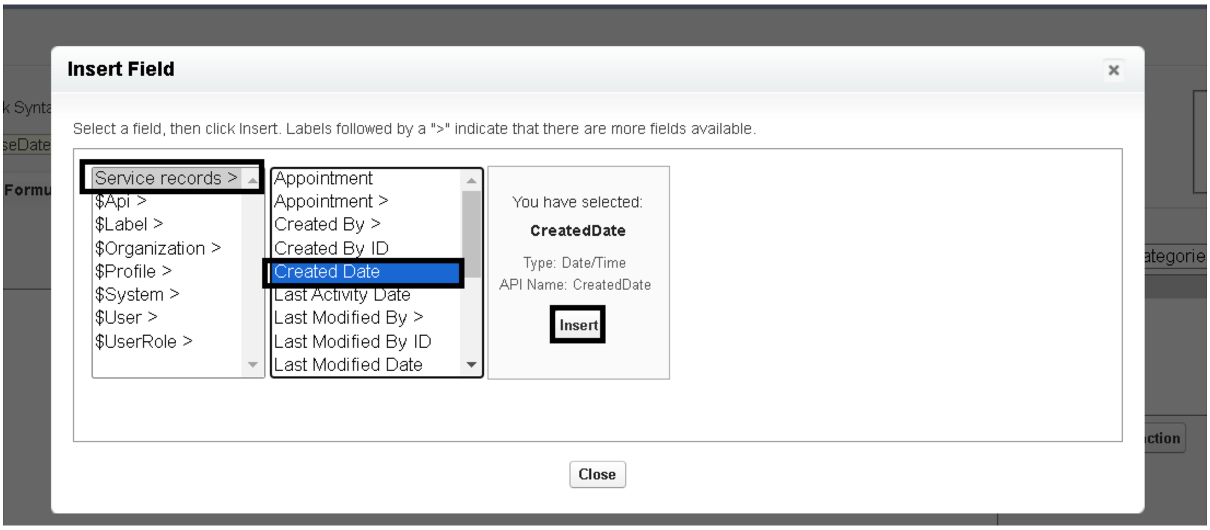
1. Go to setup >> click on Object Manager >> type object name(Billing details and feedback) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as "Picklist" and click Next.
4. Enter Field Label as "Payment Status", under values select "Enter values,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

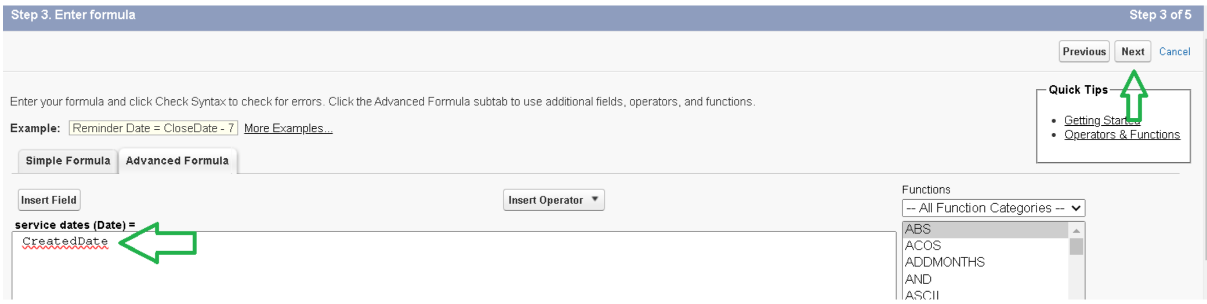
Creating Formula Field in Service records Object

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





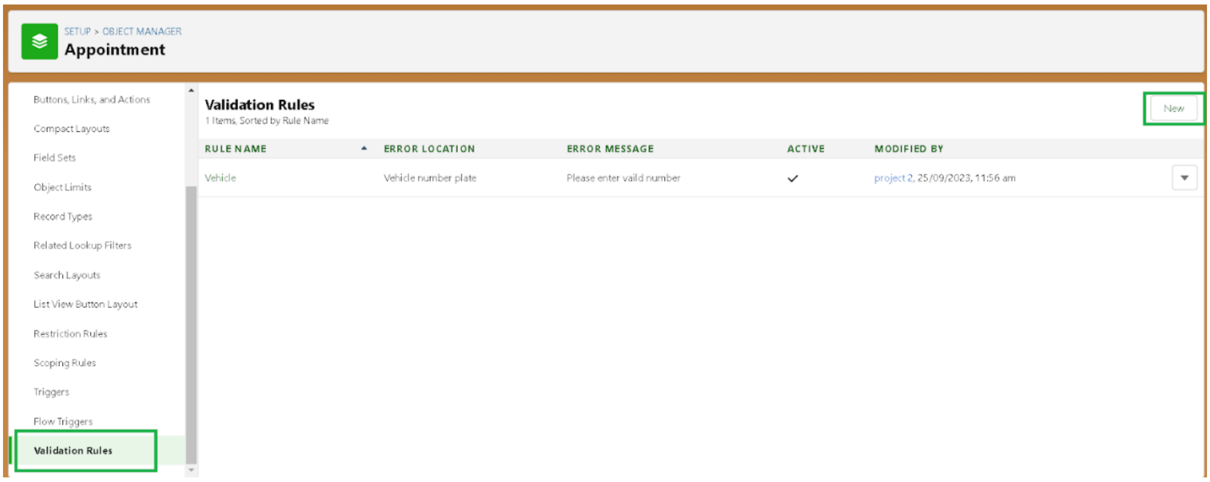
6. click “Check Syntax” .
7. Click next >> next >> Save.

Milestone 6: VALIDATION & DUPLICATE RULES

Activity 1: Create Validation Rules

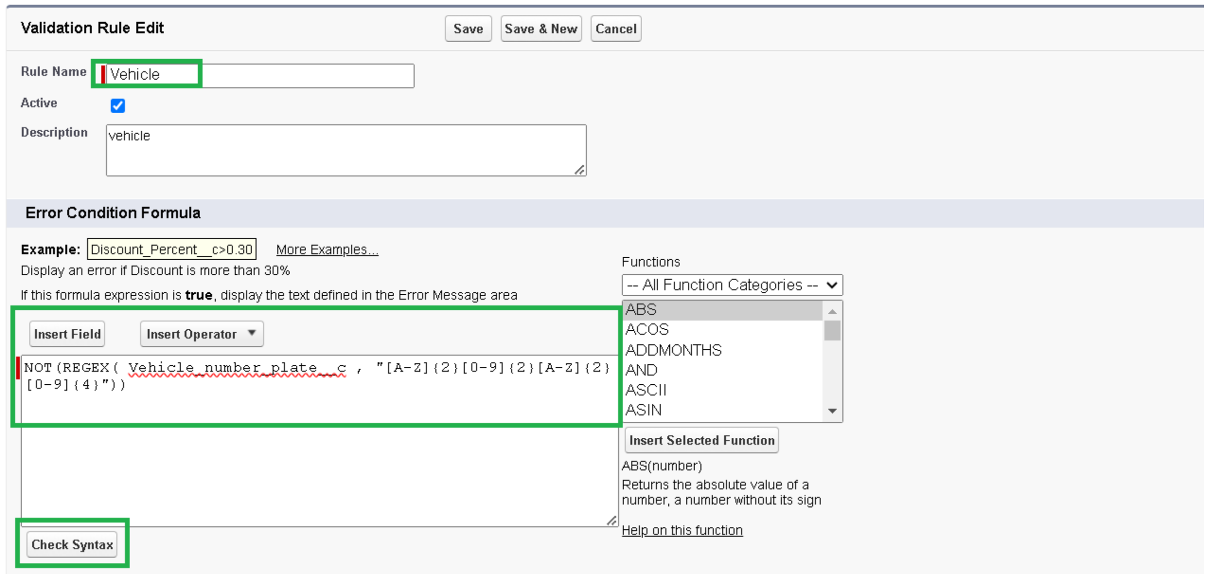
To create a validation rule to an Appointment Object:

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.

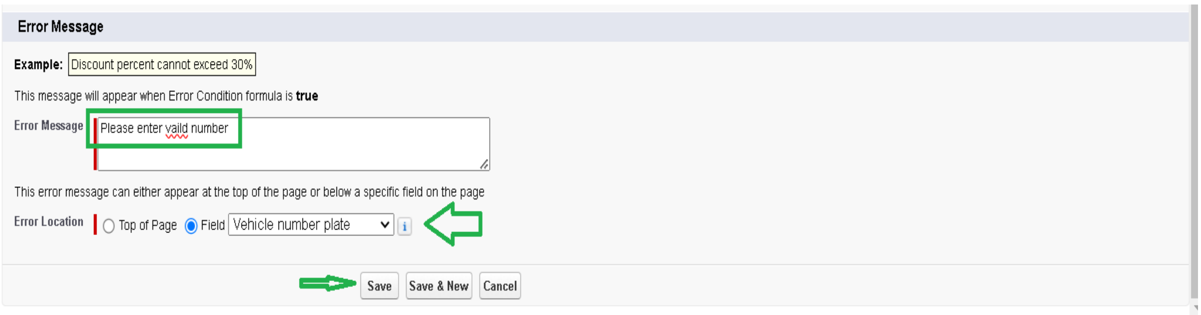


3. Enter the Rule name as " Vehicle ".

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.

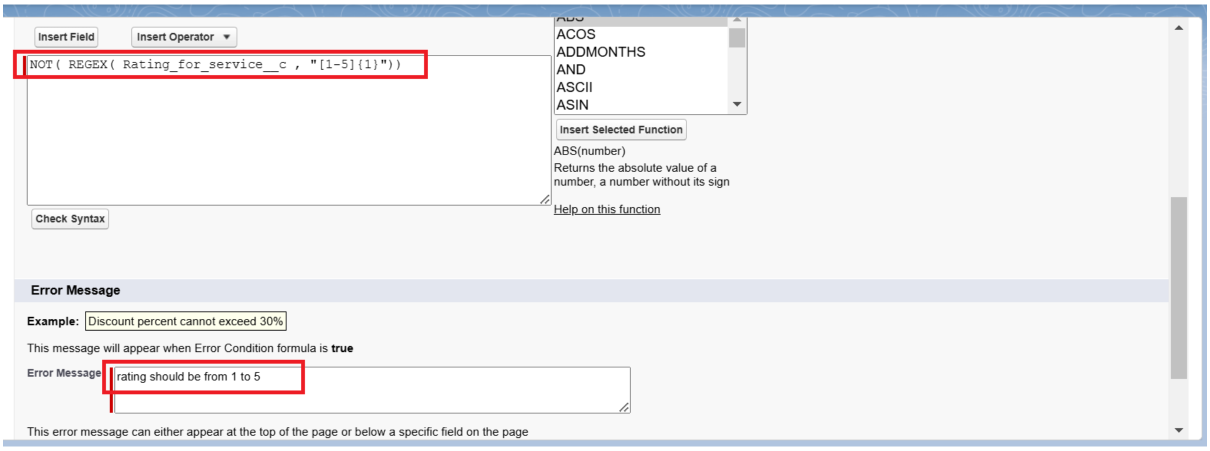


To create a validation rule to an Billing details and feedback Object

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}"))

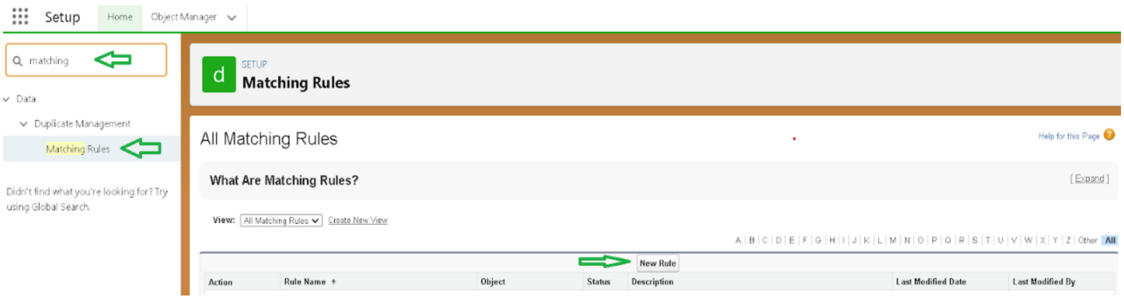
5. Enter the Error Message as “rating should be from 1 to 5”, select the Error location as Field and select the field as “Rating for Service”, and click Save.



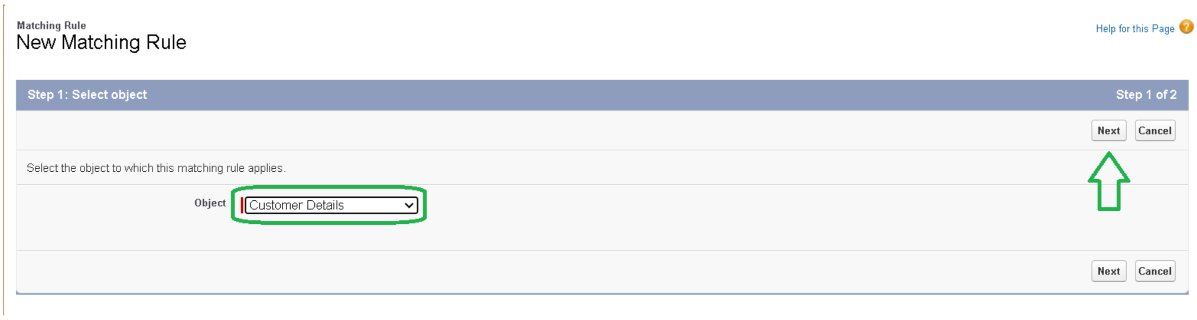
Activity 2: Create Duplicate Rules

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.

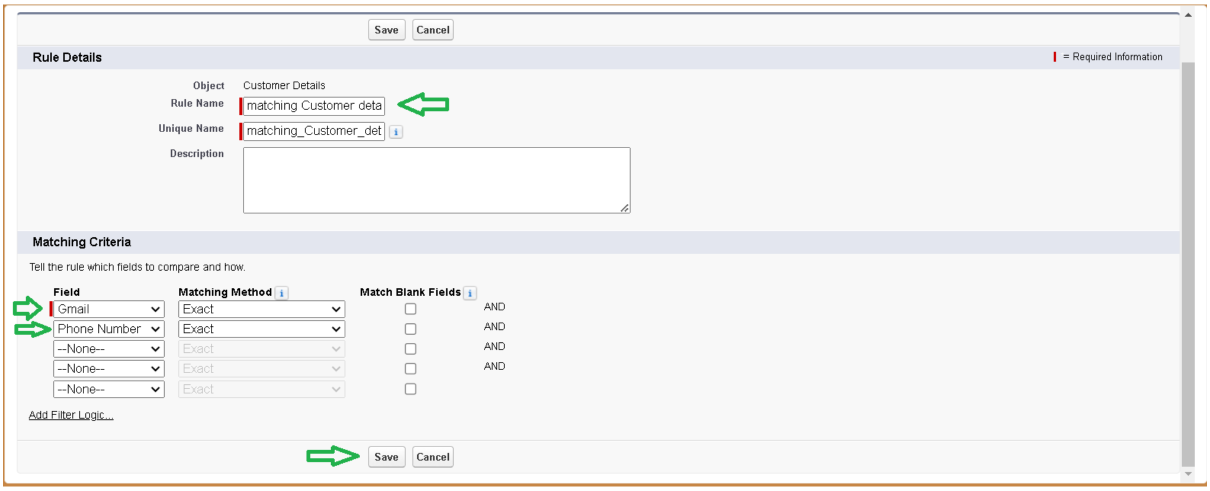


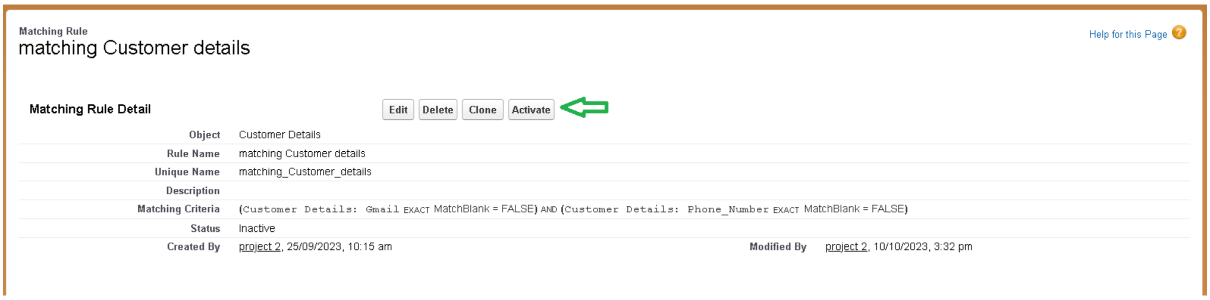
4. Give the Rule name : Matching customer details
5. Unique name : is auto populated
6. Define the matching criteria as

7.	Field	Matching Method
	Gmail	Exact
	Phone Number	Exact

8. Click save.

9. After Saving Click on Activate.

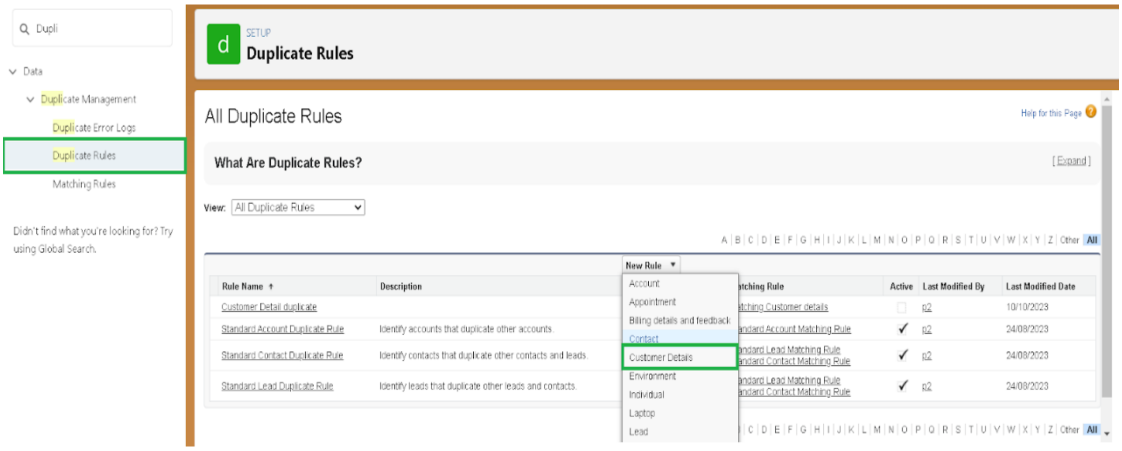




To create a Duplicate rule to an Customer details Object

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



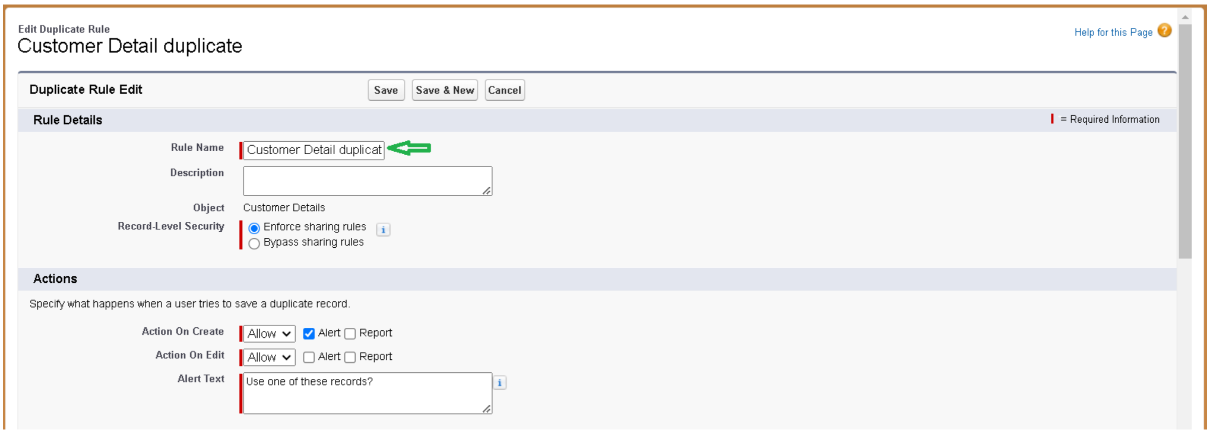
3. Give the Rule name as : Customer Detail duplicate

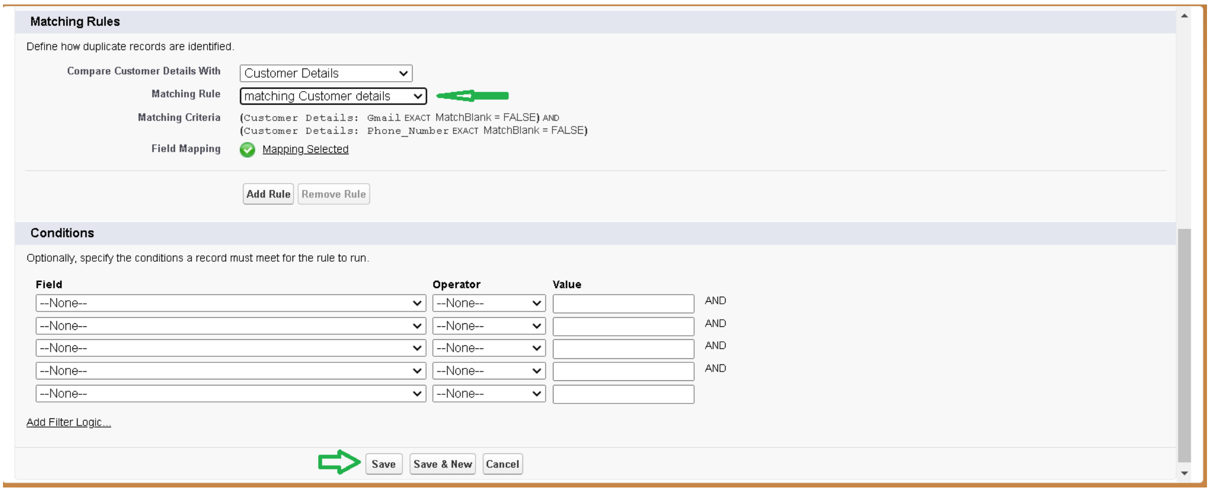
4. Scroll a little in Matching rule section

5. Select the matching rule : Matching customer details

6. And Click on save.

7. After saving the Duplicate Rule, Click on Activate.



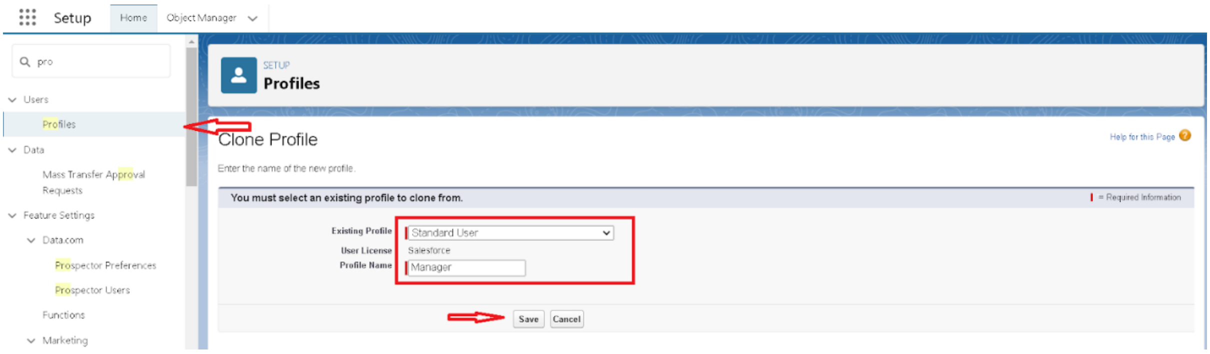


Milestone 7: SECURITY (PROFILES, ROLES & SHARING)

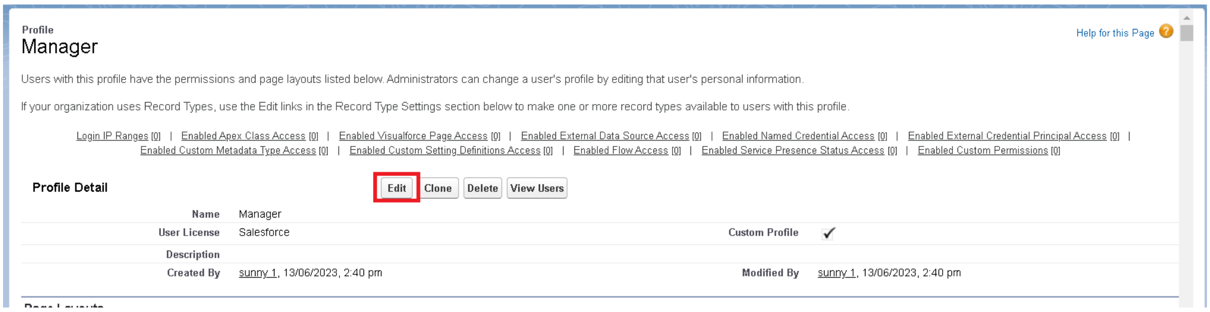
Activity 1: Create Custom Profiles

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.

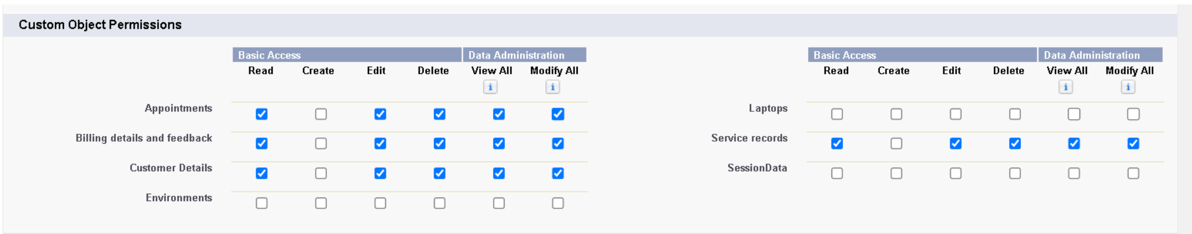


2. While still on the profile page, then click Edit.



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”.

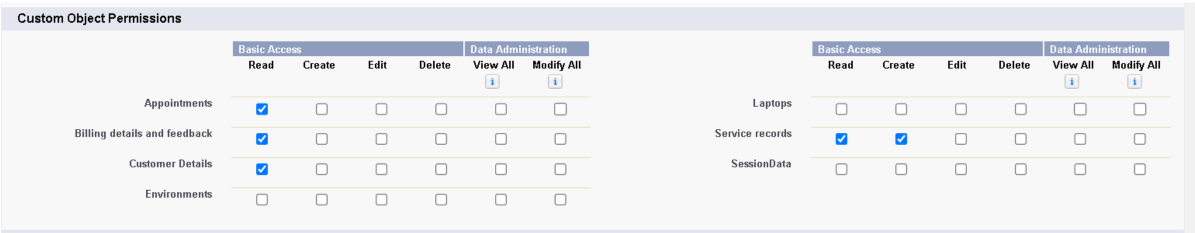
6. Change the password policies as mentioned :

7. User passwords expire in should be “ never expires ”.

8. Minimum password length should be “ 8 ”, and click save.

sales person 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.



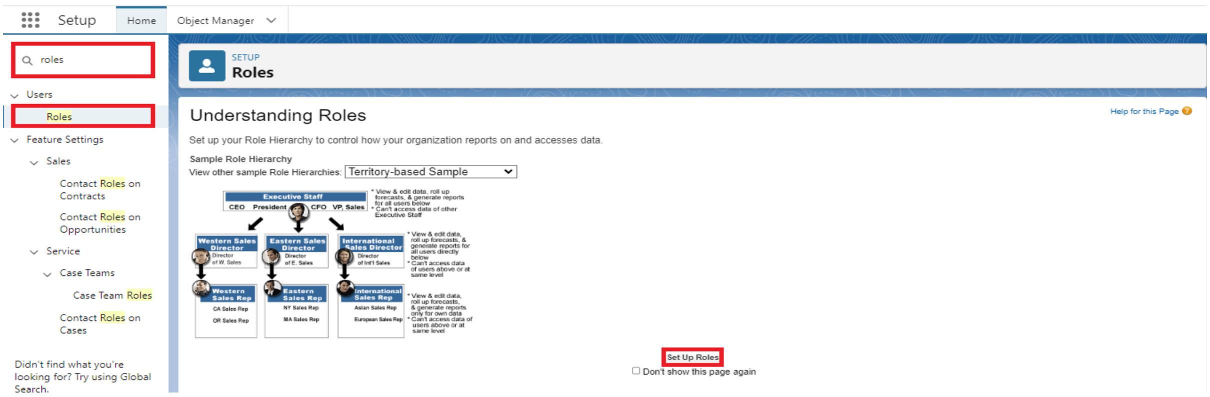
5. And click save.

Activity 2: Create Roles & Public Groups

Creating Manager Role:

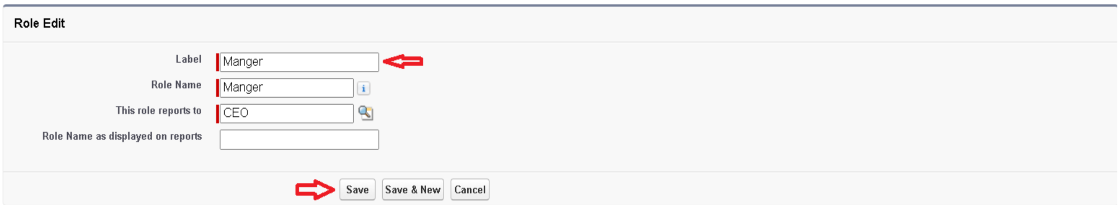
Creating Manager Role:

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.

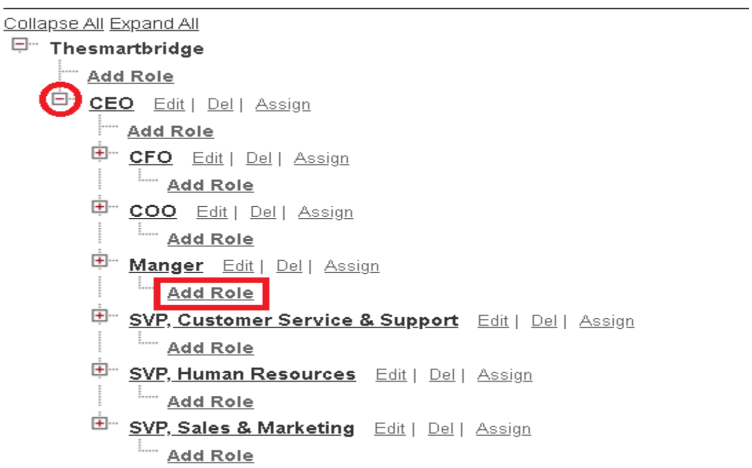
3. Give Label as "Manager" and Role name gets auto populated. Then click on Save.



Creating another roles

Creating another two roles under manager

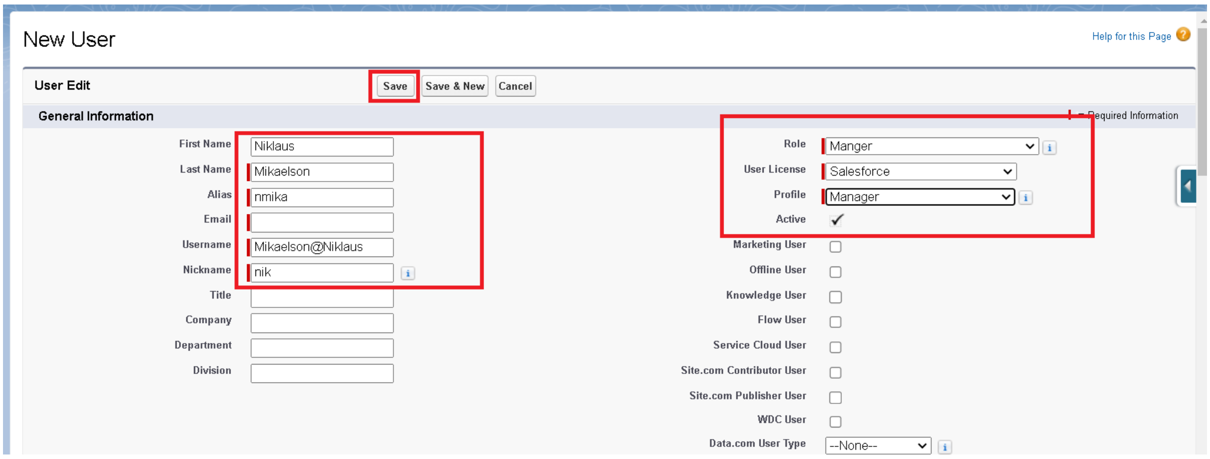
1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click plus on CEO role, and click add role under manager.



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

Create User:

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.

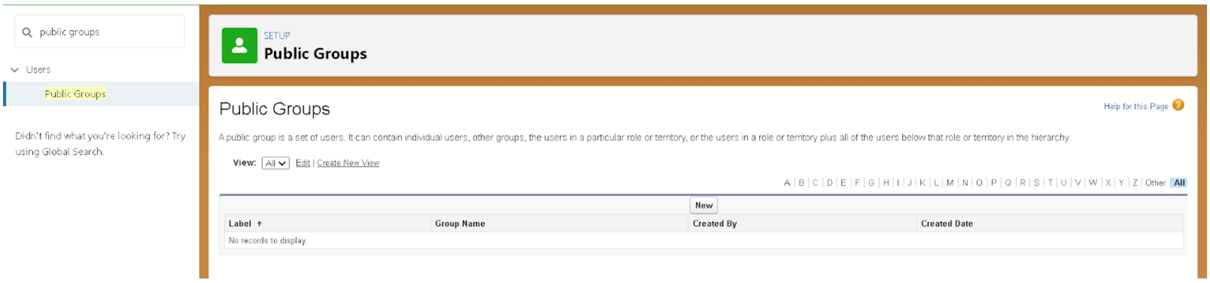
creating another users

1. Repeat the steps and create another user using

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

Creating New Public Group:

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



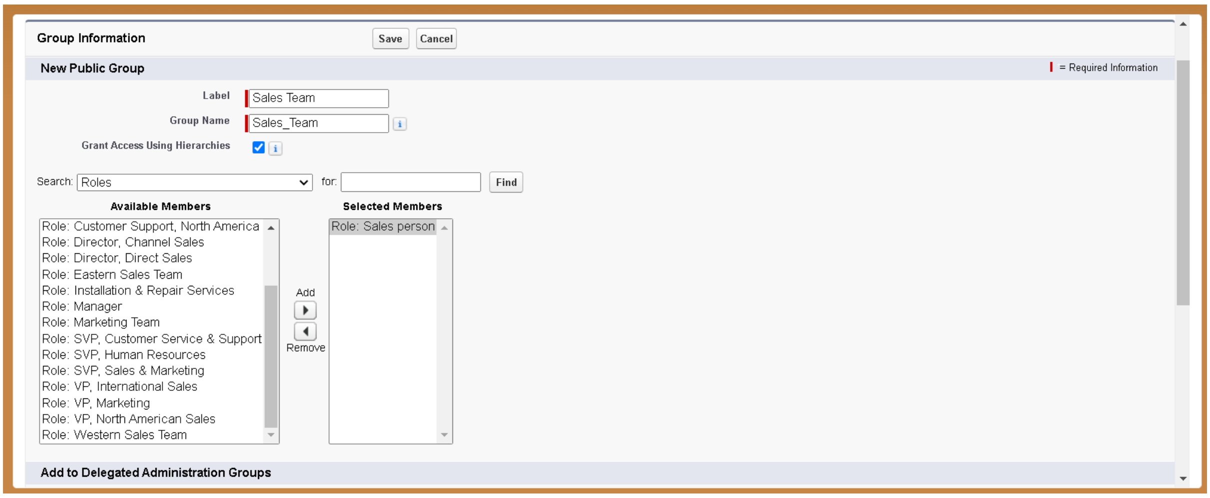
2. Give the Label as “sales team”.

3. Group name is autopopulated.

4. Search for Roles.

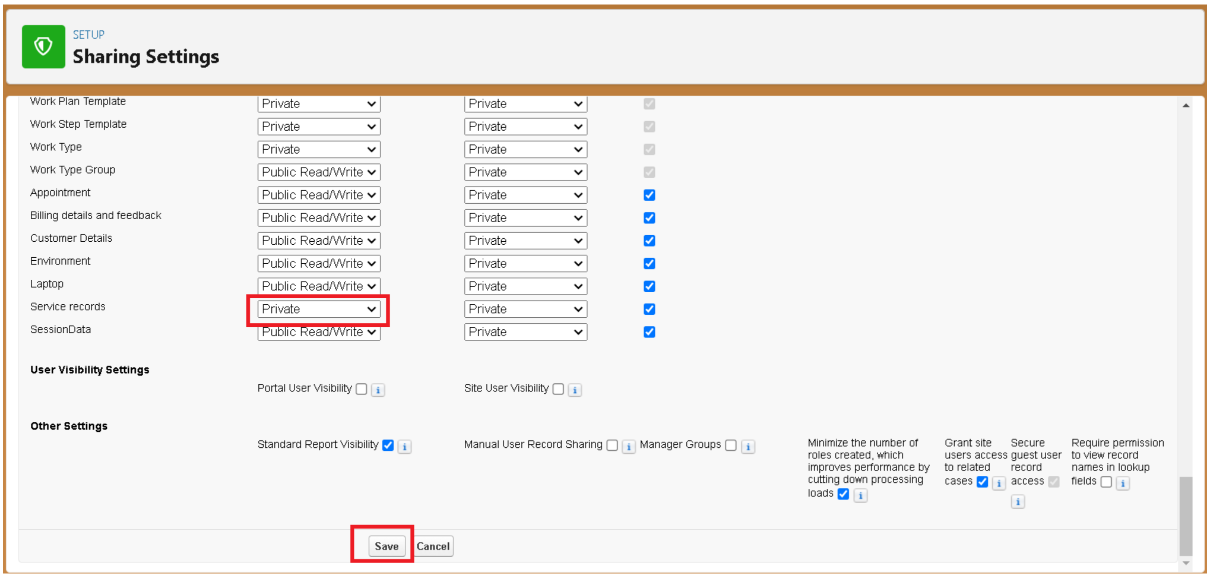
5. In Available Members select Sales person and click on add it will be moved to selected member.

6. Click on save.

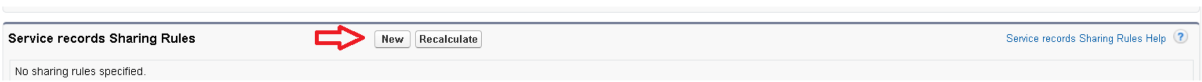


Activity 3: Configure Sharing Settings

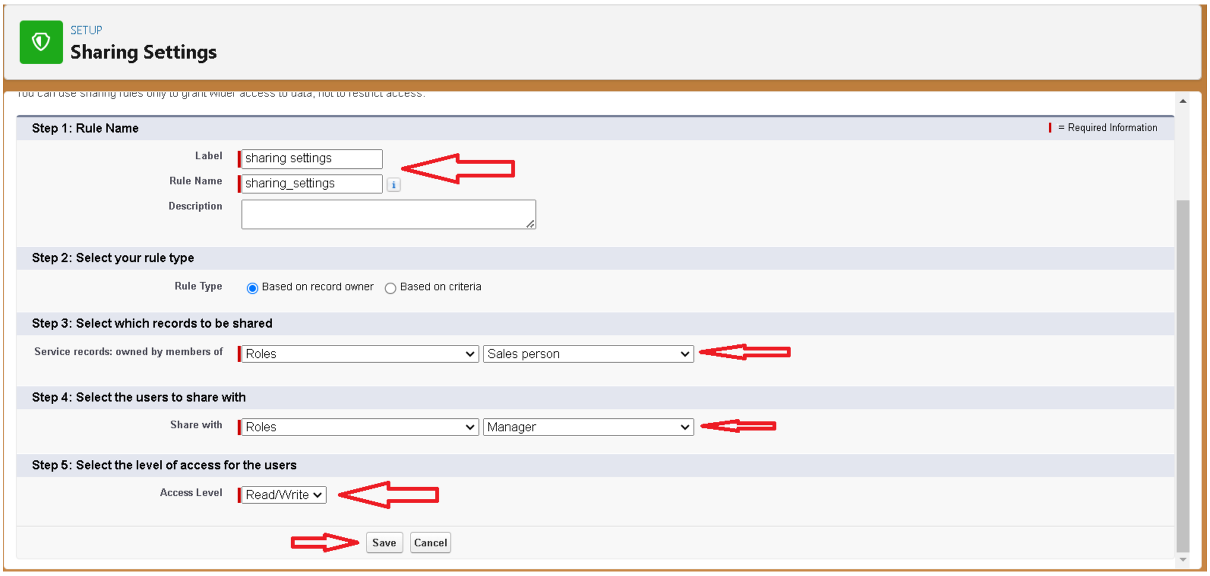
Creating Sharing settings:

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. 

4. Click on save and refresh.
5. 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.

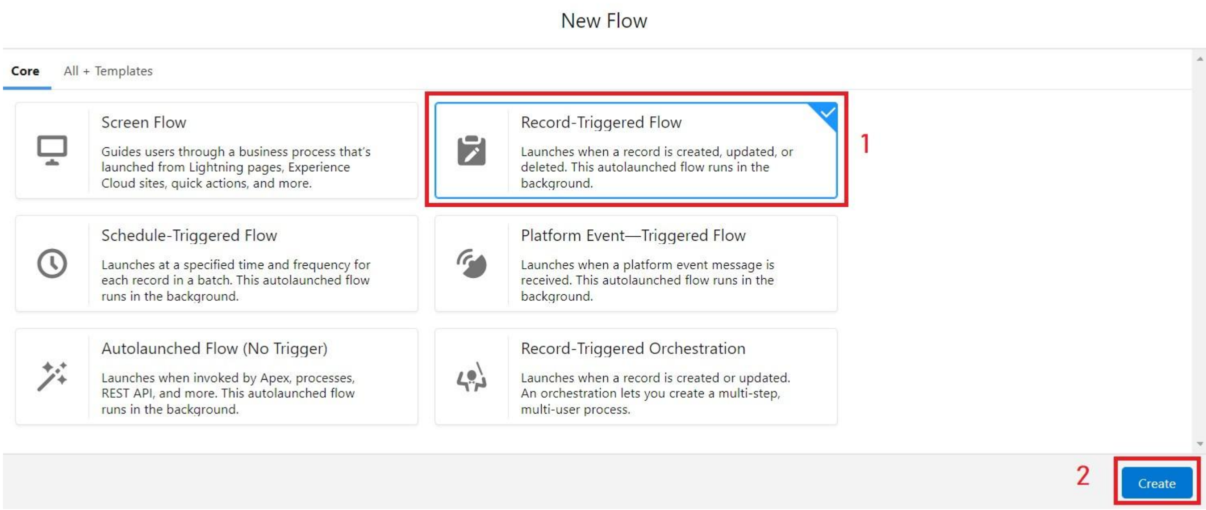


Milestone 8: AUTOMATION (FLOWS & APEX)

Activity 1: Create Flows

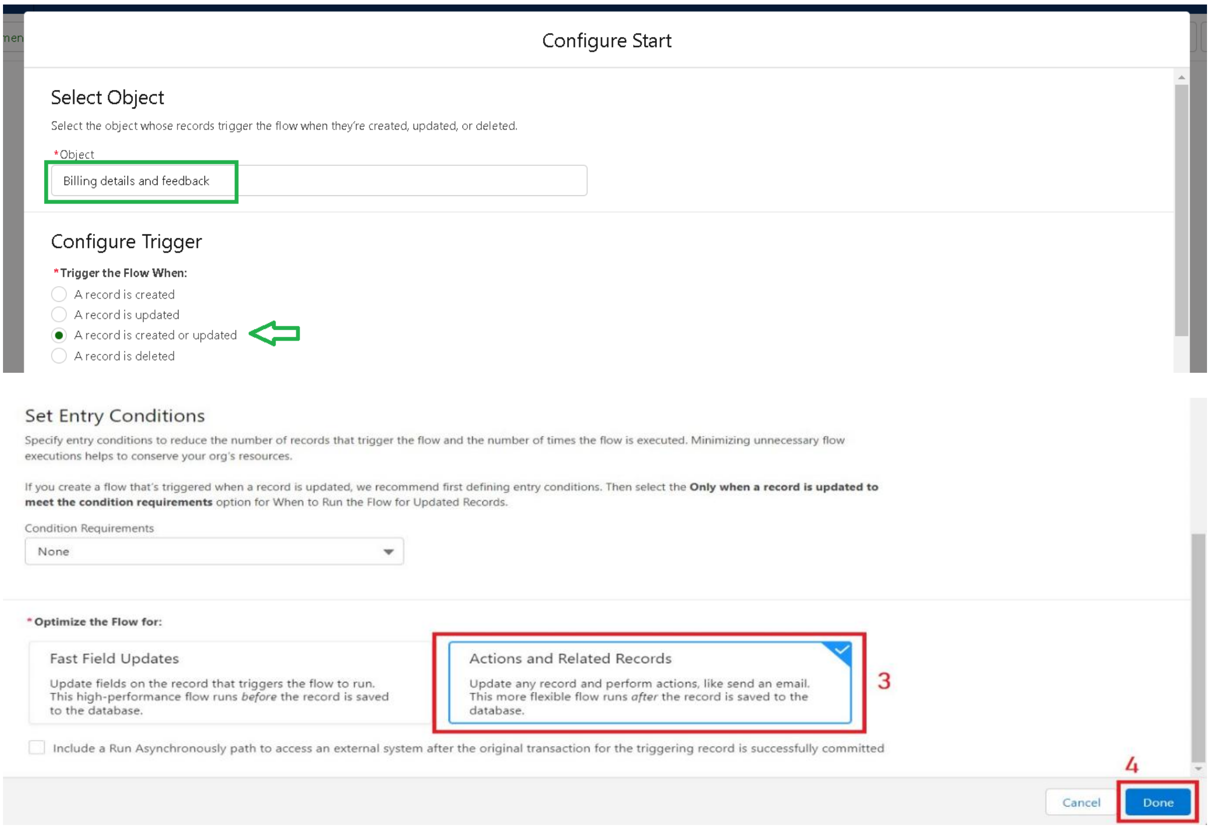
Create a Flow:

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.

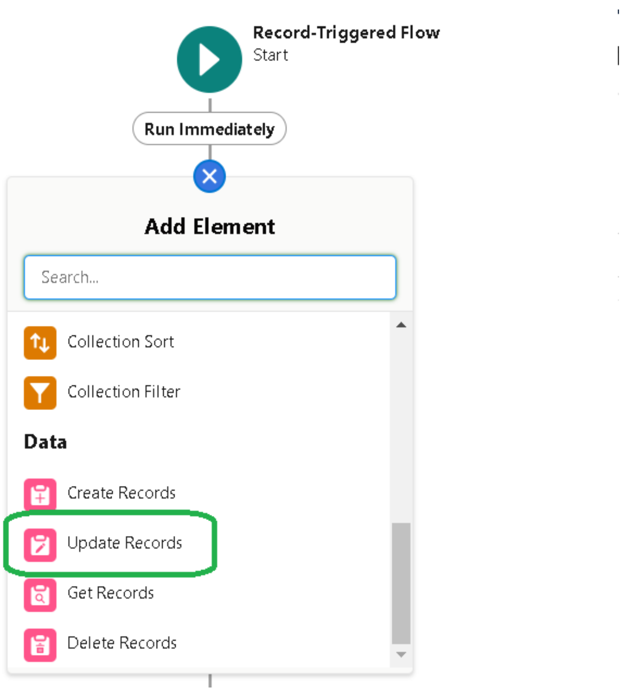


3. Select the Object as “Billing details and feedback”in the Drop down list.
4. Select the Trigger Flow when: “A record is Created or Updated”.

5. Select the Optimize the flow for: “Actions and Related Records” and Click on Done.

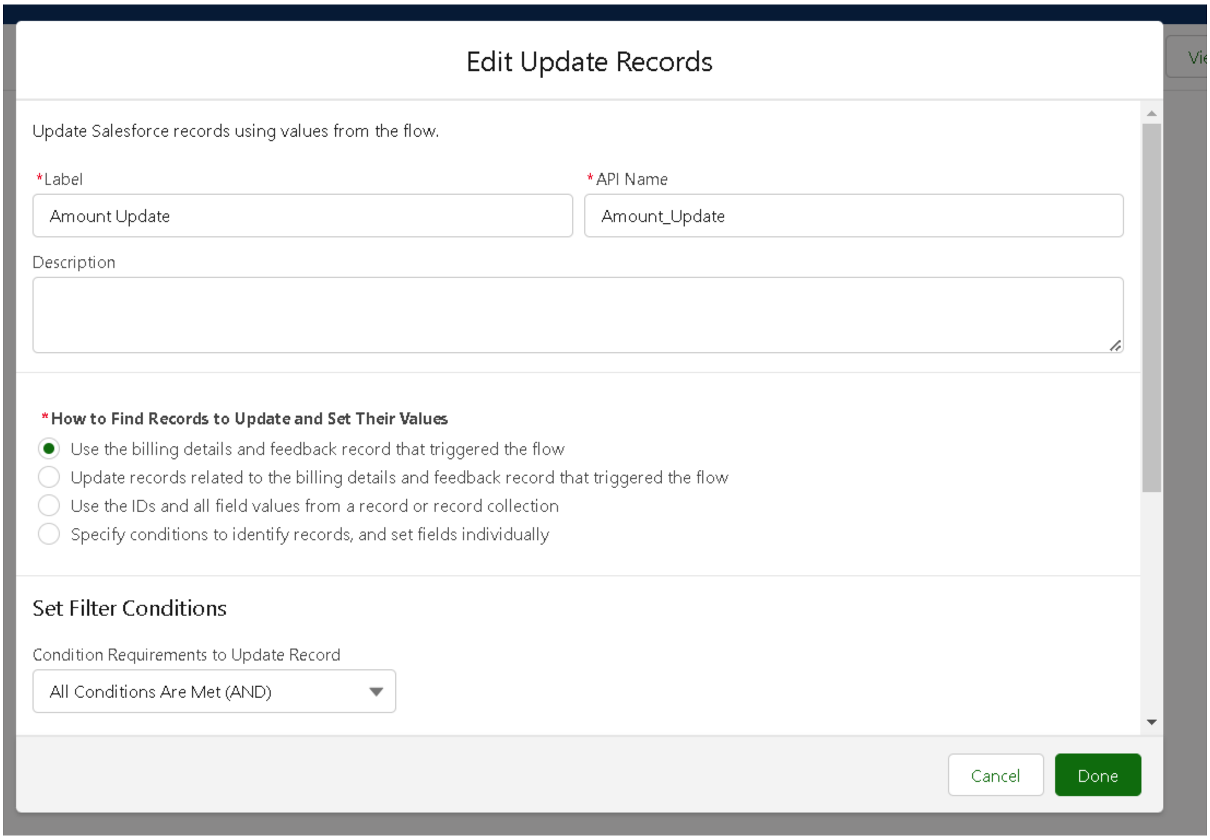


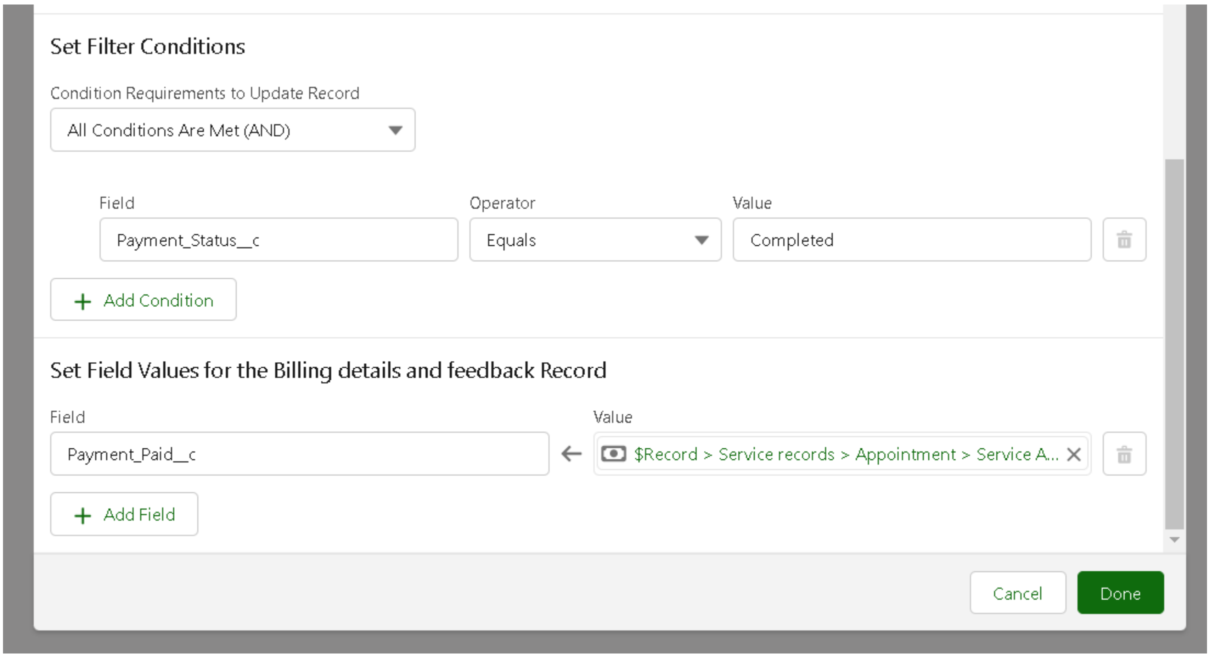
6. Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Update records Element”.



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

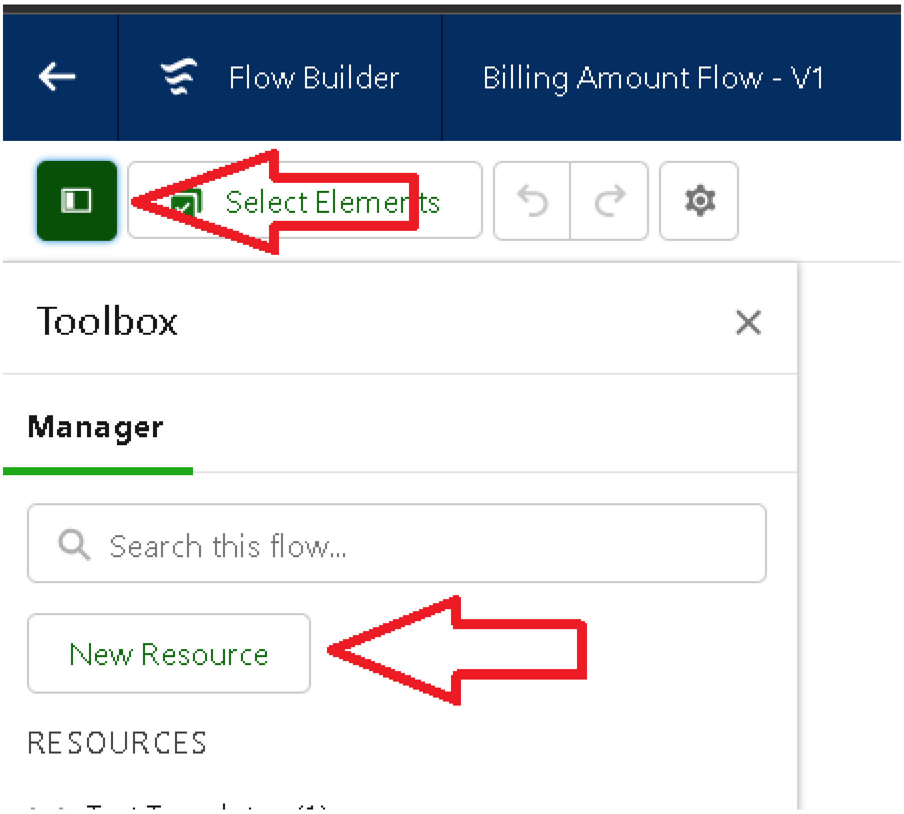
13. And Set Field Values for the Billing details and feedback Record

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.

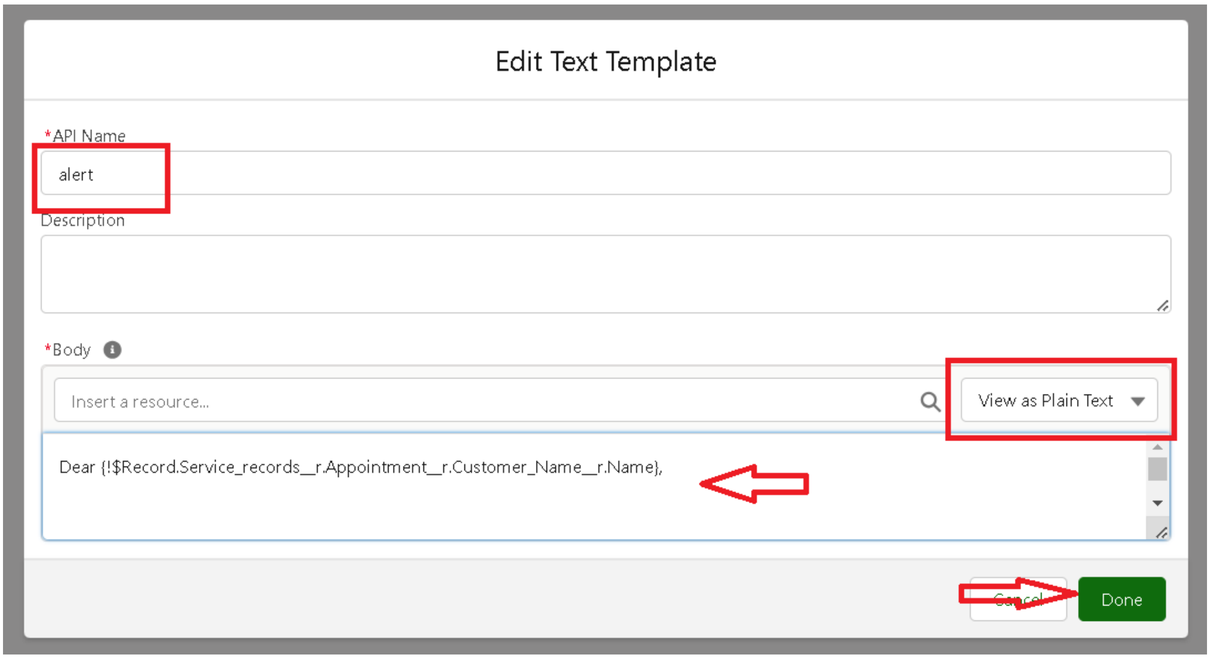
Dear {!$Record.Service_records__r.Appointment__r.Customer_Name__r.Name},

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.

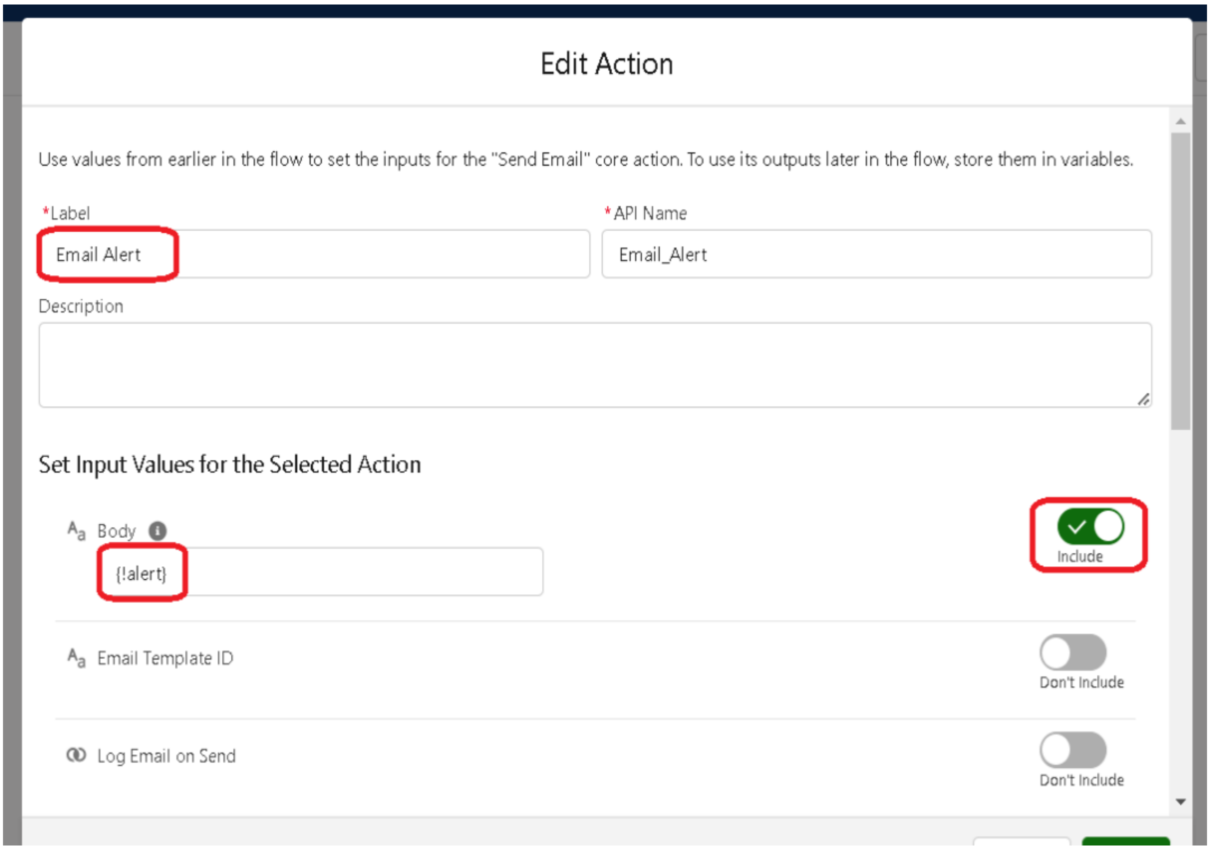
29. Select the text template that created , Body : {!alert}

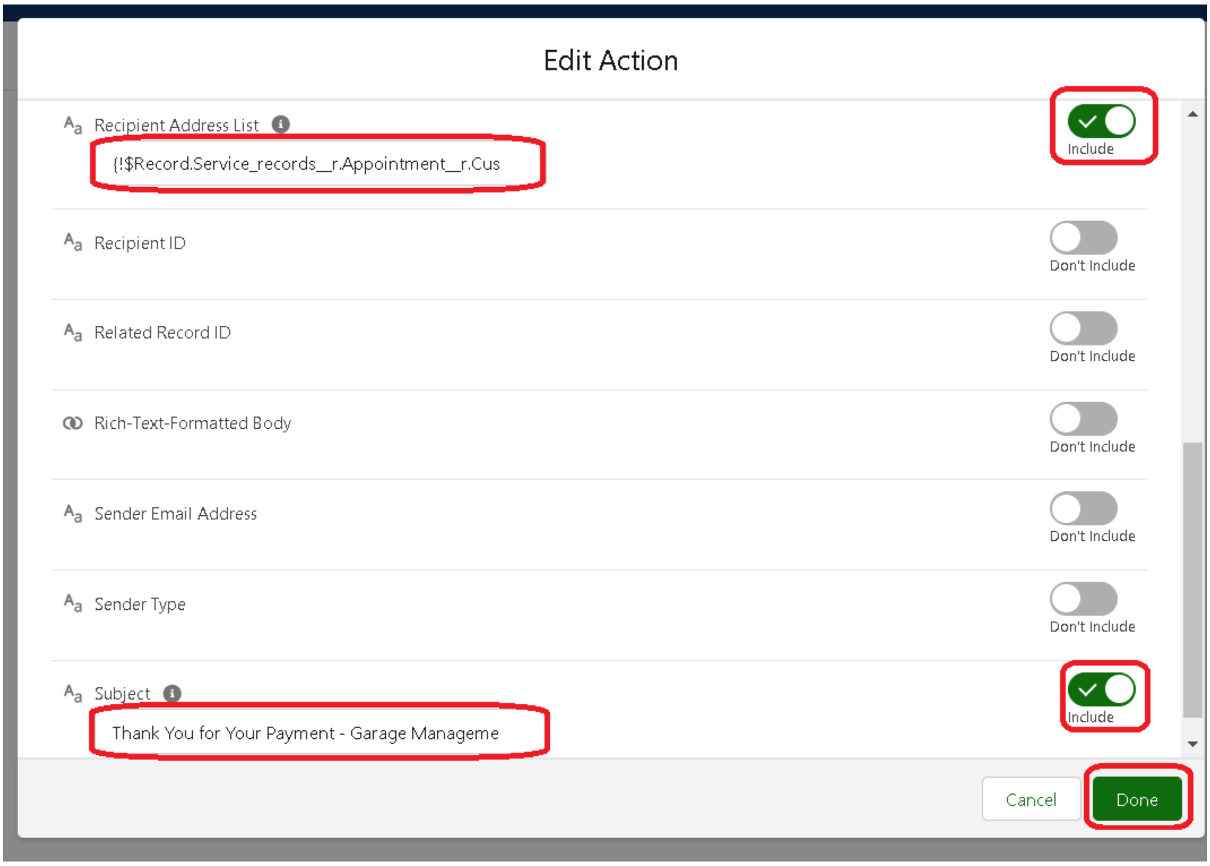
30. Include recipient address list select the email form the record.

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

32. Include subject as " Thank You for Your Payment - Garage Management".

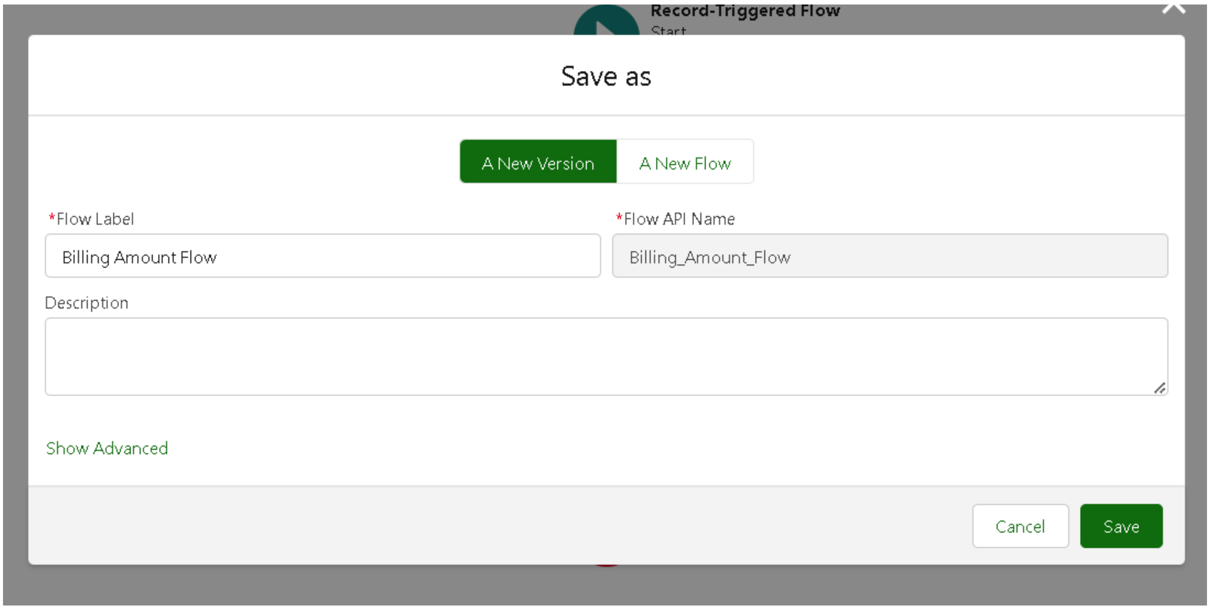
33. Click done.

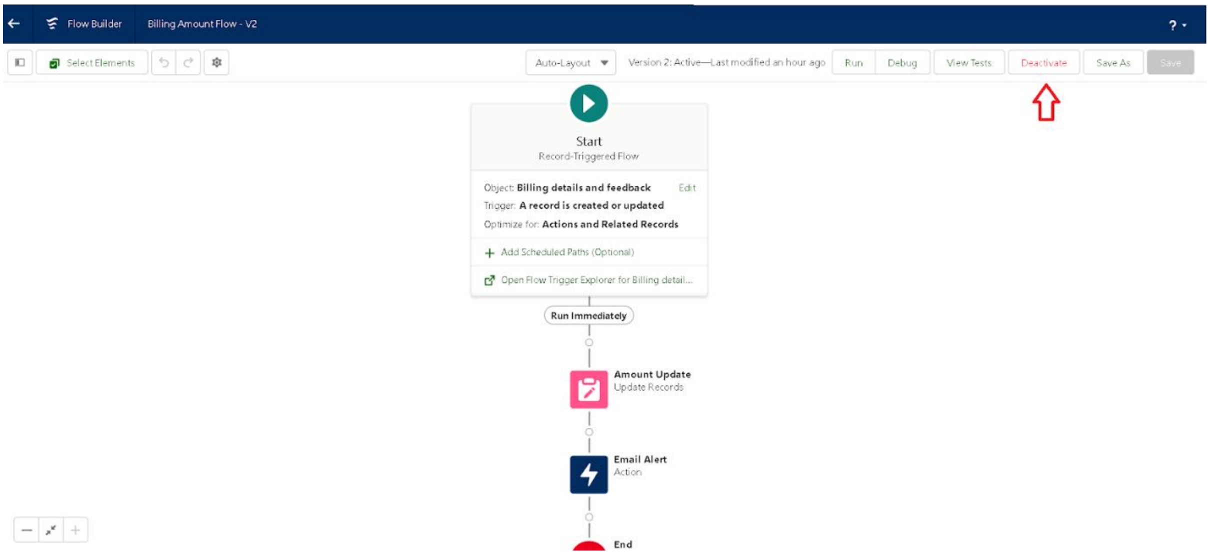




34. Click on save. Give the Flow label , Flow Api name will be autopopulated.

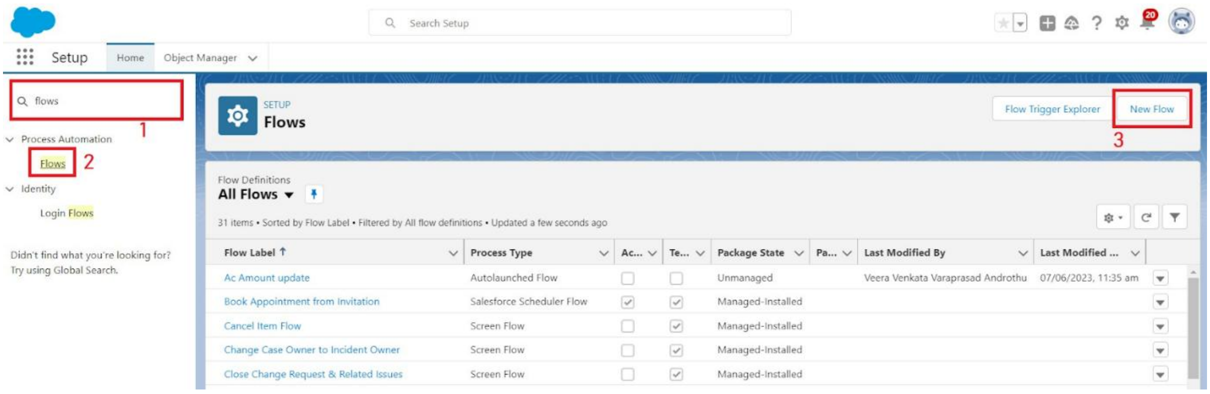
35. And click save, and click on activate.



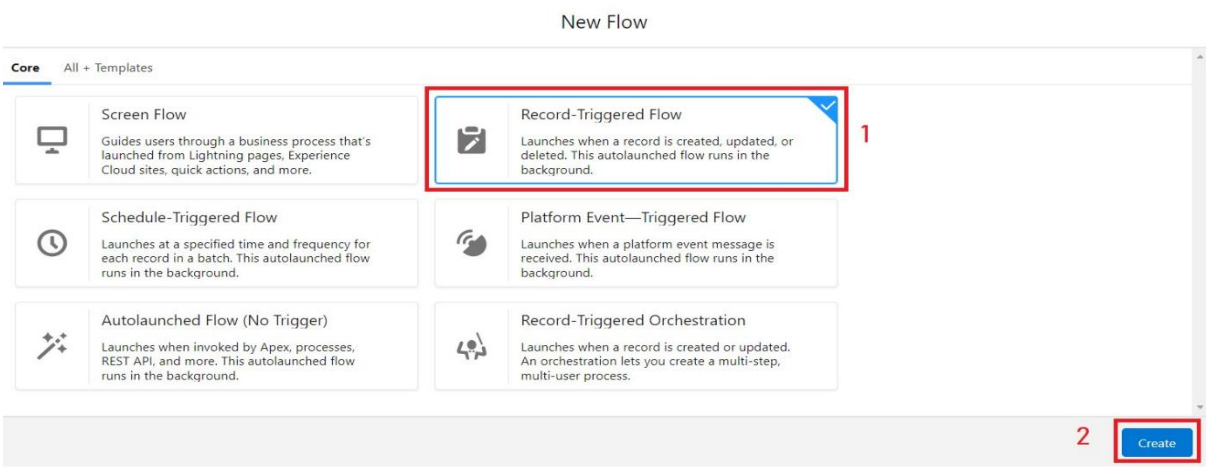


Create another Flow:

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**

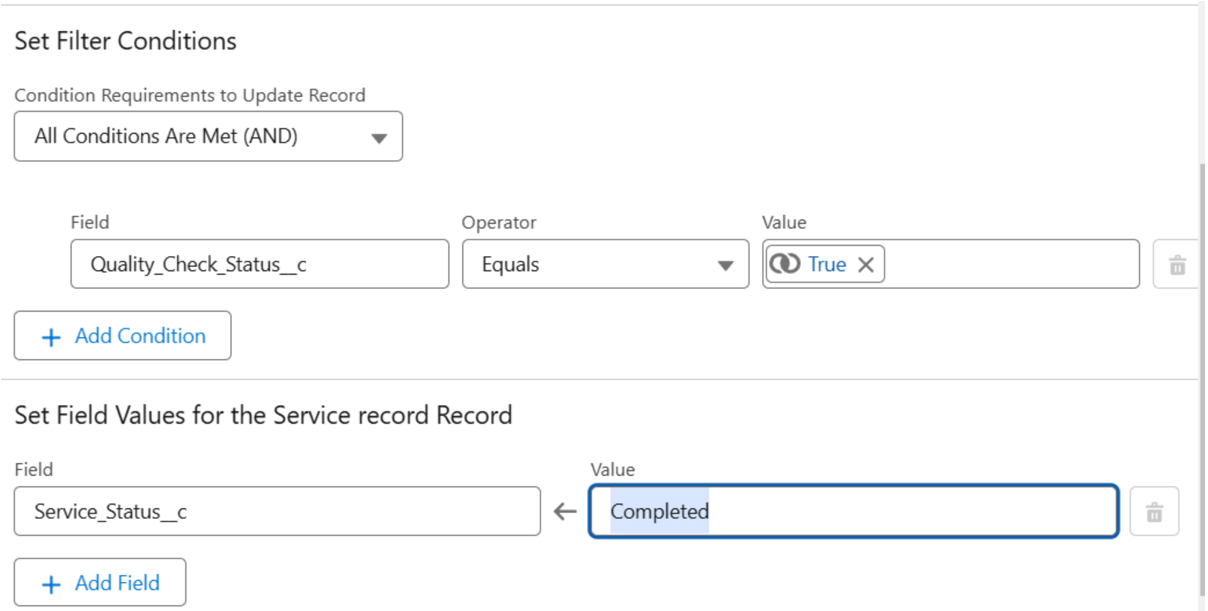
9. Operator : **Equals**

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**.

Activity 2: Create Apex Trigger

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;
```

```
            }
            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;
            }


        }
    }
}
```

Code:

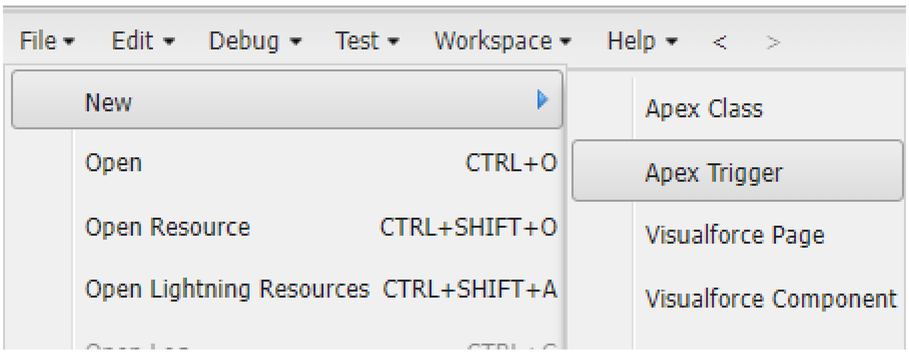
```
public class AmountDistributionHandler {
   public static void amountDist(list<Appointment__c> listApp){
      list<Service_records__c> serList = new list
<Service_records__c>();
      for(Appointment__c app : listApp){
         if(app.Maintenance_service__c == true && app.Repairs__c ==
true && app.Replacement_Parts__c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

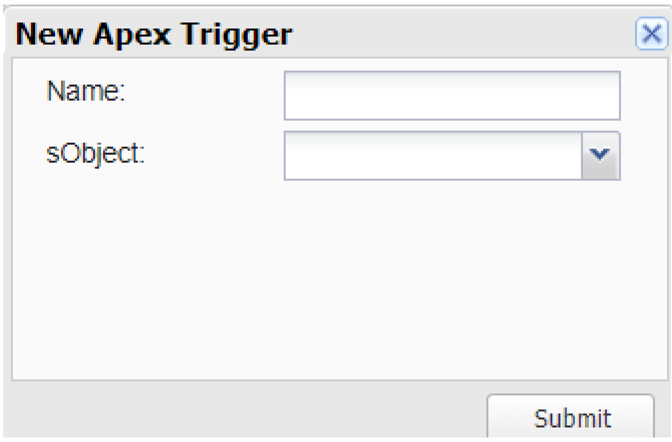
```
            else if(app.Repairs__c == true){
               app.Service_Amount__c = 3000;
            }
            else if(app.Replacement_Parts__c == true){
               app.Service_Amount__c = 5000;
            }
      }
      }
}
```

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 :

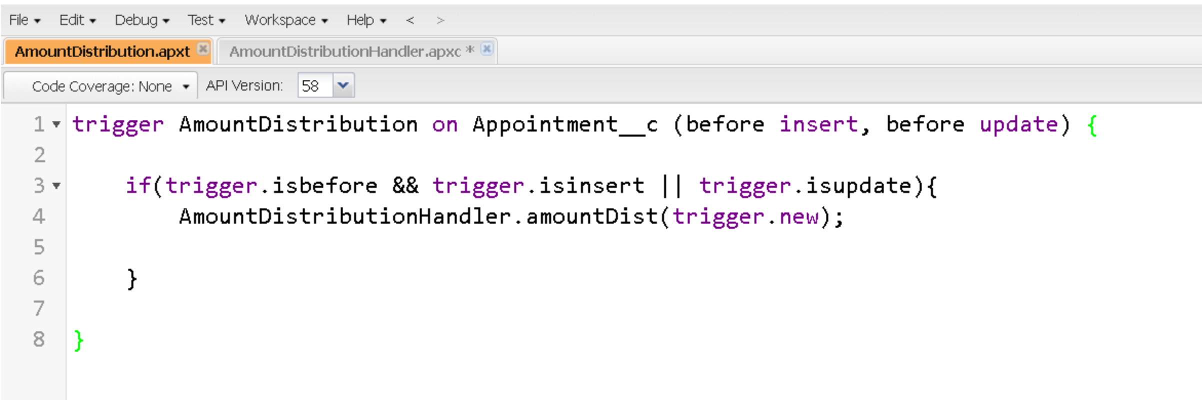
The syntax for creating trigger is :

Trigger [trigger name] on [object name](Before/After event)

{

}

Handler for the Appointment Object



Code:

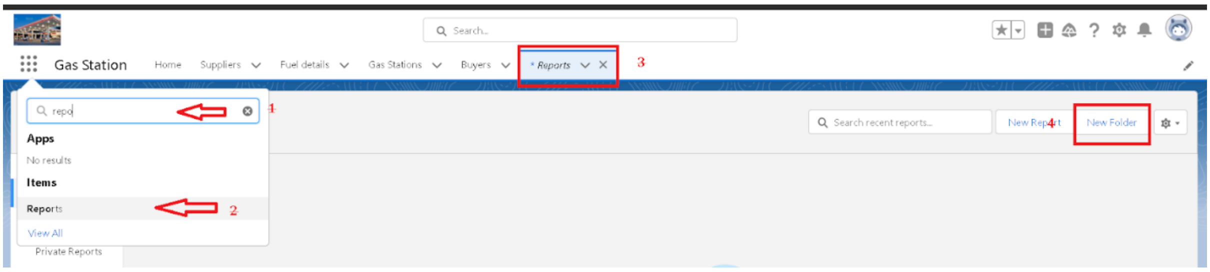
```
trigger AmountDistribution on Appointment__c (before insert, before
update) {
   if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
      AmountDistributionHandler.amountDist(trigger.new);
   }
}
```

Milestone 9: ANALYTICS (REPORTS & DASHBOARDS)

Activity 1: Create Reports

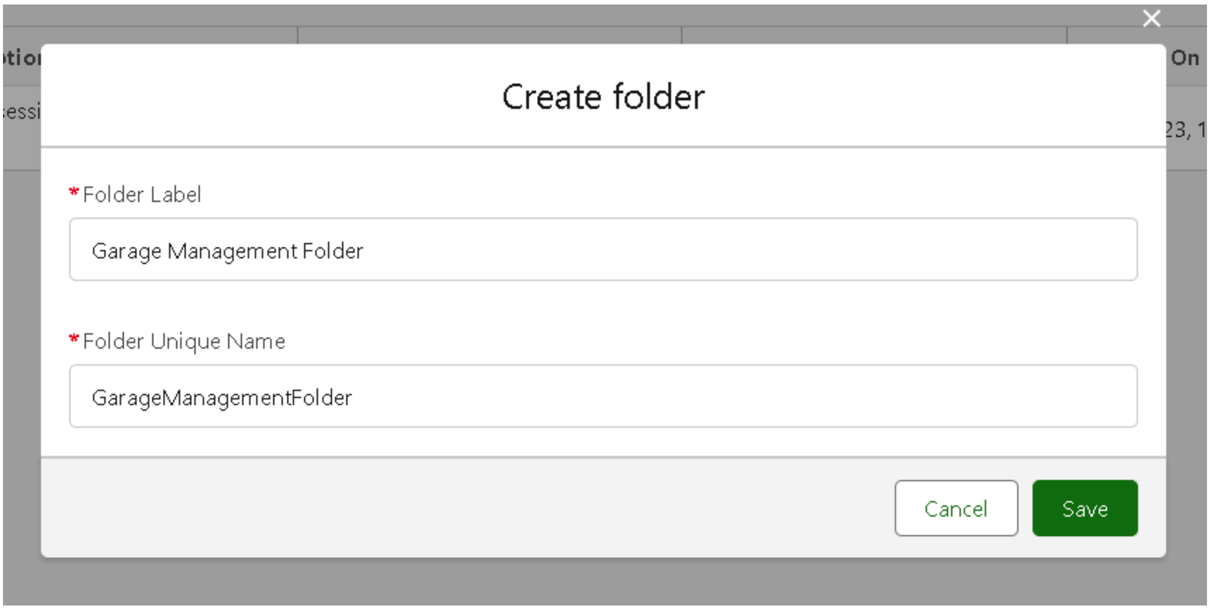
create a report folder:

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.

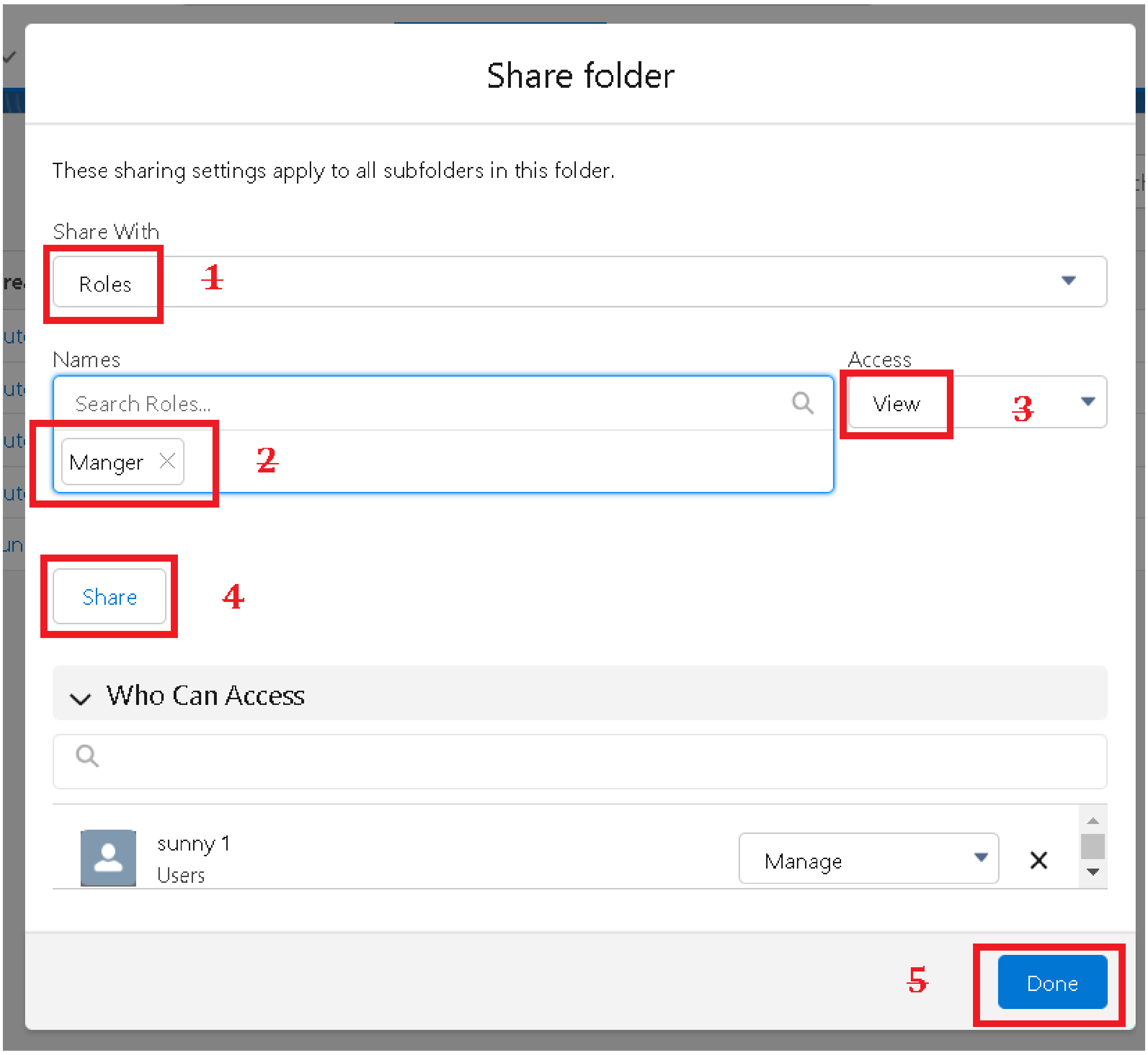
4. Click save.



Sharing a report folder

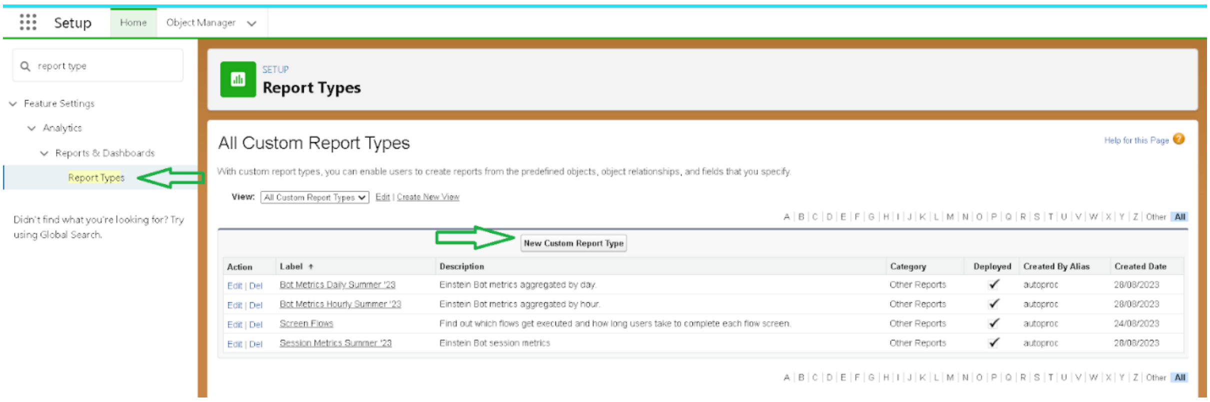
1. Go to the app >> click on the reports tab.
2. Click on the All folder , click on the Drop down arrow for Garage Management folder, and Click on share.

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.

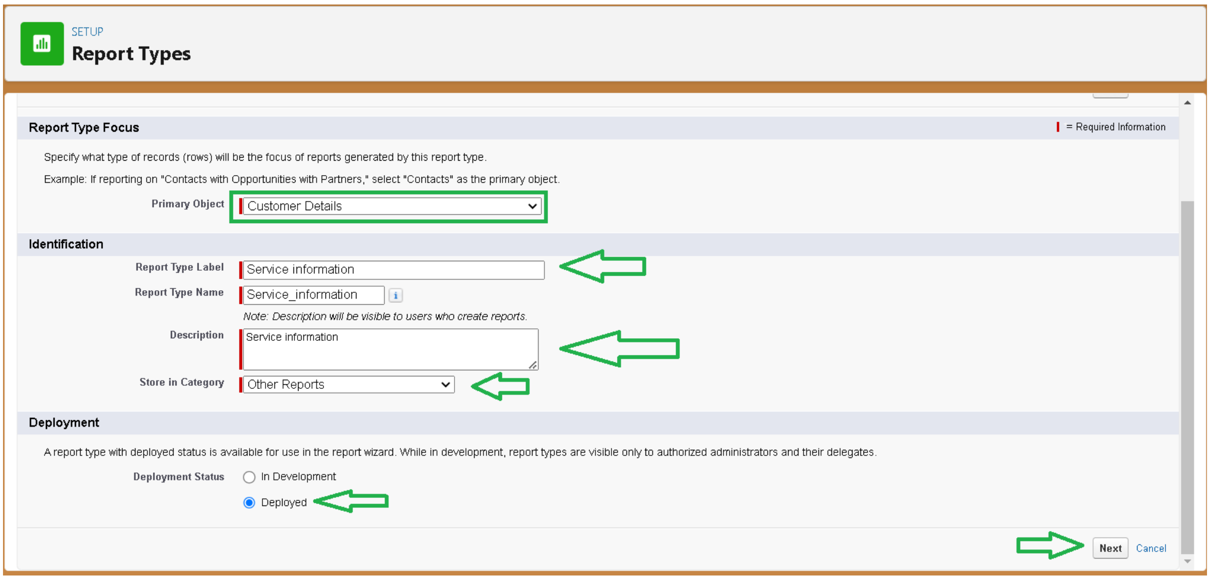


Create Report Type:

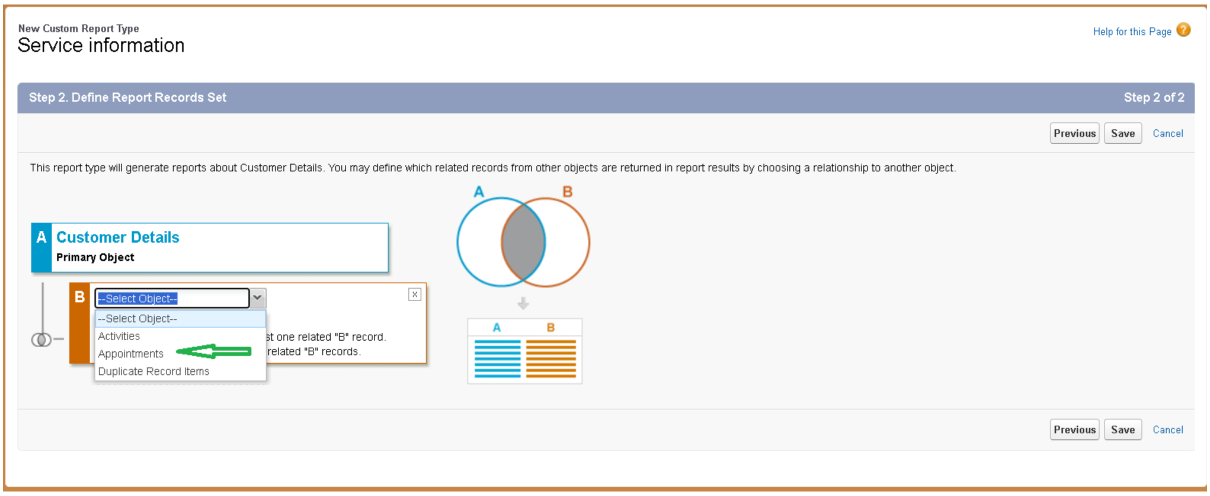
1. Go to setup >> type users in quick find box >> select Report Type >> click on Continue.
2. Click on new custom report type.

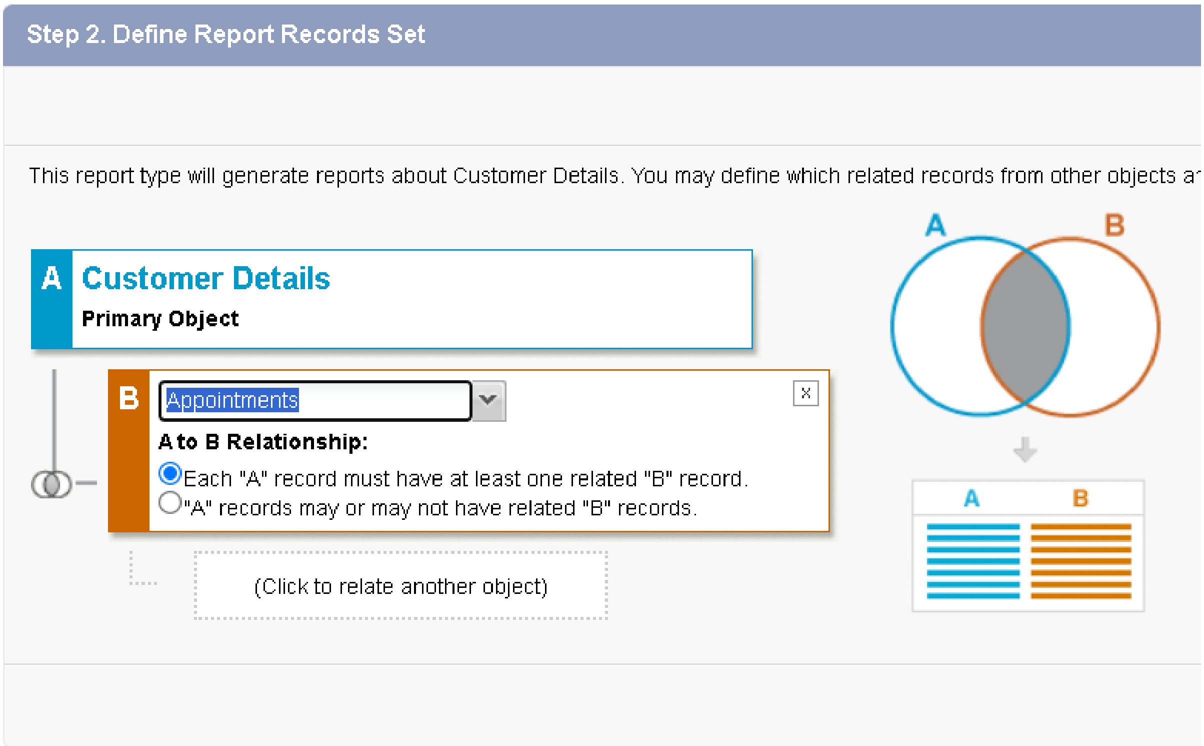


3. Select the Primary object as " Customer details" .
4. Give the Report type Label as " Service information "
5. Report type Name is autopopulated.
6. Keep the Description as same.
7. Select Store in Category as " other Reports "
8. Select the deployment status as " Depolyed ", click on Next.



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



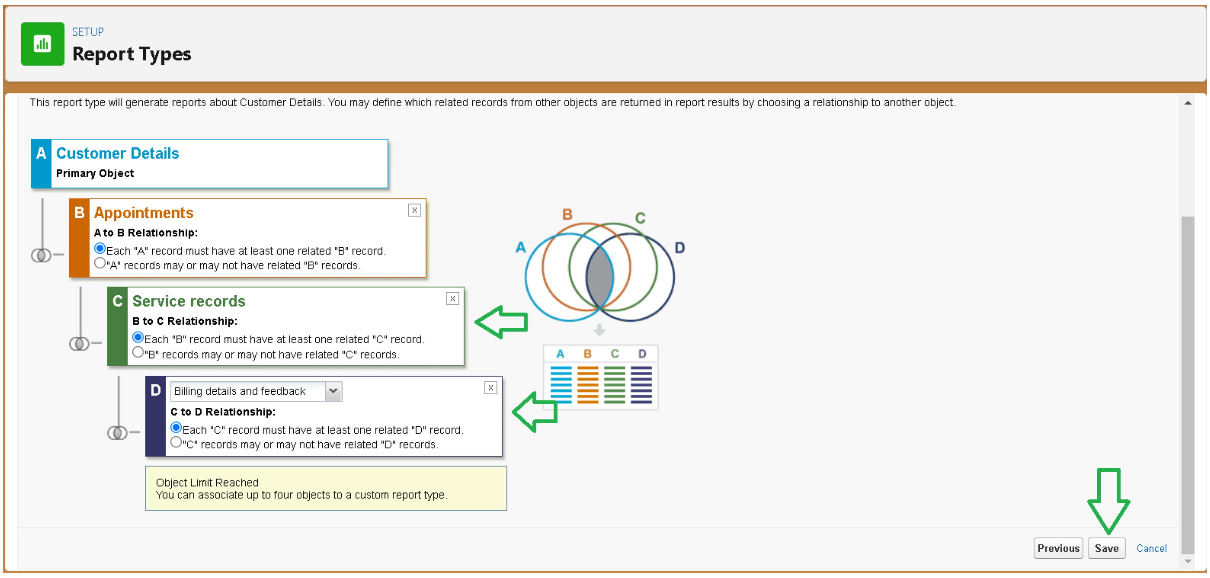


11. Again Click to relate another object.

12. And select the related object as “ service records”.

13. Repeat the process and select the related object as “ Billing details and feedback”.

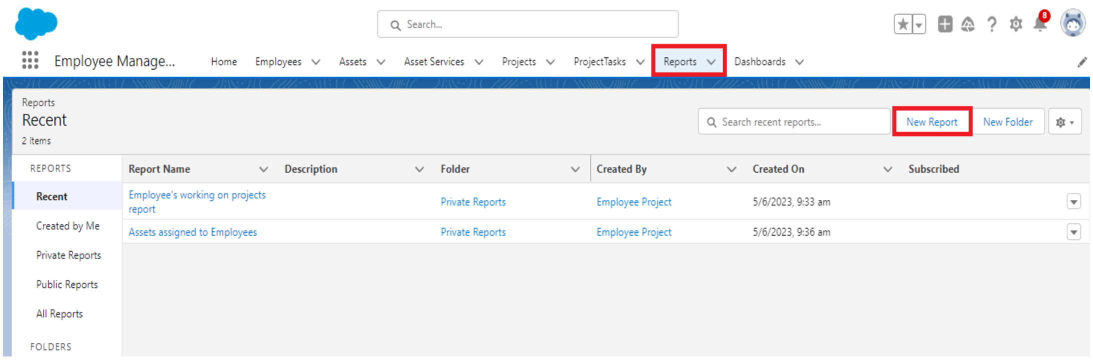
14. And click on save.



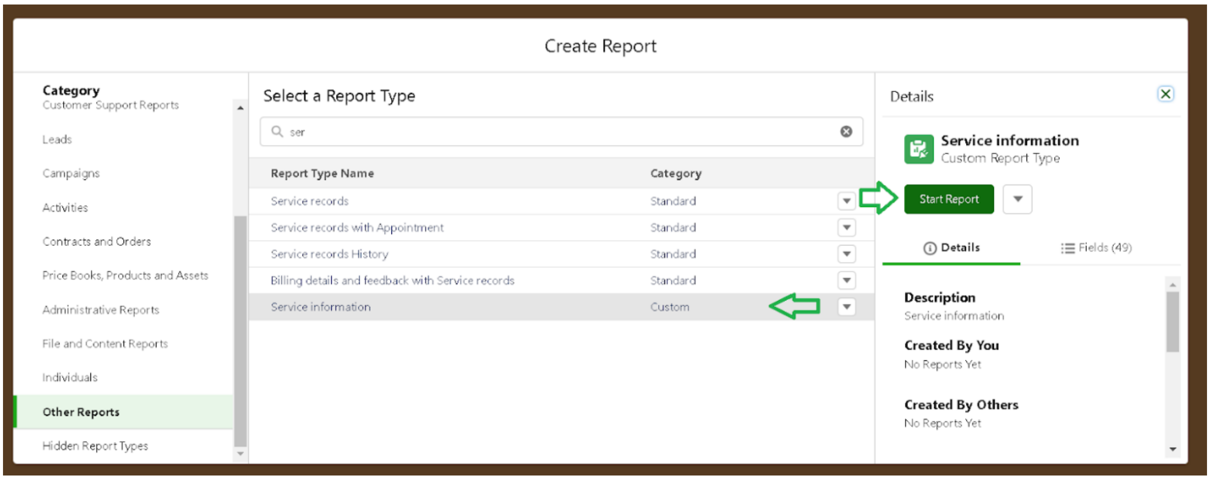
Create Report:

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 already, 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.

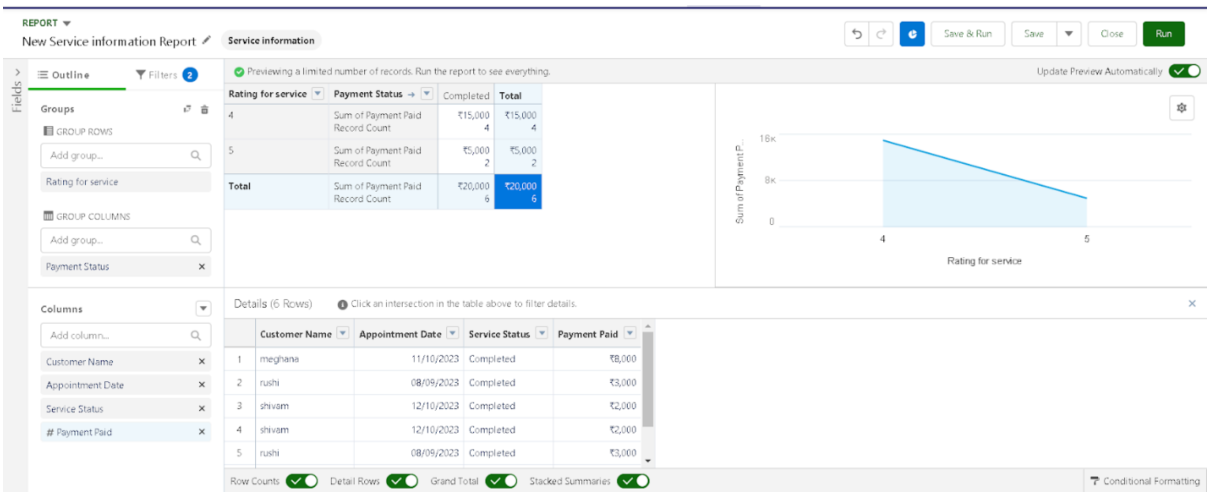
    a) Payment Status

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.

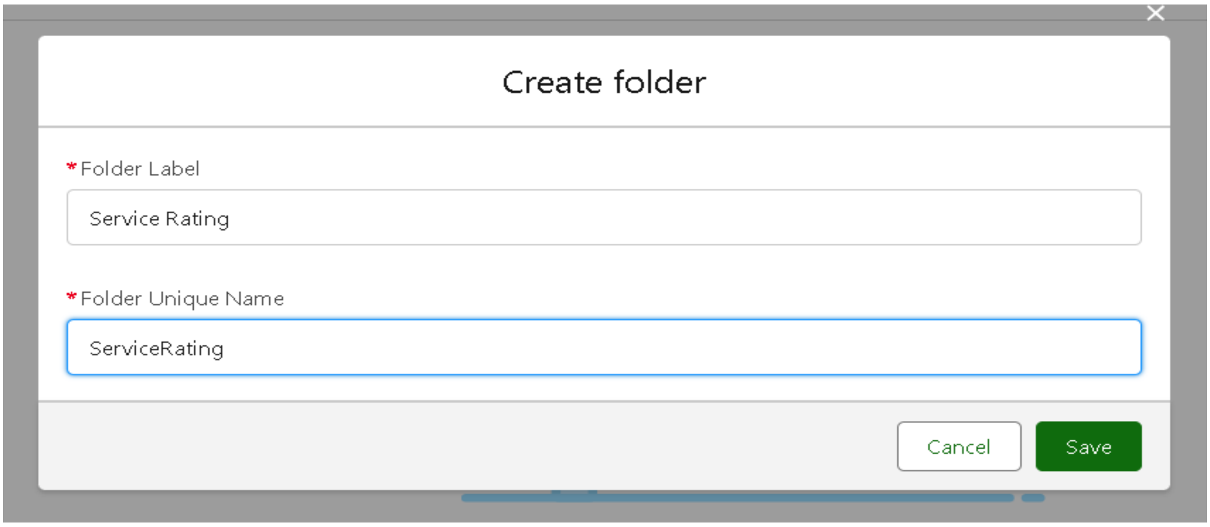
11) Select the folder the created and Click on save.



Activity 2: Create a Dashboard

Create Dashboard Folder:

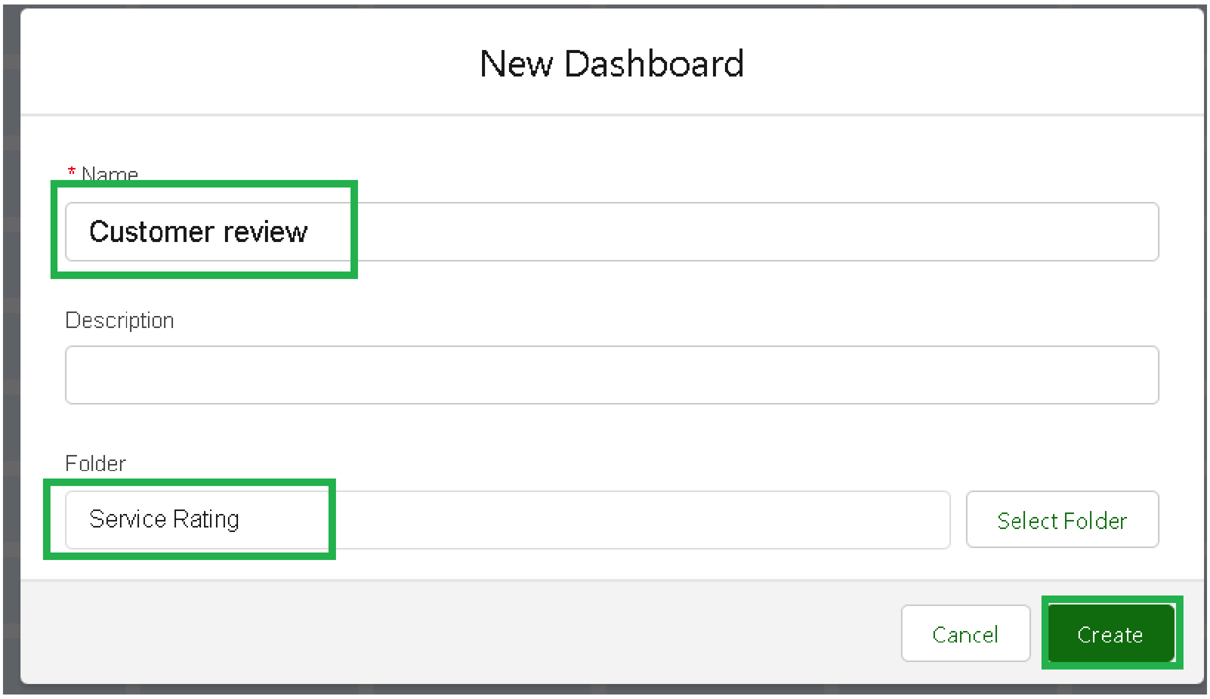
1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as “ Service Rating dashboard”.
4. Folder unique name will be auto populated.
5. Click save.



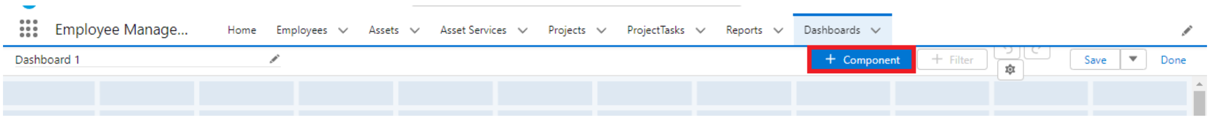
6. Follow the same steps, from Reports Milestone and Activity 2, and provide the sharing settings for the folder that was just created.

Create Dashboard:

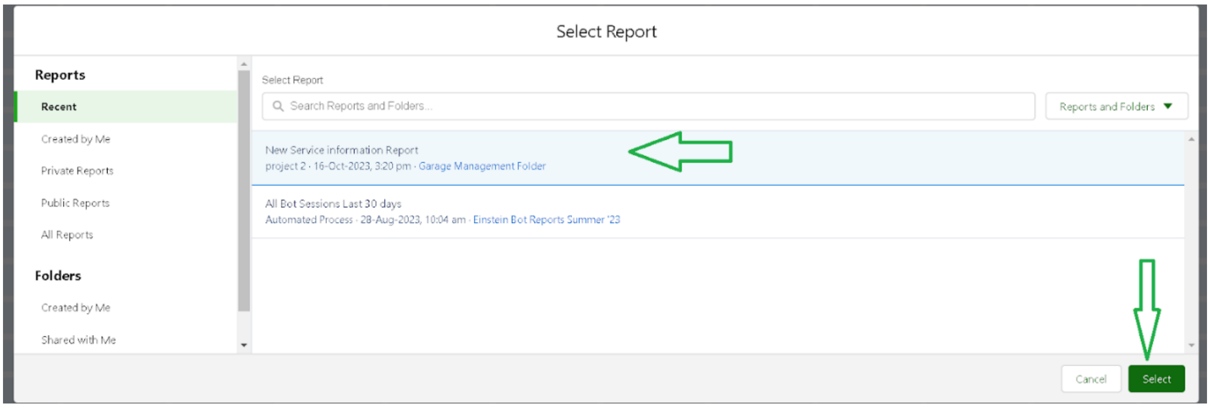
1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.



3. Select add component.

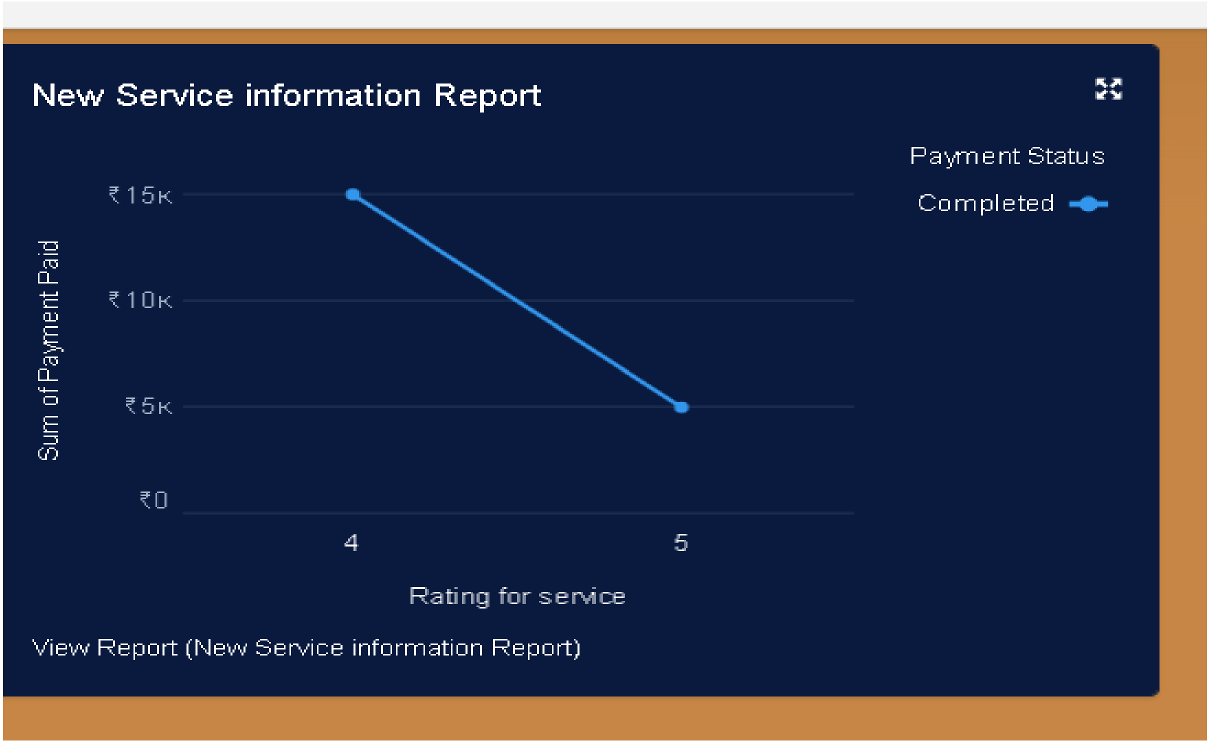


4. Select a Report and click on select.



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.

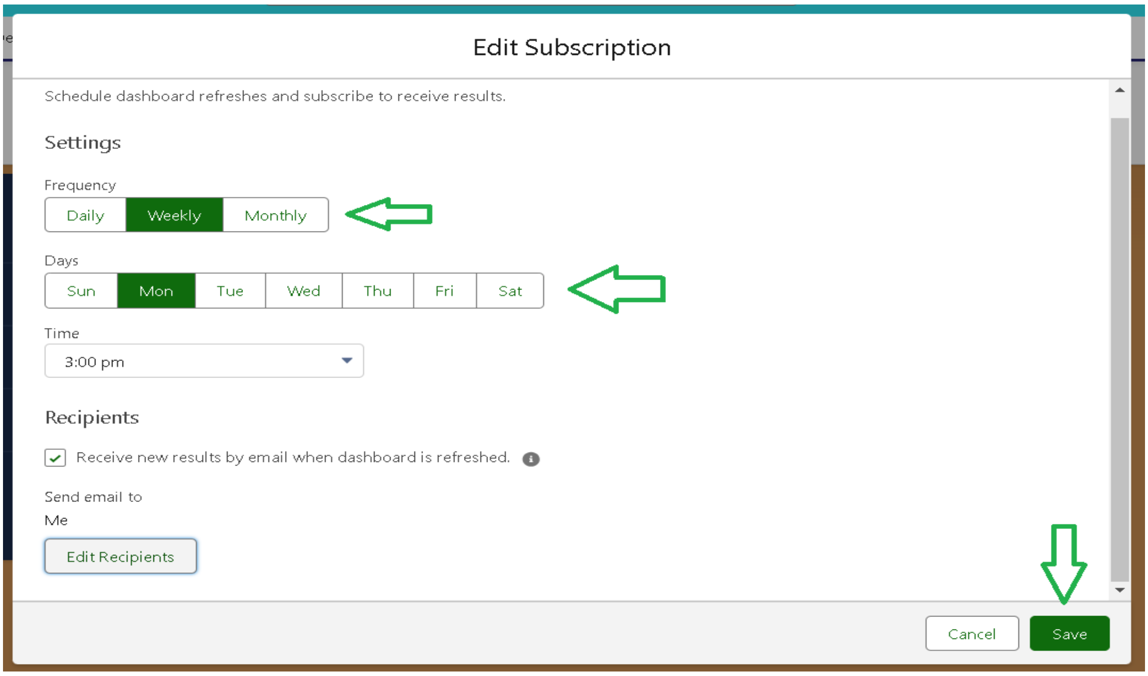


8. After that Click on Subcribe on top right.

9. Set the Frequency as “ weekly ”.

10. Set a day as monday.

11. And Click on save.

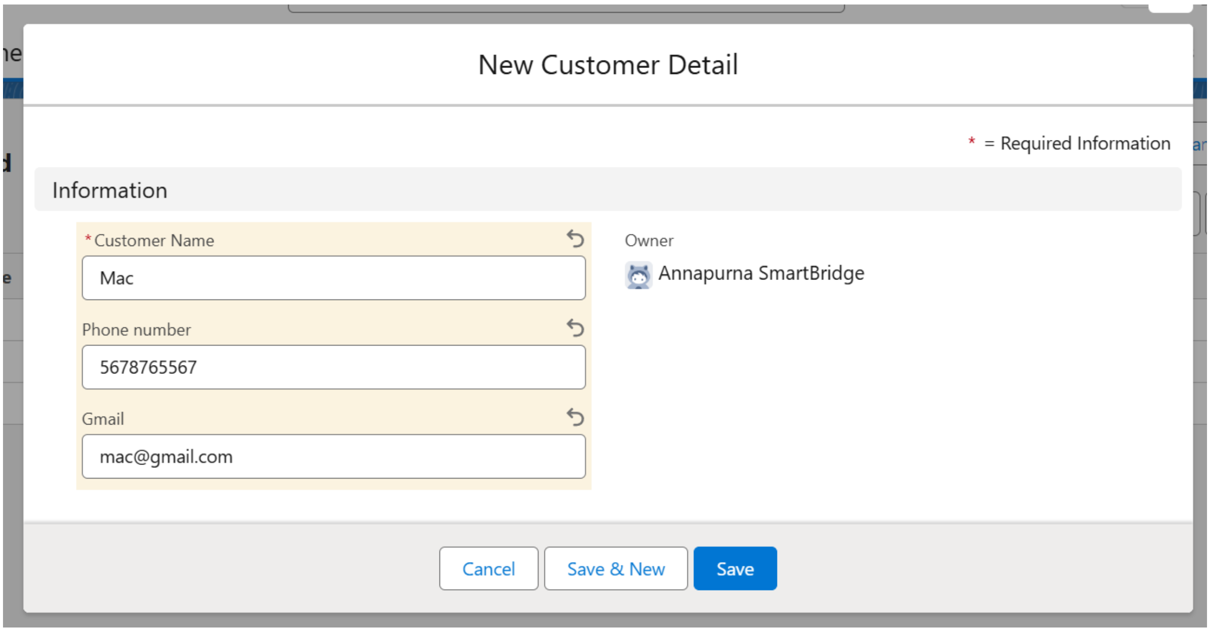


Phase 5: Deployment, Documentation & Maintenance

creating records:

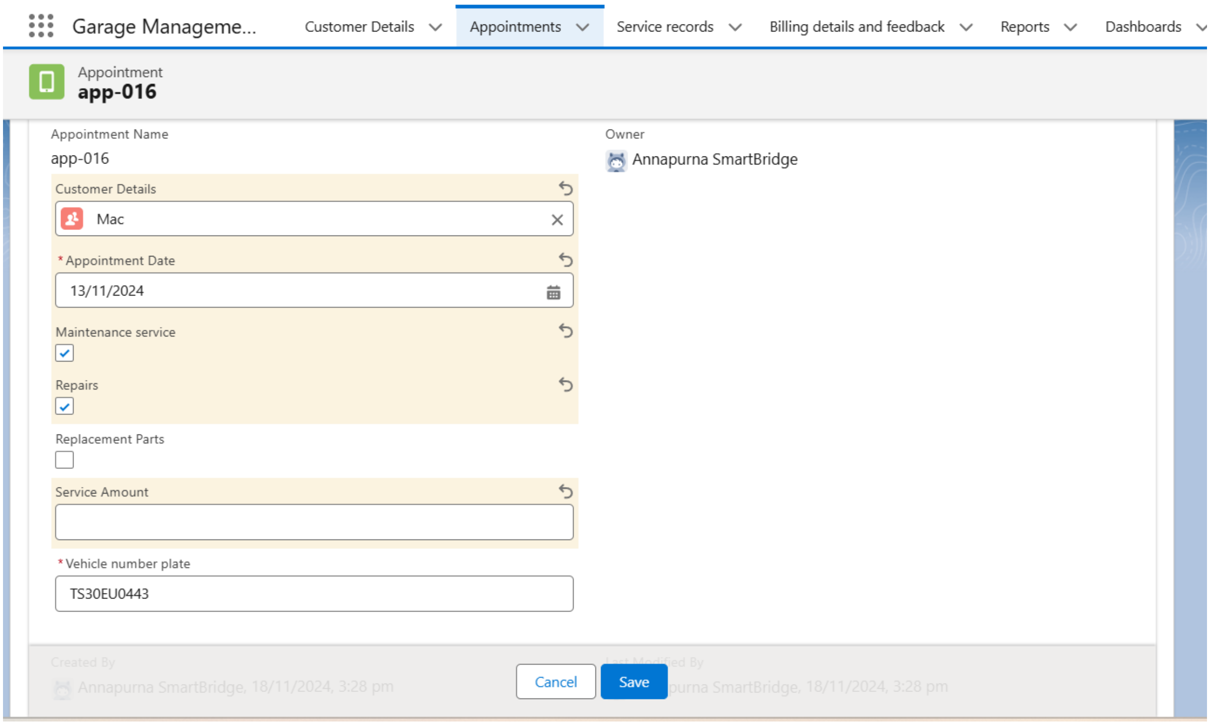
To create a record in the follow objects follow these steps

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.



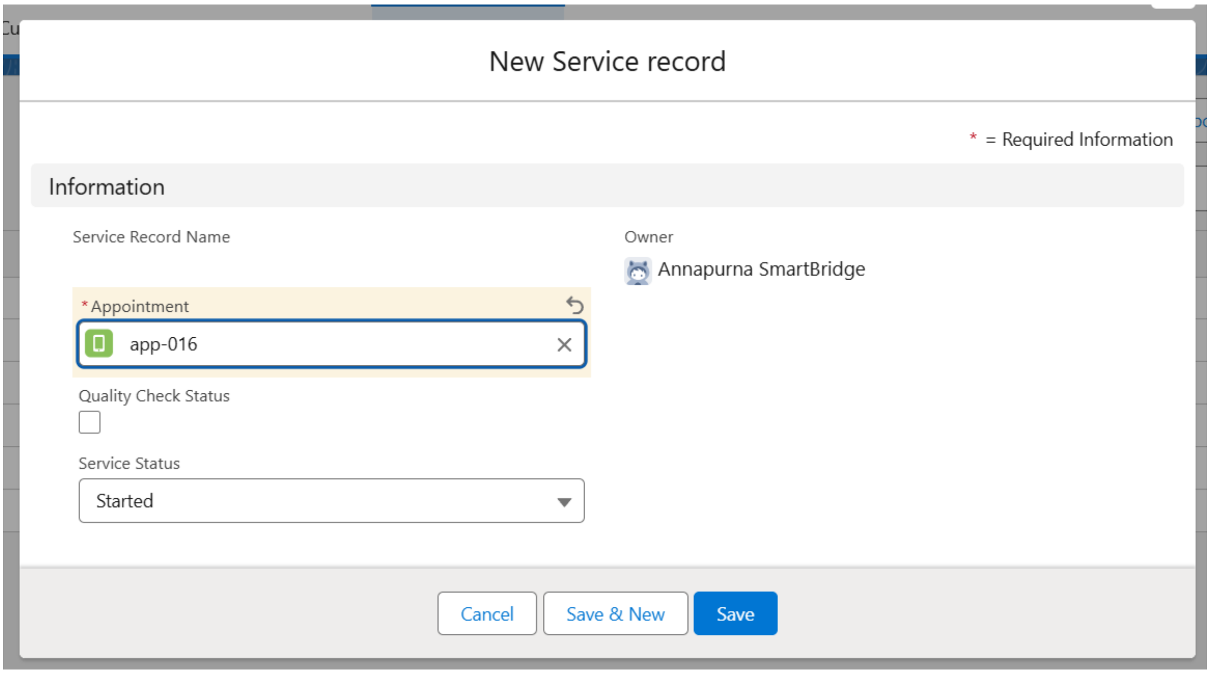
Now, Create the Appointment Record

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.



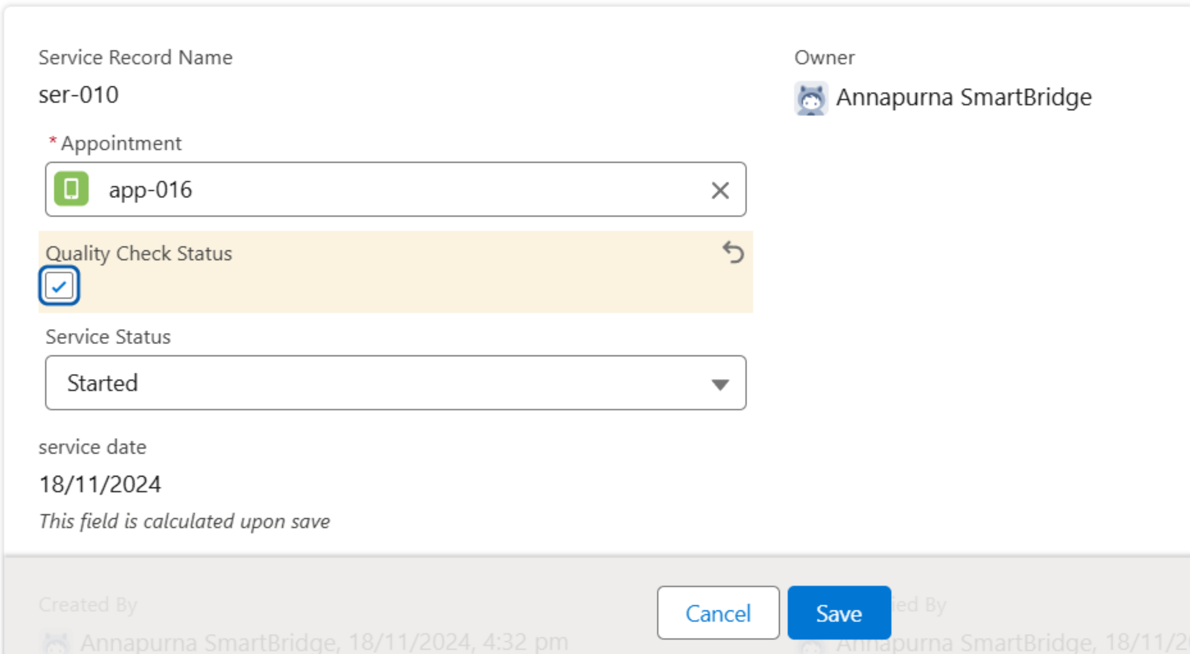
Now, Create a service Record

1. Click on the "**Service record** tab".
2. Enter the Appointment, and started is selected as default.
3. Click on save.

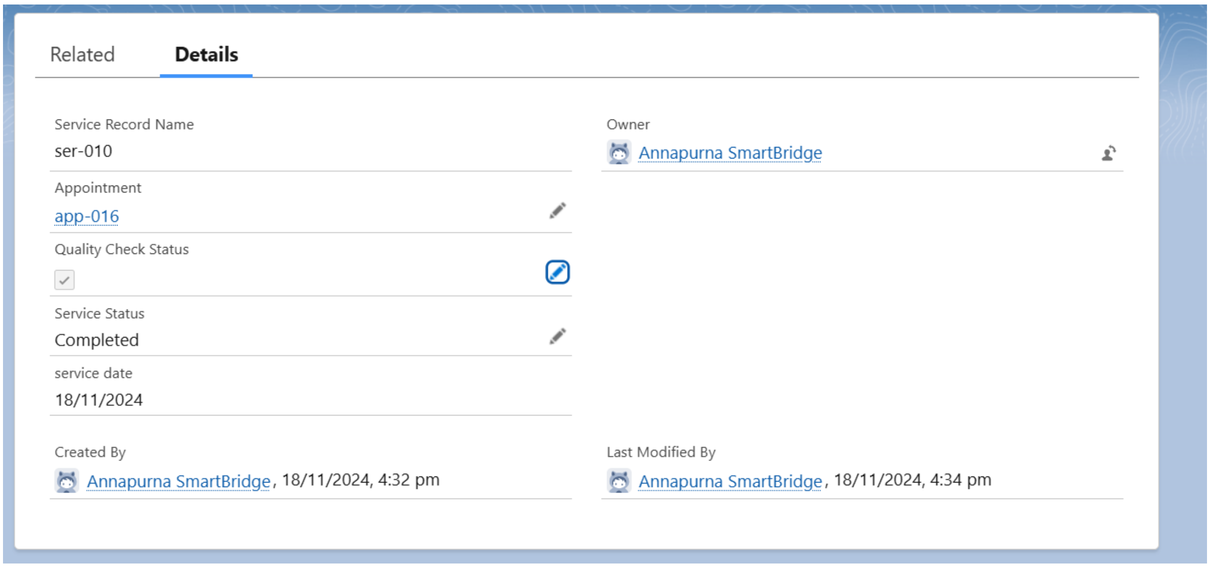


4. Open the record and click on Quality check status as true.

5. Click on save.



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



Conclusion

The "Garage Management System" project successfully demonstrates the power of the Salesforce platform in creating a specialized, efficient, and user-friendly application for a real-world business need. By automating key tasks, ensuring data integrity, and providing clear analytics, the system empowers automotive repair facilities to improve their operations and enhance customer satisfaction.

- **Project Achievements:**
  - Successfully built a streamlined process for managing customers, appointments, services, and billing.
  - Automated complex calculations and status updates using a combination of Flows and Apex triggers.
  - Implemented a robust security model to ensure users can only access appropriate data.
  - Provided clear, actionable insights through customized reports and dashboards.
- **Future Scope:**
  - **Mobile Integration:** Develop mobile-friendly components to allow technicians to update service records directly from the workshop floor.
  - **Inventory Management:** Add custom objects to track spare parts inventory, automating reorder processes when stock is low.
  - **Customer Portal:** Create a Salesforce Experience Cloud site to allow customers to book their own appointments, view their service history, and pay bills online.
  - **AI Integration:** Use Salesforce Einstein to predict service needs based on vehicle history and mileage, allowing for proactive customer outreach.