GARAGE MANAGEMENT SYSTEM

College Name: BISHOP APPASAMYCOLLEGE OF

ARTS AND SCIENCE

College Code:BRU3G

TEAM ID: NM2025TMID20453

TEAM MEMBERS:

Team Leader Name: Jeevanandham.v

Email:sanjayjeeva8870@gmail.com

Team Member 1: B.Laba rawal

Email:labarawaljr10@gmail.com

Team Member 2:R.Hafeez

Email:hafehafe23@gmail.com

Team Member 3:Ebinesarmano.B

Email:ebinasarmano2411@gmail.com

1.INTRODUCTION

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

DEVELOPMENT PHASE

Creating Developer Account: By using this URL

- Developer Edition with Agentforce and Data Cloud - Salesforce.com

Salesforce

Introduction:

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don't know where you should start on your learning journey? If you've answered yes to any of these questions, then you're in the right place. This module is for you.

Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we'll take you through these features and answer the question, "What is Salesforce, anyway?".

What Is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

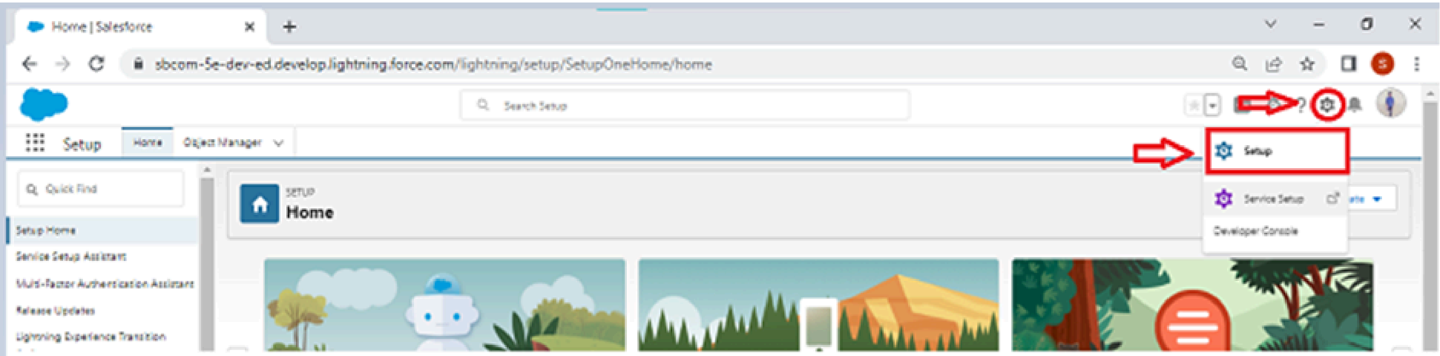
Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud.

So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something like this:

Object

1. **What Is an Object?**
2. Salesforce objects are database tables that permit you to store data that is specific to an organization. What are the types of Salesforce objects
3. Salesforce objects are of two types:
    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.
4.

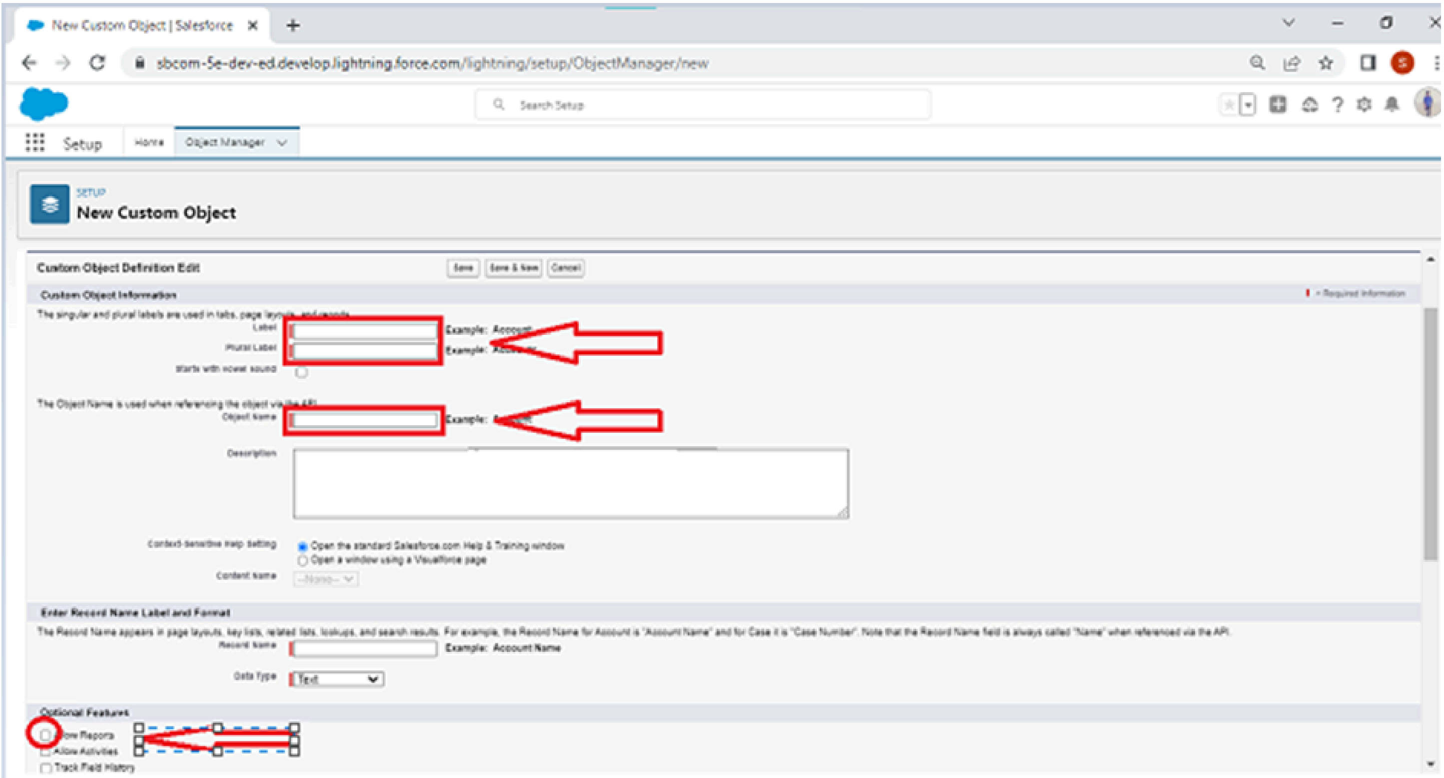
    To Navigate to Setup page:
5. Click on gear icon ? click setup.



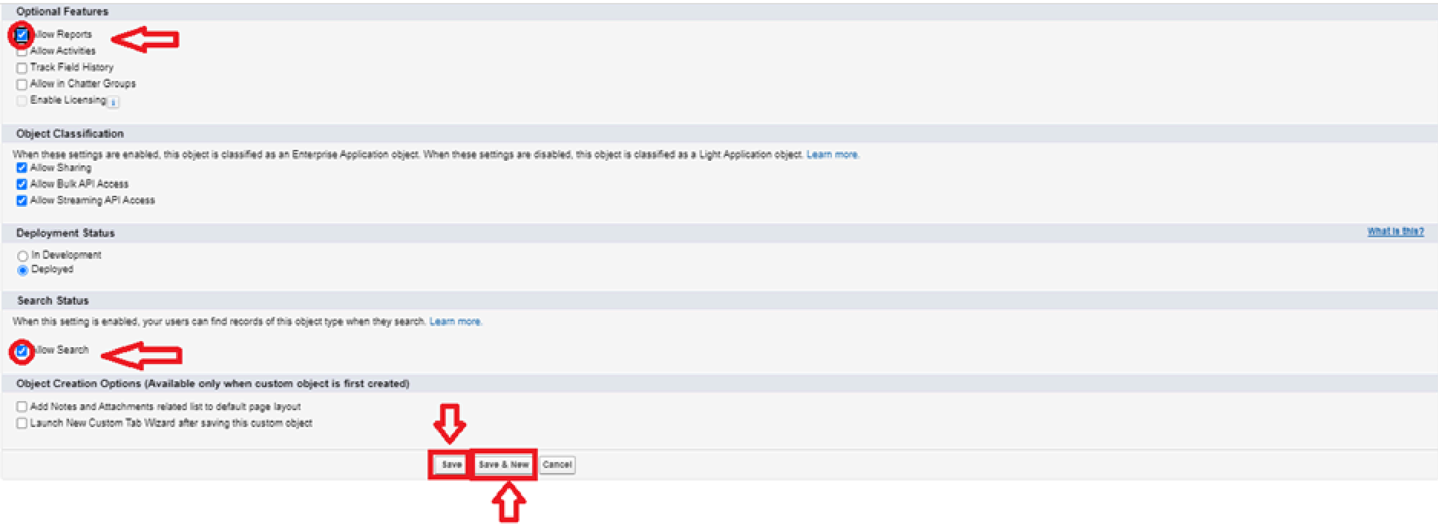
1.
2.

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

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


New Custom Object | Salesforce
sbcom-5e-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/new
Search Setup
Setup
Home
Object Manager
SETUP
New Custom Object
Custom Object Definition Edit
Save
Save & New
Cancel
Custom Object Information
Example: Account
Description
Open the standard Salesforce.com Help & Training window
Open a window using a Visualforce page
Enter Record Name Label and Format
Example: Account Name
Text
Optional Features
Allow Activities
Track Field History

1.
2.


Optional Features
Allow Reports
Allow Activities
Track Field History
Allow in Chatter Groups
Enable Licensing
Object Classification
When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. Learn more.
Allow Sharing
Allow Bulk API Access
Allow Streaming API Access
Deployment Status
What is this?
In Development
Deployed
Search Status
When this setting is enabled, your users can find records of this object type when they search. Learn more.
Allow Search
Object Creation Options (Available only when custom object is first created)
Add Notes and Attachments related list to default page layout
Launch New Custom Tab Wizard after saving this custom object
Save
Save & New
Cancel

1.

Tabs

What is Tab : A tab is like a user interface that is used to build records for objects and to view the records in the objects.

Types of Tabs:

1. **Custom Tabs**

   Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

1. **Web Tabs**

   Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application.

1. **Visualforce Tabs**

   Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

1. **Lightning Component Tabs**

   Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

1. **Lightning Page Tabs**

   Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

   Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show up in the Available Tabs list when you customise the tabs for your apps.

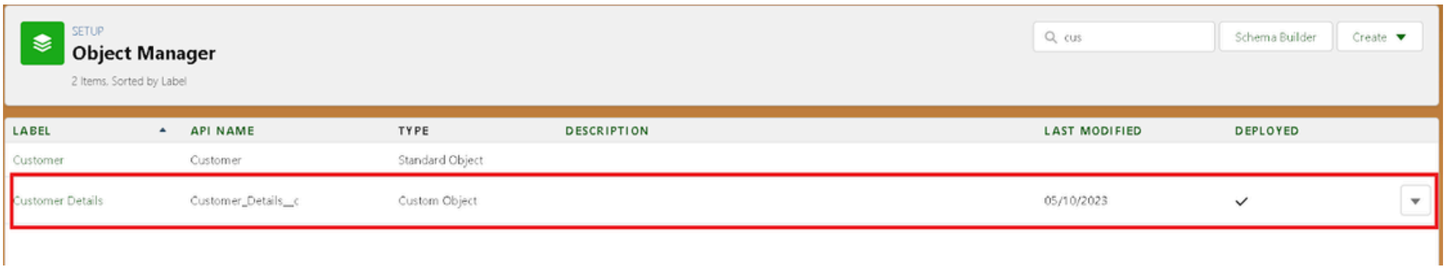
An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

Lightning apps let you brand your apps with a custom colour and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps

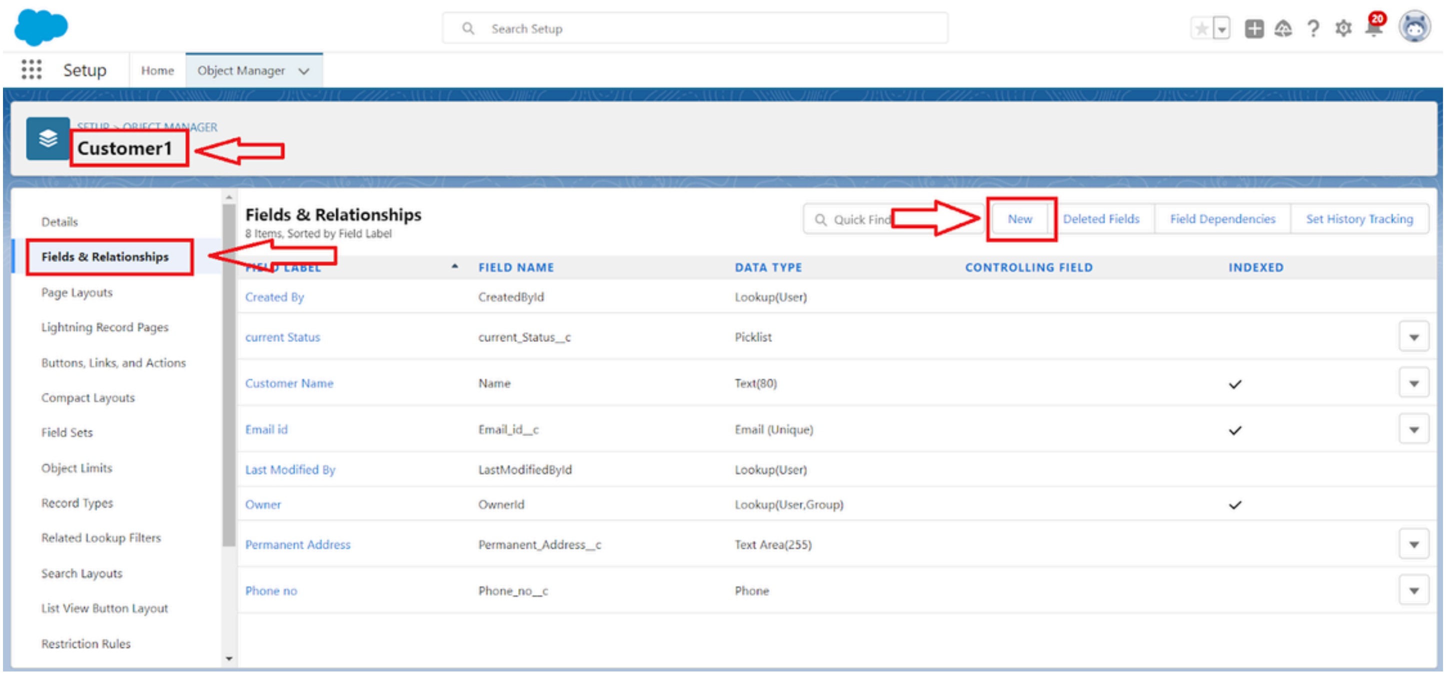
Creation of fields for the Customer Details object

1. To create fields in an object:

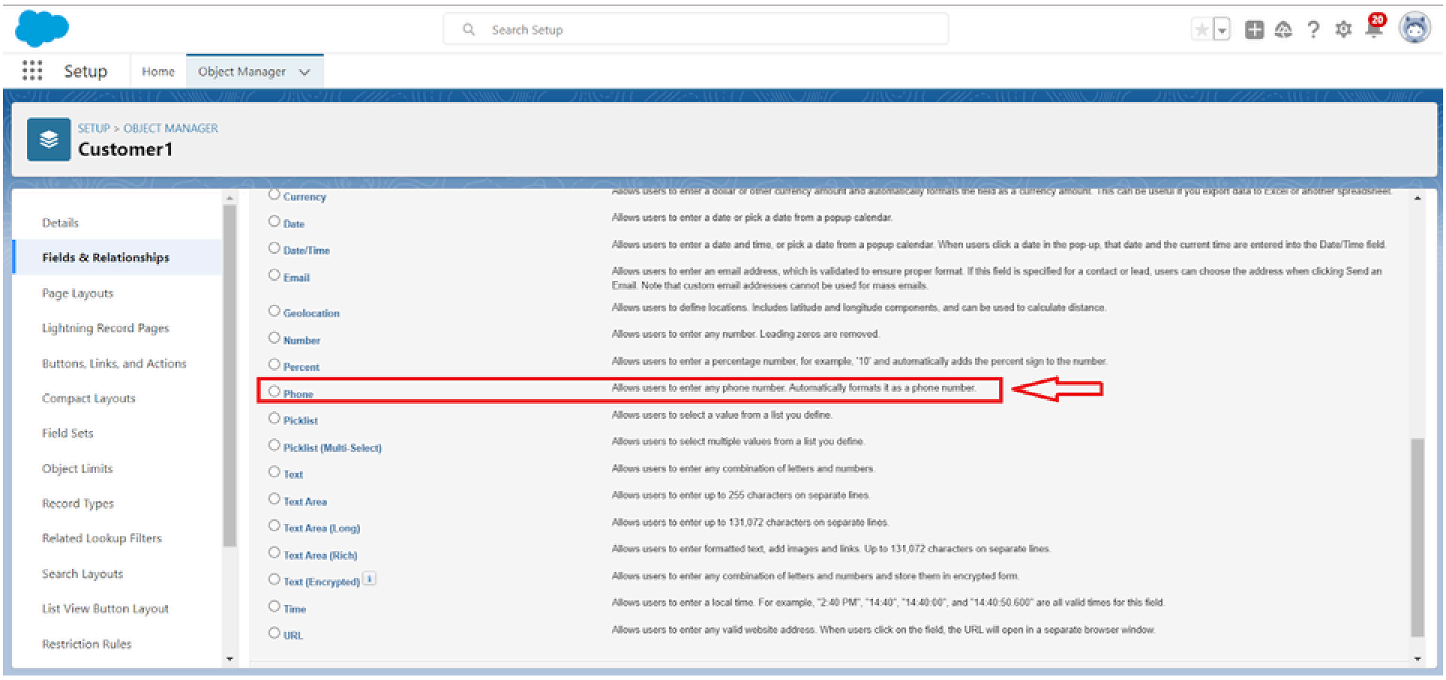
1. Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.



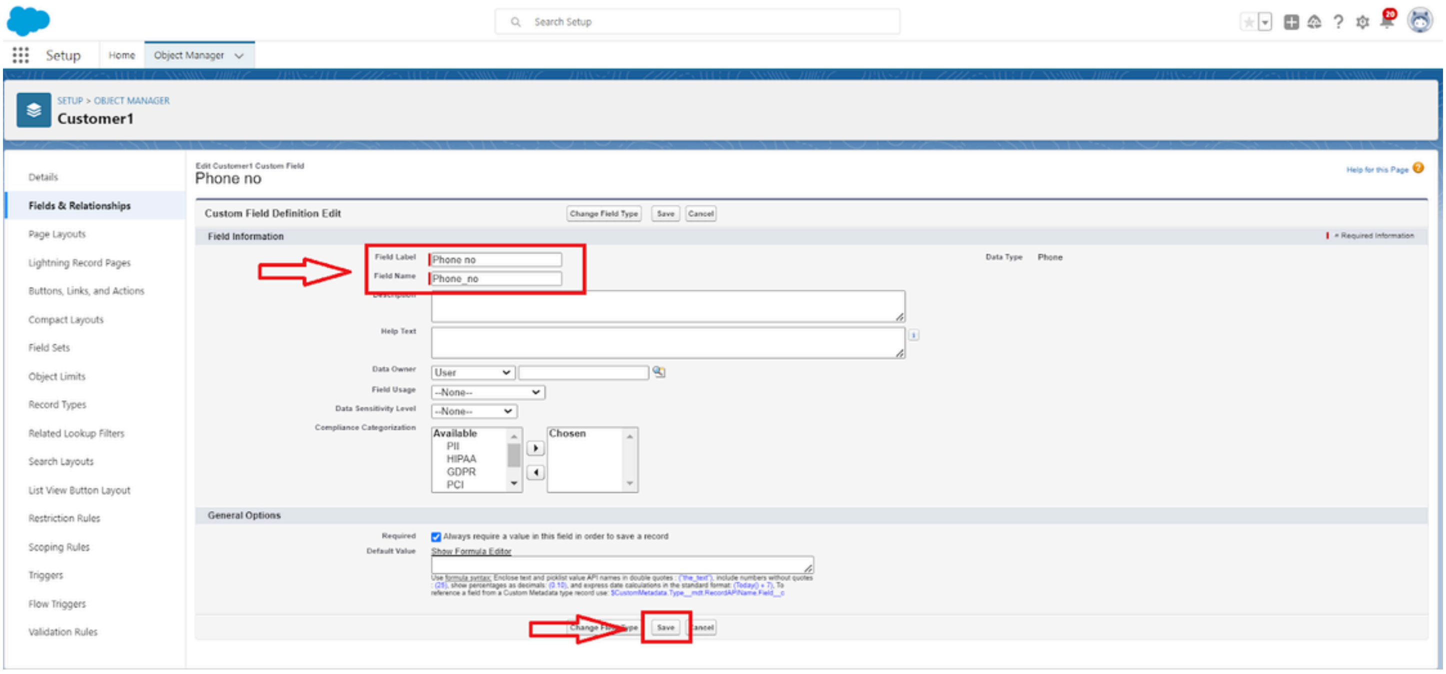
1. Now click on “Fields & Relationships” >> New



1.
Select Data Type as a “Phone”



1. 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:

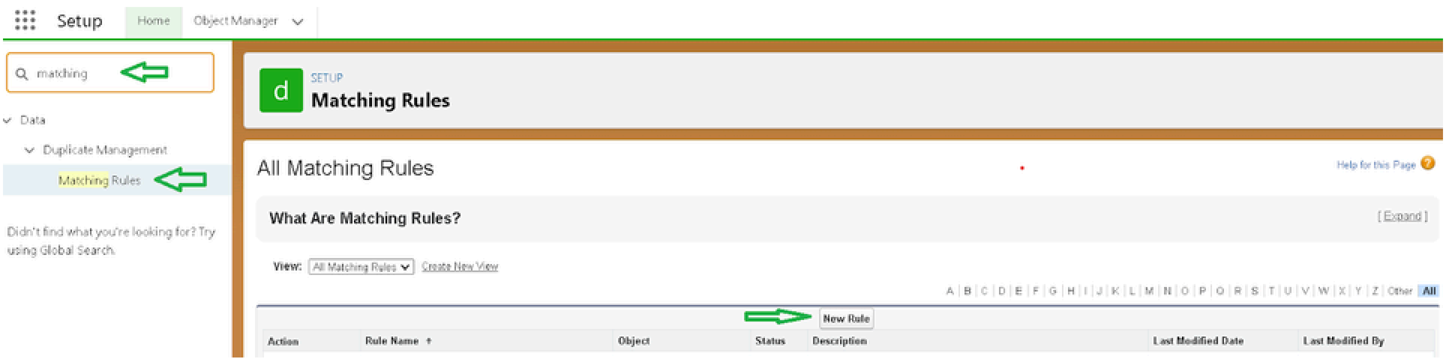
- Field Label : Gmail
- Field Name : gets auto generated
- Click on Next >> Next >> Save and new.

Validation rule

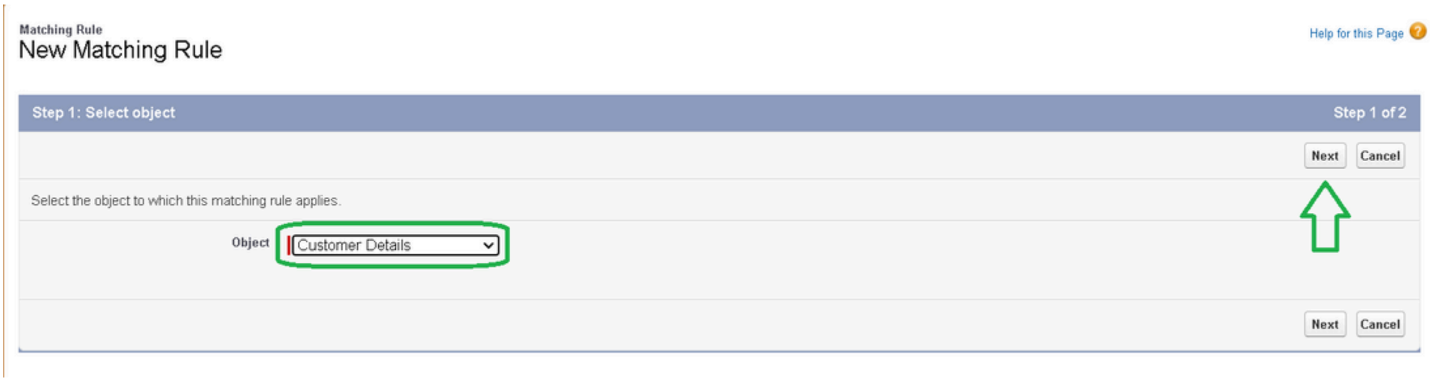
Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.

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.



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

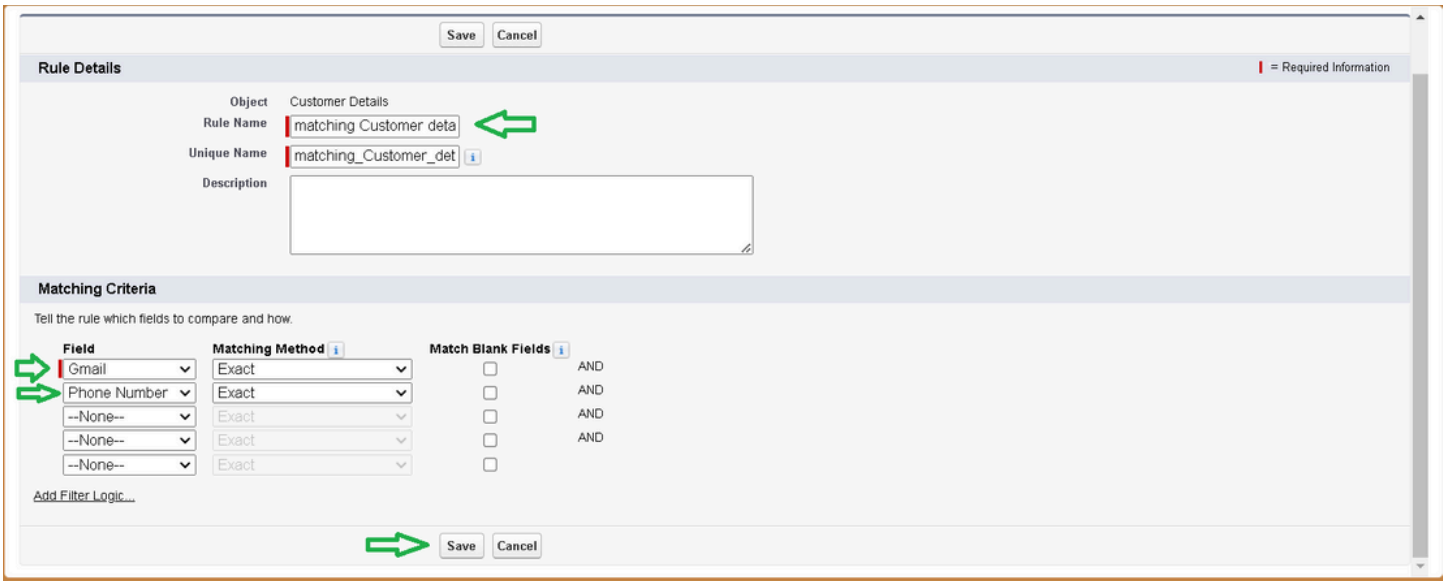


1. Give the Rule name : Matching customer details
2. Unique name : is auto populated
3. Define the matching criteria as
4. Field Matching Method

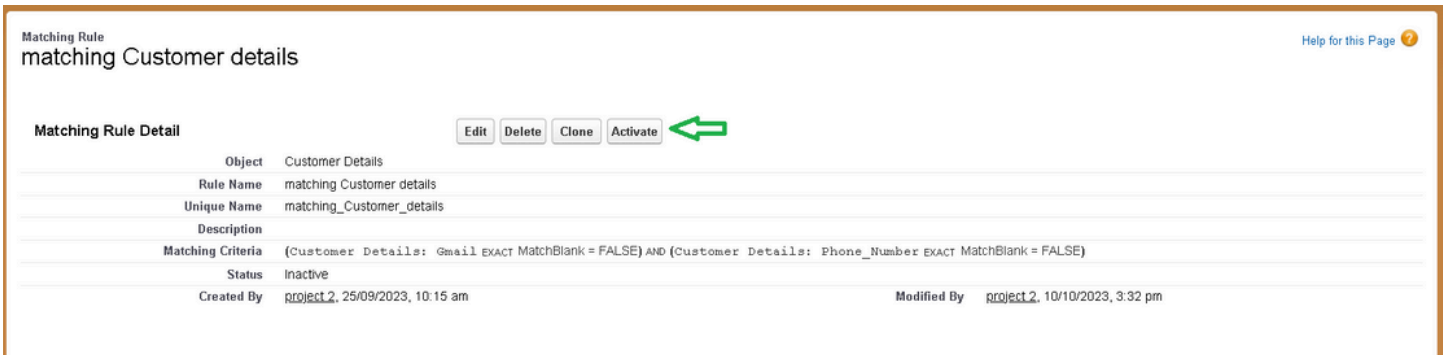
   1. Gmail Exact

   2. Phone Number Exact

1. Click save.
2. After Saving Click on Activate.



Save
Cancel
Rule Details
= Required Information
Object
Customer Details
Rule Name
matching Customer deta
Unique Name
matching_Customer_det
Description
Matching Criteria
Tell the rule which fields to compare and how.
Field
Matching Method
Match Blank Fields
Gmail
Exact
AND
Phone Number
Exact
AND
--None--
Exact
AND
--None--
Exact
AND
--None--
Exact
Add Filter Logic...
Save
Cancel



Matching Rule
matching Customer details
Help for this Page
Matching Rule Detail
Edit
Delete
Clone
Activate
Object
Customer Details
Rule Name
matching Customer details
Unique Name
matching_Customer_details
Description
Matching Criteria
(Customer Details: Gmail EXACT MatchBlank = FALSE) AND (Customer Details: Phone_Number EXACT MatchBlank = FALSE)
Status
Inactive
Created By
project 2, 25/09/2023, 10:15 am
Modified By
project 2, 10/10/2023, 3:32 pm

Apex Trigger

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions

before or after changes to Salesforce records, such as insertions, updates, or deletions.

A trigger is Apex code that executes before or after the following types of operations:

- insert
- update
- delete
- merge
- upsert
- undelete

For example, you can have a trigger run before an object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle Bin.

You can define triggers for top-level standard objects that support triggers, such as a Contact or an Account, some standard child objects, such as a CaseComment, and custom objects. To define a trigger, from the object management settings for the object whose triggers you want to access, go to Triggers.

There are primarily two types of Apex Triggers:

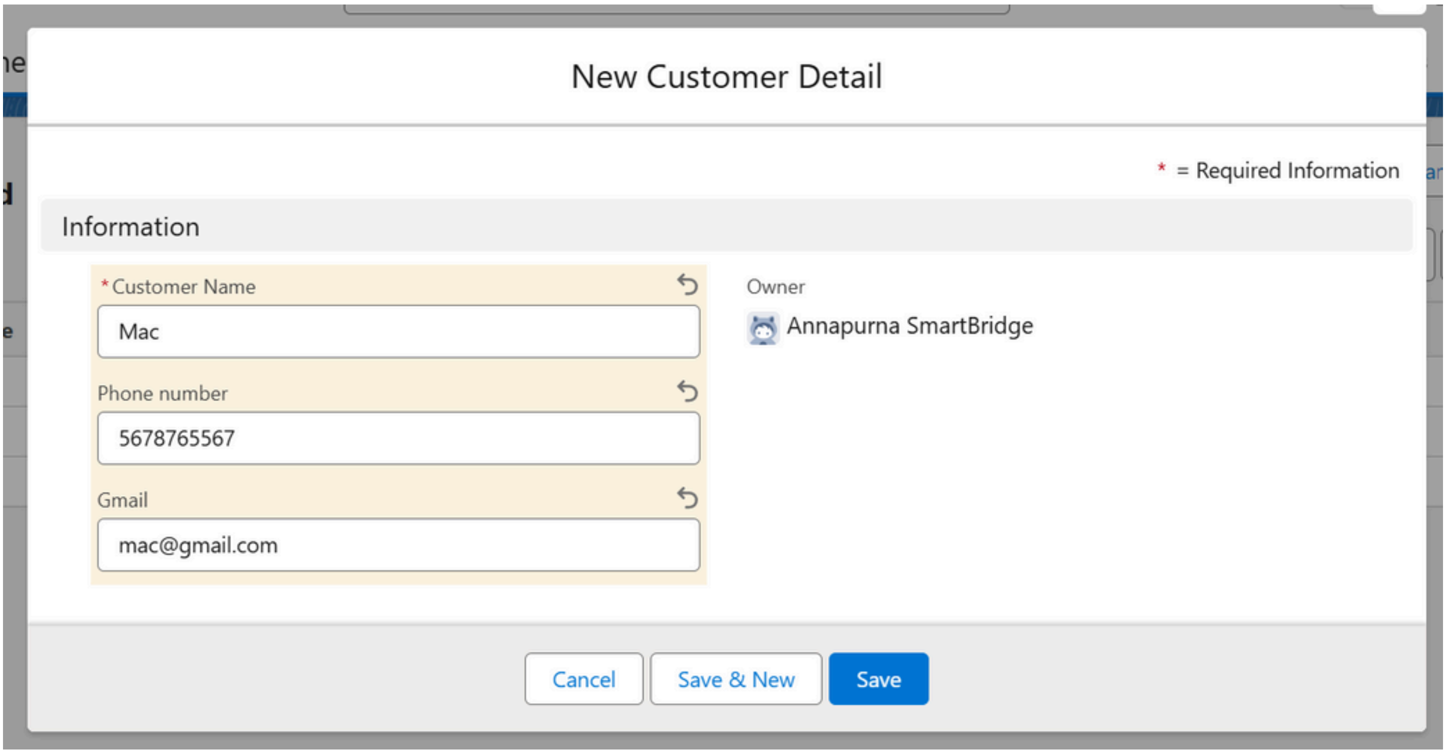
Before Trigger: This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets ready to be inserted into the database.

After Trigger: This type of trigger in Salesforce is used to access the field values set by the system and affect any change in the record. In other words, the after trigger makes changes to the value from the data inserted in some other record.

creating records

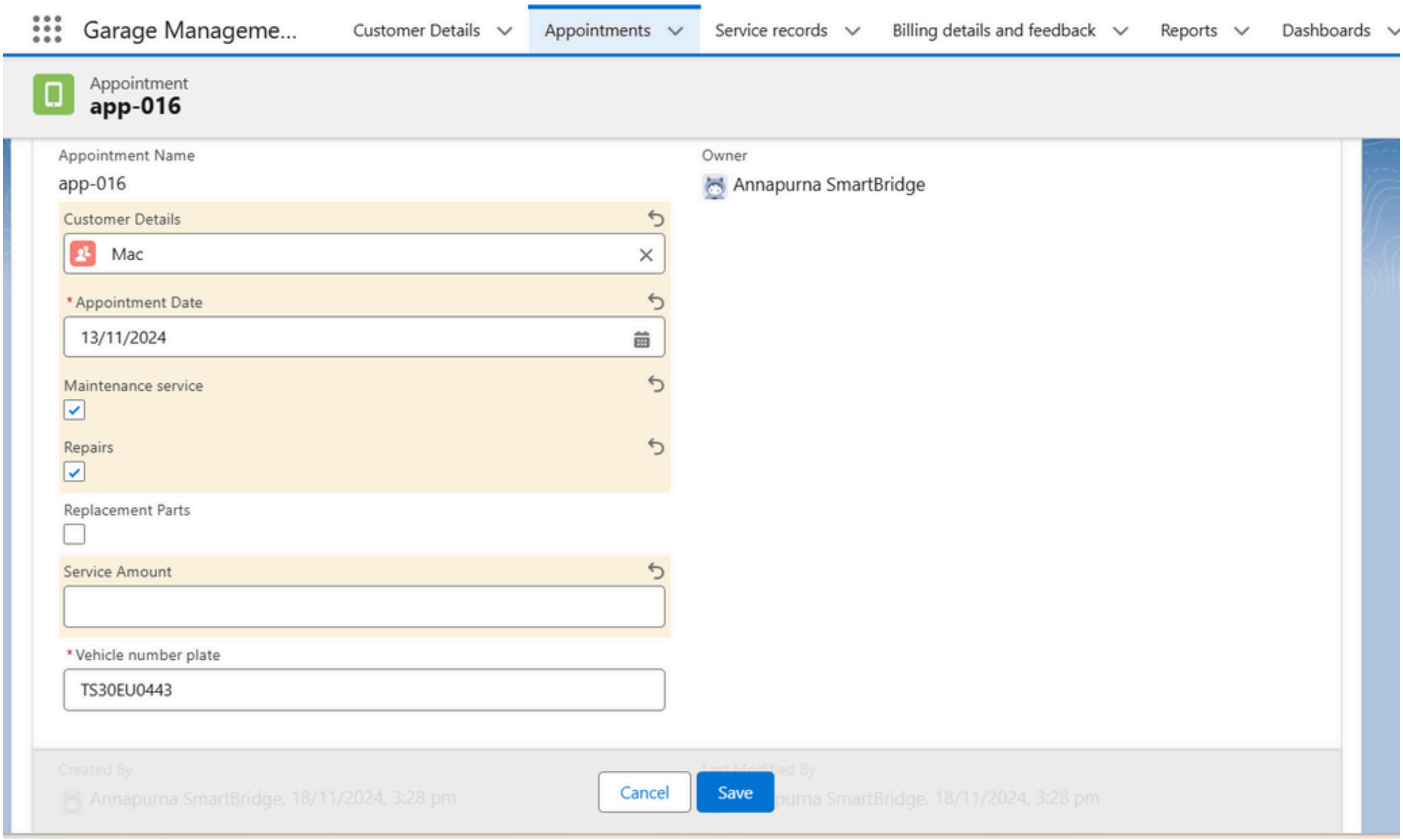
To create a record in the follow objects follow these steps

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.



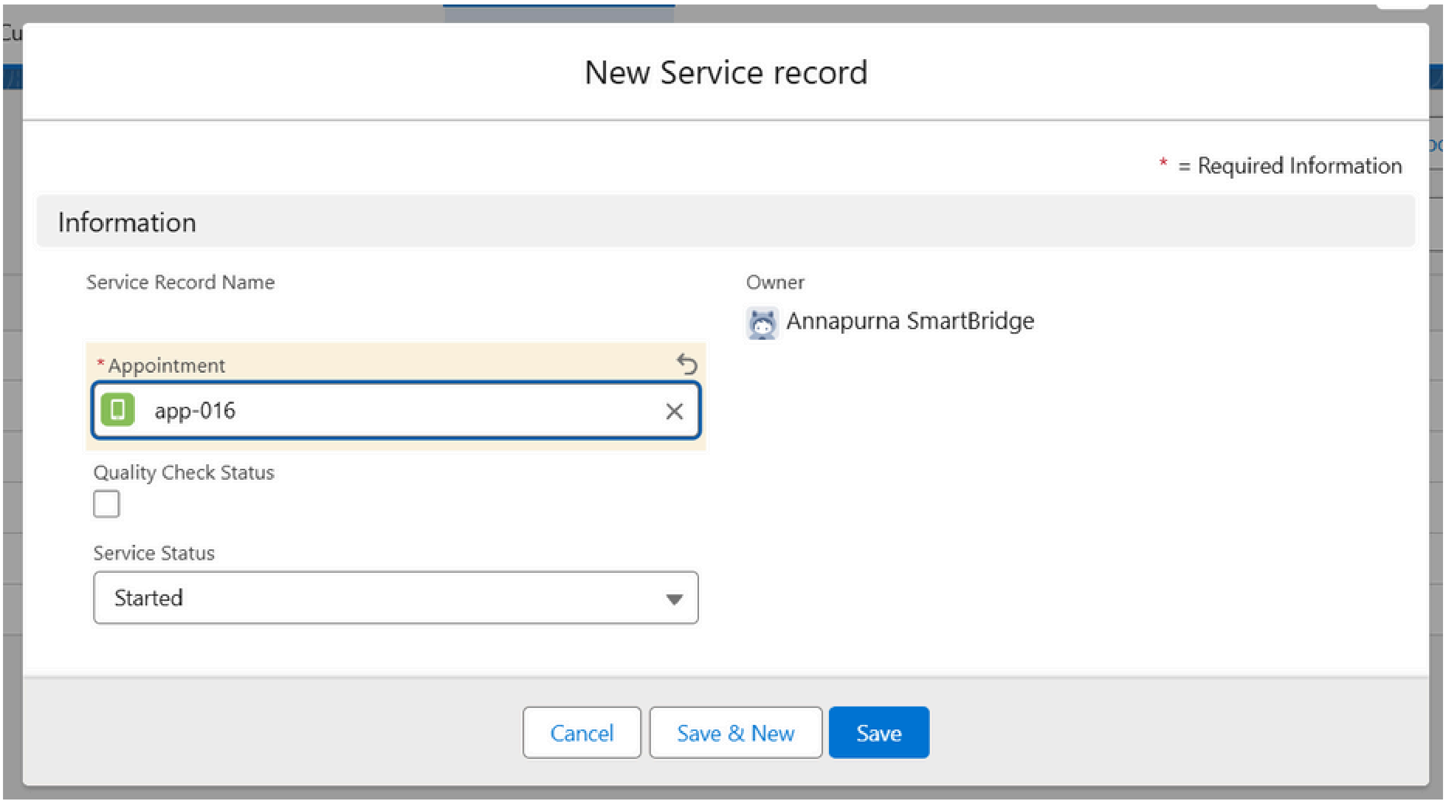
Now, Create the Appointment Record

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.

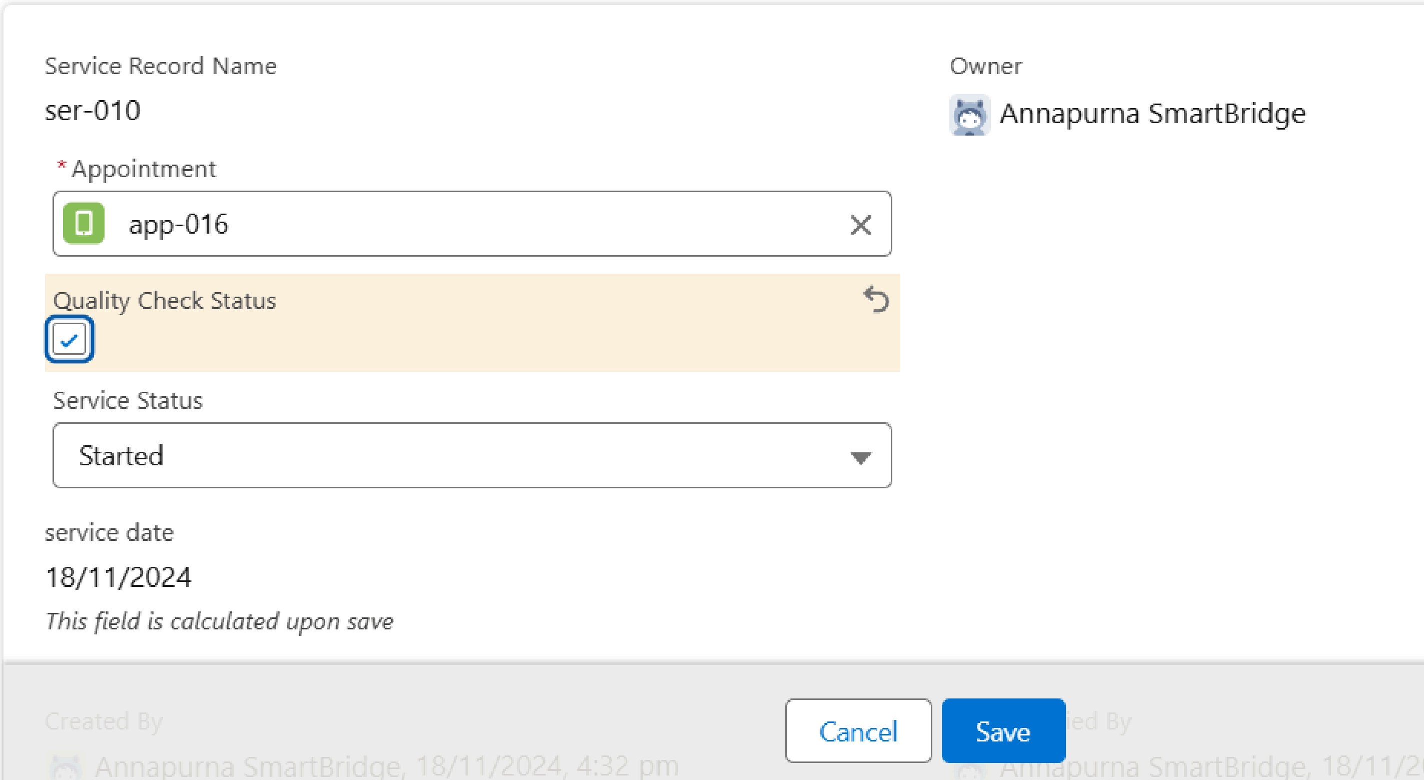


Now, Create a service Record

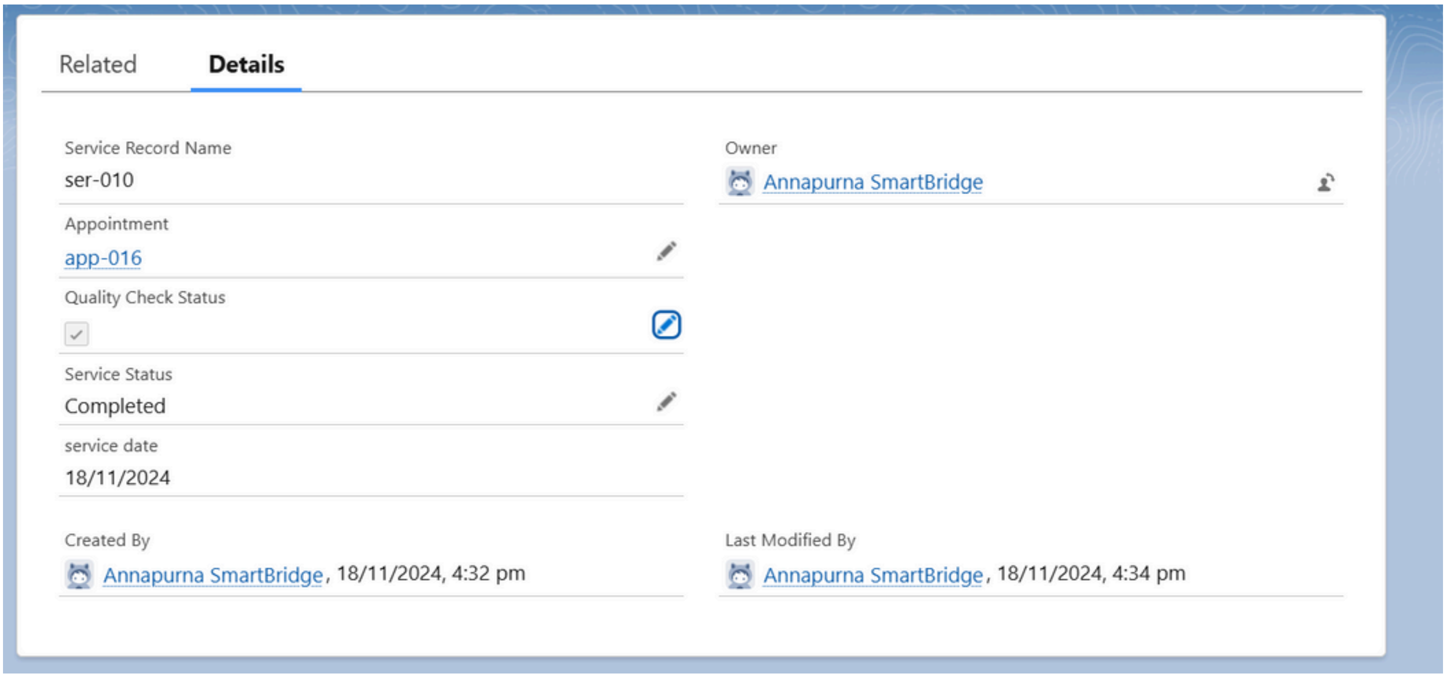
1. Click on the "**Service record** tab".
2. Enter the Appointment, and started is selected as default.
3. Click on save.



1. Open the record and click on Quality check status as true.
2. Click on save.



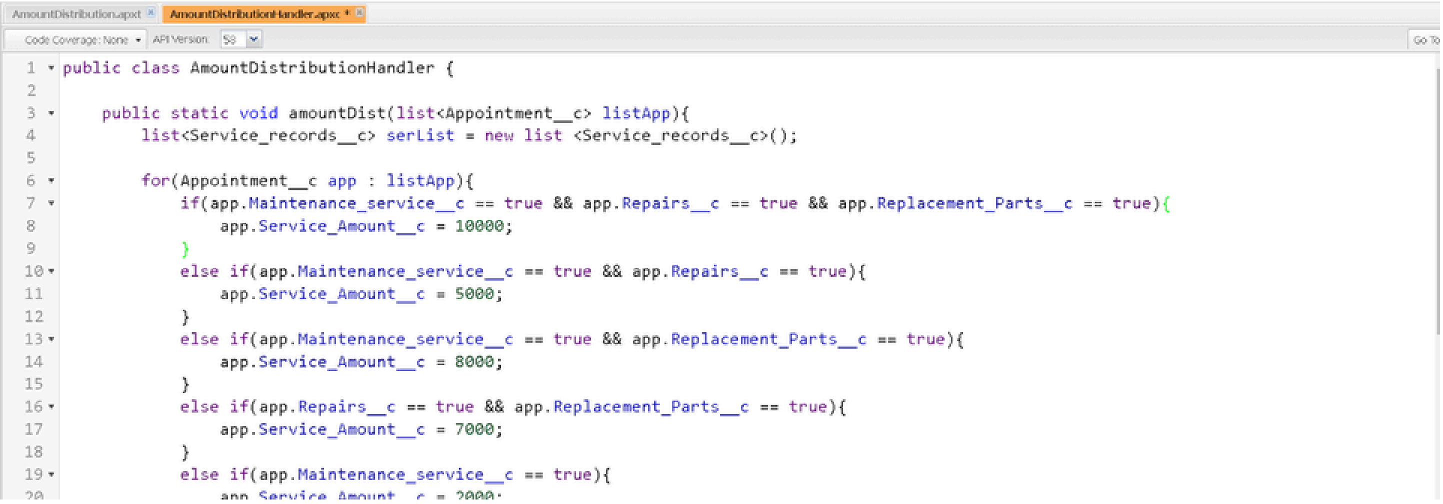
1. Now automatically Service status will be moved to completed.



Apex handler

UseCase : This use case works for Amount Distribution for each Service the customer selected for there Vehicle.

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 ”.



Code:

public class AmountDistributionHandler {

public static void amountDist(list<Appointment__c> listApp){

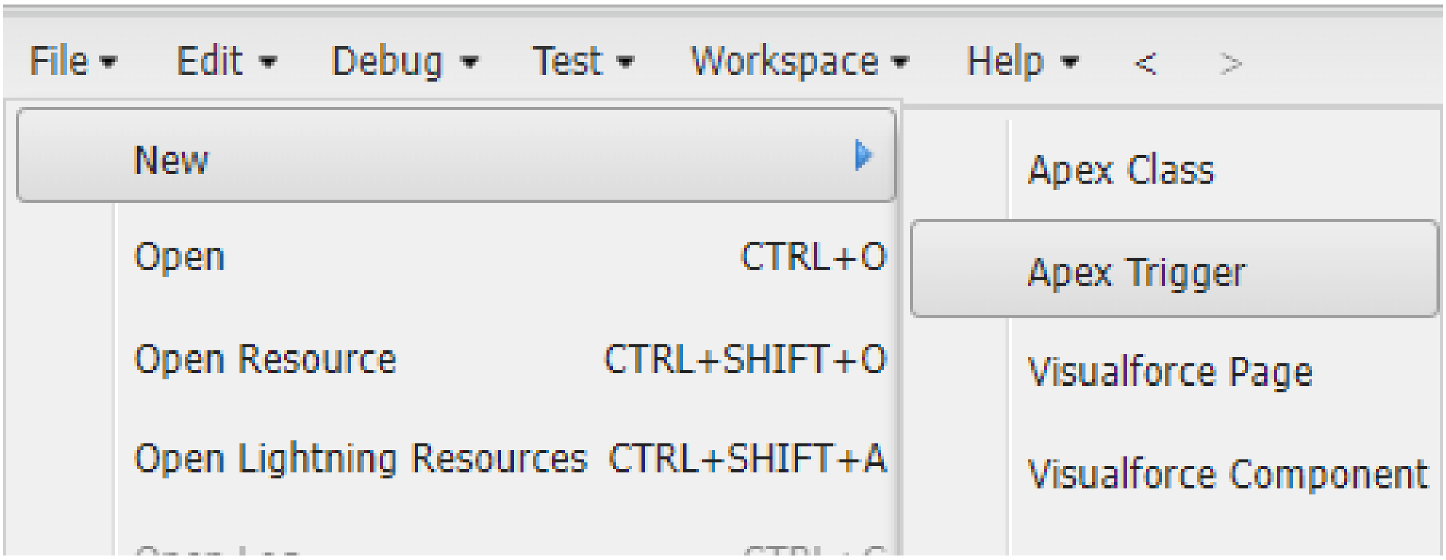
list<Service_records__c> serList = new list <Service_records__c>();

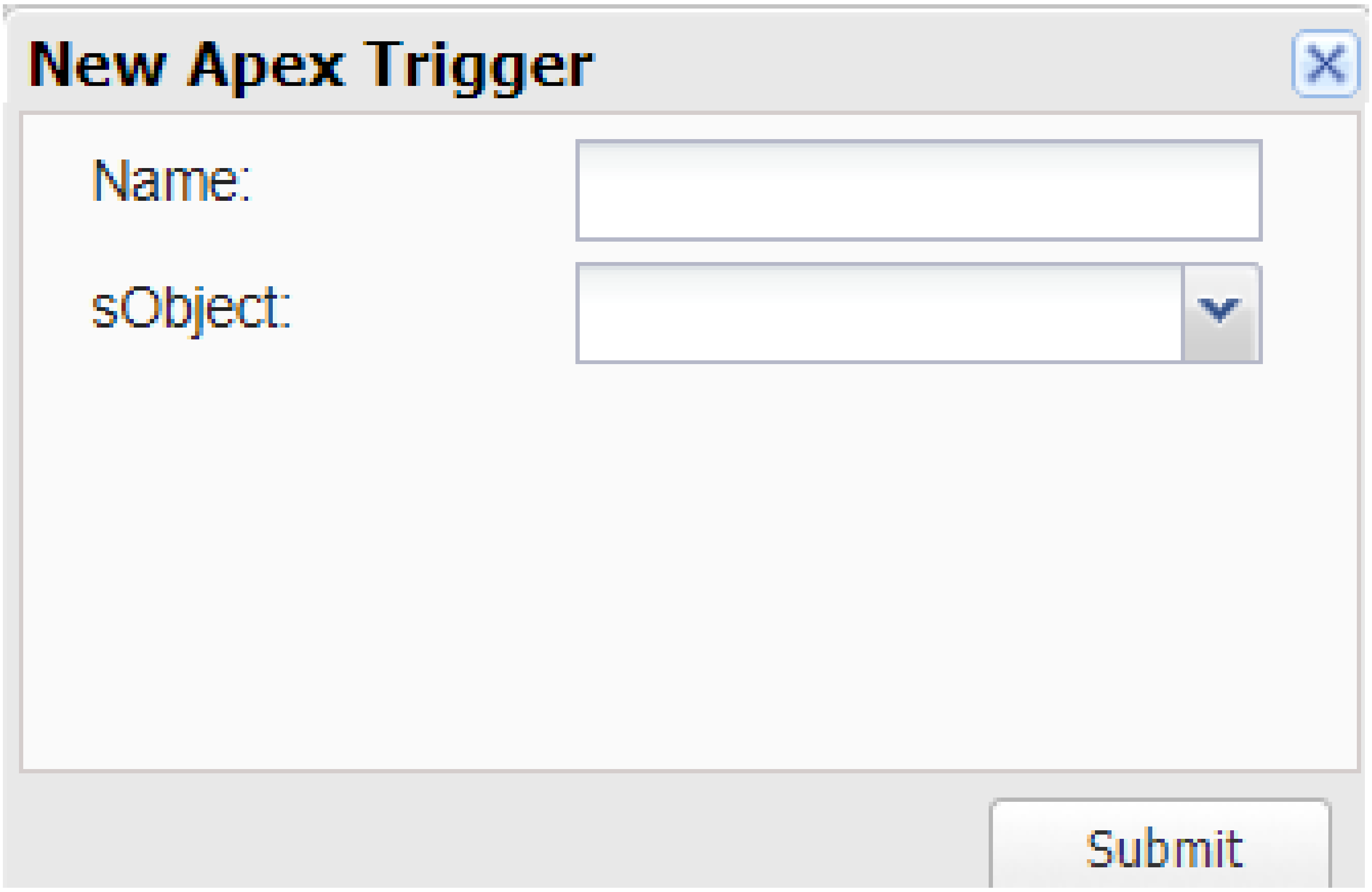
```
for(Appointment__c app : listApp){

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 :

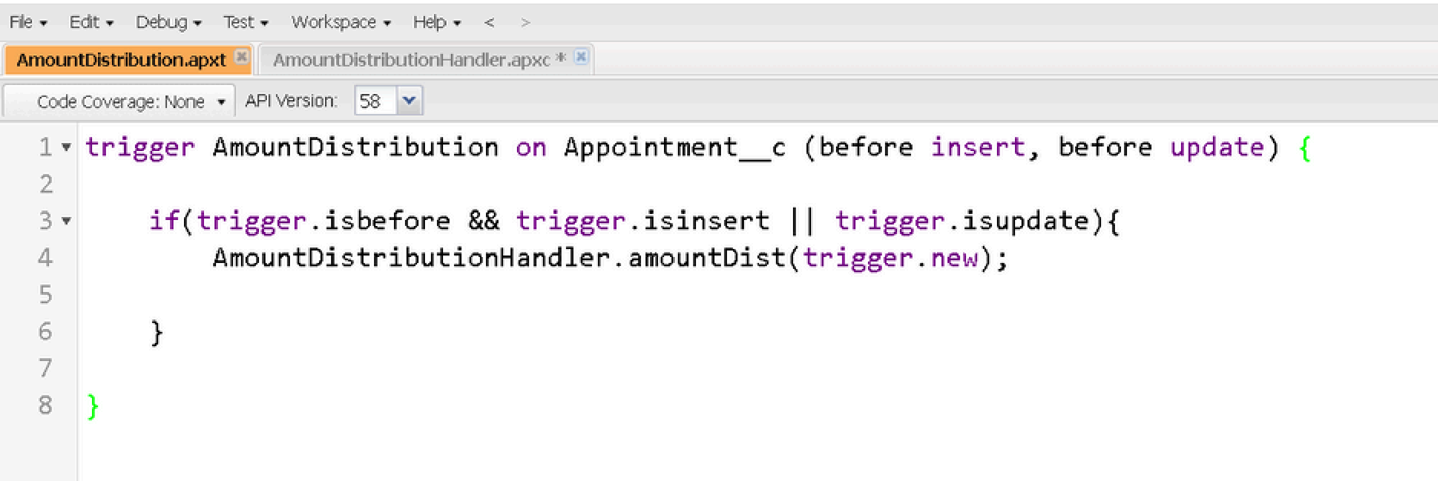
Trigger [trigger name] on [object name](Before/After event)

{

}

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




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

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

    }

}
```

Code:

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

if(trigger.isbefore && trigger.isinsert || trigger.isupdate){

AmountDistributionHandler.amountDist(trigger.new);

```
 }

}
```