Garage Management System in Salesforce
Project Documentation

Prepared for: Salesforce Developer Implementation

Project: Garage Management Application

Team members:

Lavanya Pinnamaraju

Chapala Sneha

Devarakonda Yeswanthi Sai Bala Sushma

Akula Vindhya

Table of Contents

Introduction

The Garage Management System (GMS) is a cloud-native solution developed on Salesforce...

Certainly! Here's a detailed **Introduction** you can include at the beginning of your **Salesforce Garage Management System** project documentation:

The **Garage Management System (GMS)** is a comprehensive cloud-based application built on the Salesforce platform, designed to streamline and automate the daily operations of a garage or vehicle service center. As automotive service businesses face increasing demand for efficient customer handling, appointment scheduling, billing, and service tracking, GMS offers a centralized digital solution to address these needs.

This project leverages the powerful features of **Salesforce CRM**, including its customizable objects, automation tools, role-based access control, and reporting capabilities, to manage garage operations effectively. It is developed using **Salesforce Developer Edition**, utilizing tools such as **Object Manager, Flows, Validation Rules, Apex, Lightning App Builder, Reports, and Dashboards**.

The system is designed with four major modules:

- **Customer Management** – storing customer details like name, phone number, and email.
- **Appointment Scheduling** – managing service appointments, vehicle information, and types of services required.
- **Service Record Management** – tracking the services performed, their statuses, and quality checks.
- **Billing and Feedback** – recording payments, customer feedback, and payment status.

Each of these modules is built using **custom Salesforce objects** with necessary fields, relationships, and user interfaces. Additionally, **automation is handled via Flows and Apex Triggers**, ensuring tasks such as billing calculations, notifications, and status updates are performed without manual effort.

To maintain data integrity and security, **validation rules, sharing settings, profiles, and roles** are implemented. Managers and salespersons have different levels of access to data, and sensitive service information is protected through **OWD (Organization-Wide Defaults)** and **role-based sharing rules**.

Objective

The main objective of this project is to create a streamlined management system for garage services using Salesforce...

The primary objective of the **Garage Management System (GMS)** project is to **digitize and automate** the end-to-end operations of a vehicle garage or automotive service center using the Salesforce platform. The system is designed to enhance service quality, streamline processes, and improve customer experience through a centralized, cloud-based solution.

- **Customer Management:**
  - Store and manage essential customer information (name, contact number, email).
  - Enable quick retrieval of customer details during appointments and billing.
- **Appointment Scheduling:**
  - Allow users to create and track service appointments.
  - Capture service types such as maintenance, repairs, and replacement parts.
  - Automatically calculate service charges based on selected services.
- **Service Tracking:**
  - Maintain detailed records of each service performed.
  - Implement a quality check system and update service status automatically.

  - Link appointments to services for full traceability.

- **Billing and Feedback Collection:**

  - Record payments made by customers and track their payment status.
  - Collect service feedback and ratings.
  - Automate thank-you emails upon successful payment.

- **Security and Access Control:**

  - Assign roles (Manager, Salesperson) with appropriate permissions.
  - Control access to records using sharing settings and public groups.

- **Automation and Workflow Optimization:**

  - Use Salesforce Flows and Apex triggers to automate:
    - Payment updates,
    - Email alerts,
    - Service status changes.

- **Reporting and Analysis:**

  - Generate real-time reports and dashboards to track:
    - Appointment data,
    - Payments,
    - Customer satisfaction.
  - Share reports with specific user roles for informed decision-making.

- **Scalability and Maintenance:**

  - Create a system that is modular, customizable, and easy to expand for future needs

Tools and Technologies Used

- Salesforce Developer Edition
- Apex Programming
- Lightning App Builder
- Flows
- Reports and Dashboards

System Architecture

The system follows a modular architecture based on custom Salesforce objects with lookup relationships.

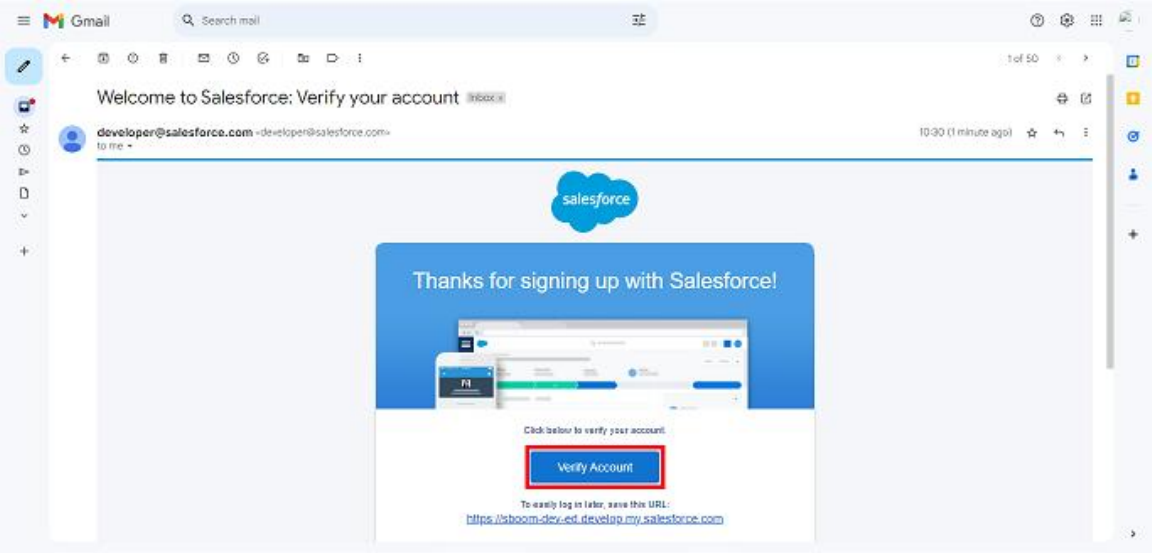
Project Setup

Step-by-step instructions to create a developer org, activate it, and log into the Salesforce environment.

Setting up a garage management project typically involves several key steps:

1. **Define Requirements**: Outline the functionalities required, such as vehicle registration, service history tracking, customer management, and inventory control.
2. **Choose Technology Stack**: Select appropriate technologies like a backend framework (e.g., Django, Node.js), frontend framework (e.g., React, Angular), and database (e.g., PostgreSQL, MySQL).
3. **Set Up Development Environment**:
   - Install necessary software (IDE, database management tools).
   - Initialize a version control system (e.g., Git) for collaborative development.
4. **Backend Development**:
   - Create models for vehicles, customers, services, etc.
   - Implement APIs for CRUD operations (Create, Read, Update, Delete).
   - Set up authentication and authorization mechanisms.
5. **Frontend Development**:
   - Design UI/UX wireframes and layouts.
   - Develop frontend components to interact with backend APIs.
   - Implement user interfaces for managing vehicles, customers, services, etc.
6. **Database Setup**:
   - Design and create database schemas based on the application's requirements.

   - Set up database migrations for version control and schema updates.
7. **Integration and Testing**:
   - Integrate frontend with backend APIs.
   - Perform unit testing for each module and integration testing for the entire system.
8. **Deployment and Maintenance**:
   - Deploy the application on a suitable server (e.g., AWS, Heroku).
   - Monitor performance and handle maintenance tasks like updates and backups.
9. **Documentation and User Training**:
   - Prepare user manuals and technical documentation.
   - Provide training sessions for garage staff on using the management system effectively.
10. **Support and Iteration**:
    - Gather feedback from users and stakeholders.
    - Iterate on the application based on feedback and evolving requirements.



Welcome to Salesforce: Verify your account

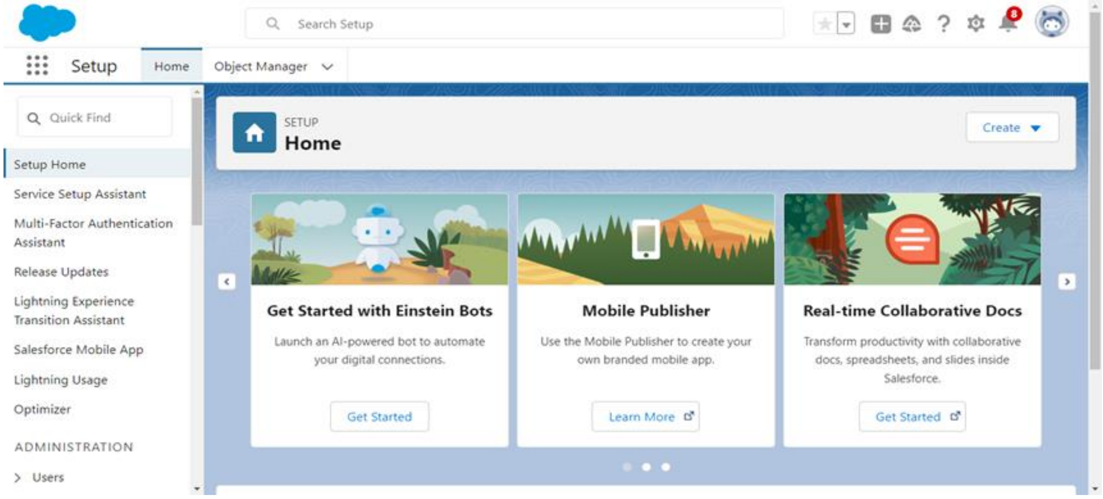
developer@salesforce.com <developer@salesforce.com>

10:30 (1 minute ago)

Thanks for signing up with Salesforce!

Click below to verify your account.

To easily log in later, save this URL:
https://sboom-dev-ed.develop.my.salesforce.com



Get Started with Einstein Bots

Launch an AI-powered bot to automate your digital connections.

Mobile Publisher

Use the Mobile Publisher to create your own branded mobile app.

Real-time Collaborative Docs

Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce.

Object Creation

Detailed creation steps for Customer Details, Appointment, Service Records, and Billing Details objects.

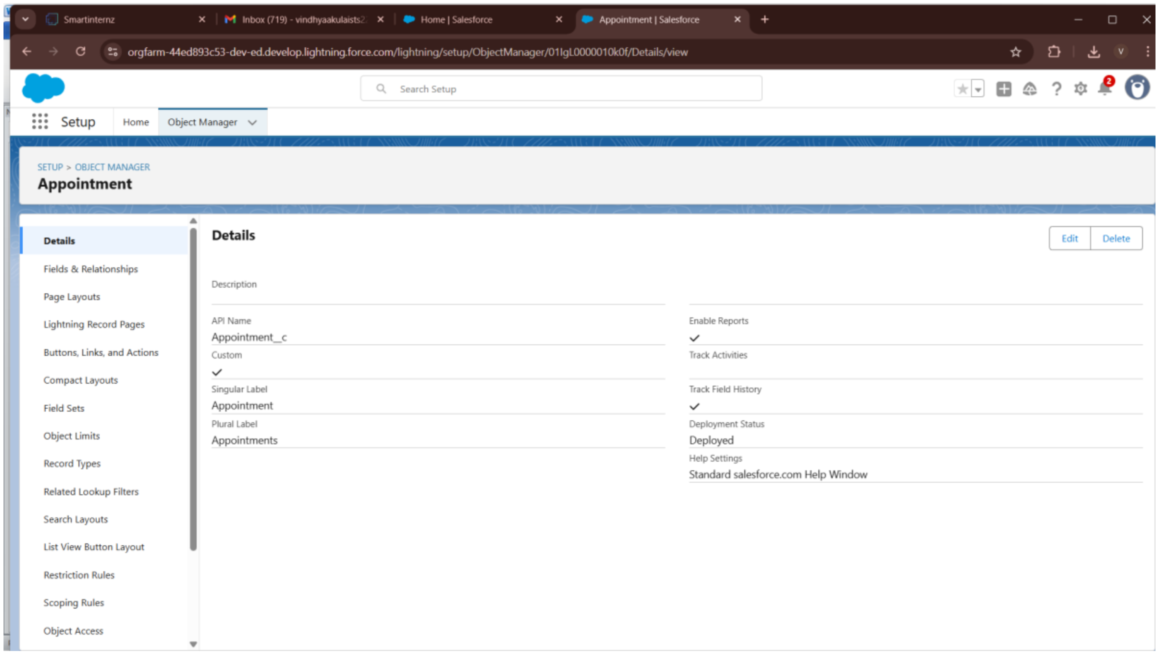
Object Name	Record Name Type	Key Purpose
Customer Details	Text	Store customer personal and contact details
Appointment	Auto Number	Service booking for vehicles
Service Records	Auto Number	Record type of service performed
Billing Details and Feedback	Auto Number	Bill payment tracking and feedback

Salesforce objects are database tables that permit you to store data that is specific to an organization. What are the types of Salesforce objects

Salesforce objects are of two types:

Standard Objects: Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.

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.


Smartinternz
Inbox (719) - vindhyaakulaists2
Home | Salesforce
Appointment | Salesforce
orgfarm-44ed893c53-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgL0000010k0f/Details/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER

Appointment

Details

Description

API Name
Appointment__c

Custom
✓

Singular Label
Appointment

Plural Label
Appointments

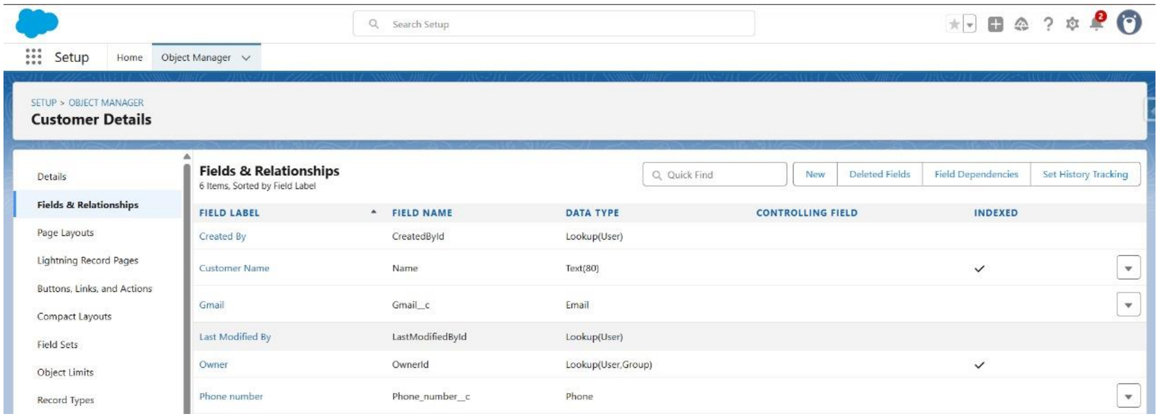
Enable Reports
✓

Track Activities

Track Field History
✓

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

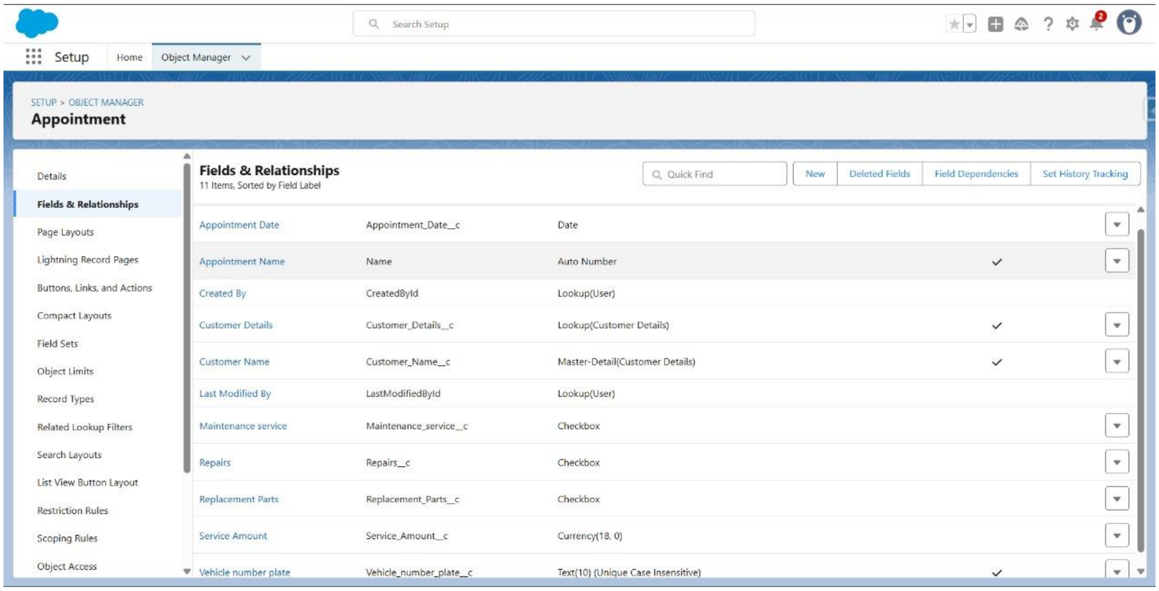


Customer Details

Fields & Relationships

6 Items, Sorted by Field Label

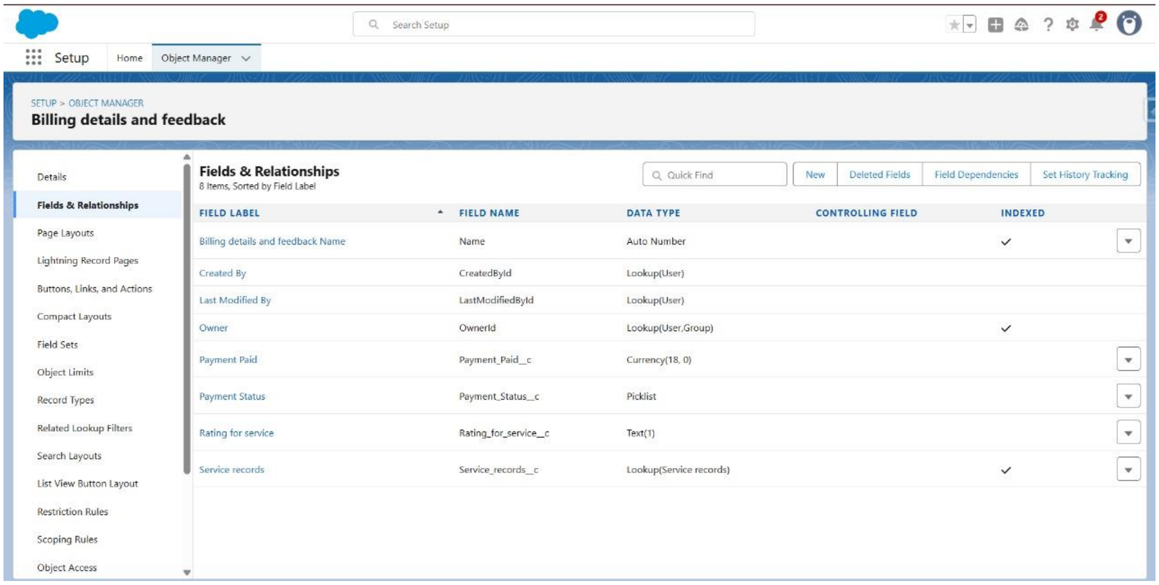
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer Name	Name	Text(80)		✓
Gmail	Gmail__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone number	Phone_number__c	Phone		



Appointment

Fields & Relationships
11 Items, Sorted by Field Label

Appointment Date	Appointment_Date__c	Date	
Appointment Name	Name	Auto Number	✓
Created By	CreatedById	Lookup(User)	
Customer Details	Customer_Details__c	Lookup(Customer Details)	✓
Customer Name	Customer_Name__c	Master-Detail(Customer Details)	✓
Last Modified By	LastModifiedById	Lookup(User)	
Maintenance service	Maintenance_service__c	Checkbox	
Repairs	Repairs__c	Checkbox	
Replacement Parts	Replacement_Parts__c	Checkbox	
Service Amount	Service_Amount__c	Currency(18, 0)	
Vehicle number plate	Vehicle_number_plate__c	Text(10) (Unique Case Insensitive)	✓



Billing details and feedback

Fields & Relationships
8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Billing details and feedback Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment Paid	Payment_Paid__c	Currency(18, 0)		
Payment Status	Payment_Status__c	Picklist		
Rating for service	Rating_for_service__c	Text(1)		
Service records	Service_records__c	Lookup(Service records)		✓

Field Creation

Includes Phone, Email, Text, Currency, Date, Picklist, Checkbox, and Formula fields with purpose and usage.

Checkbox Fields

- **Appointment:** Maintenance Service, Repairs, Replacement Parts
- **Service Records:** Quality Check Status

Date Fields

- **Appointment:** Appointment Date (Required)

Currency Fields

- **Appointment:** Service Amount
- **Billing Details and Feedback:** Payment Paid

Text Fields

- **Appointment:** Vehicle Number Plate (Required & Unique, Length 10)
- **Billing Details and Feedback:** Rating for Service (Length 1, Required)

Picklist Fields

- **Service Records:** Service Status (Started, Completed)
- **Billing Details and Feedback:** Payment Status (Pending, Completed)

Formula Fields

- **Service Records:** Service Date → `CreatedDate`

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can also hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.

Types of Fields

Standard Fields

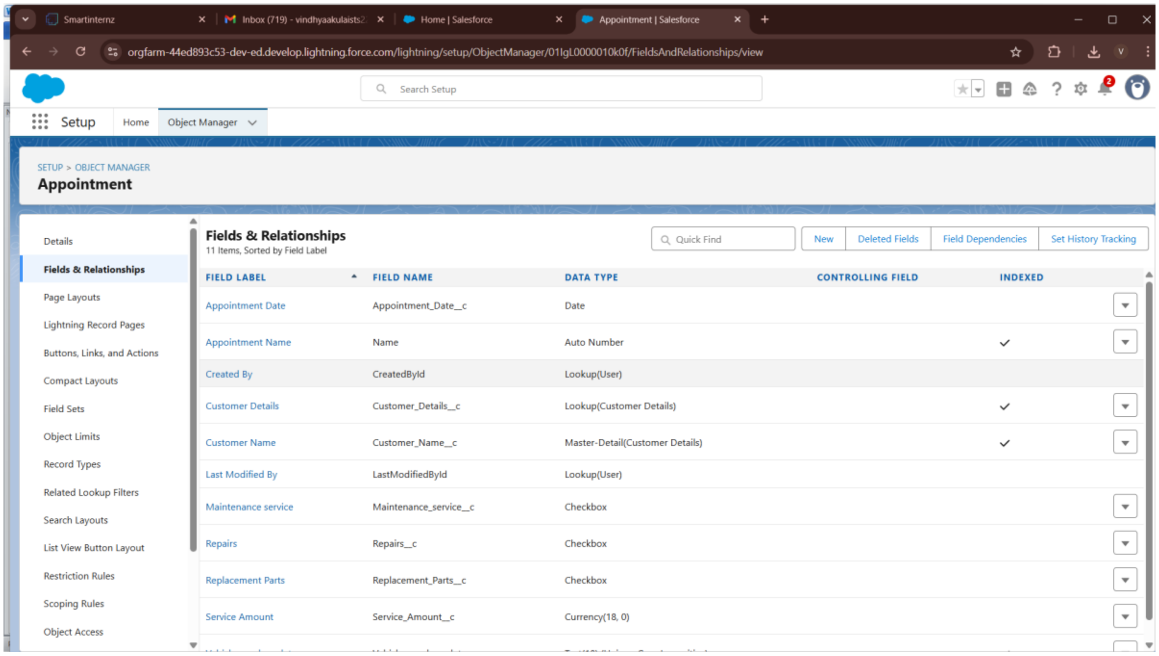
Custom Fields

Standard Fields:

As the name suggests, the Standard Fields are the predefined fields in Salesforce that perform a standard task. The main point is that you can't simply delete a Standard Field until it is a non-required standard field. Otherwise, users have the option to delete them at any point from the application freely. Moreover, we have some fields that you will find common in every Salesforce application. They are,

Custom Fields:

On the other side of the coin, Custom Fields are highly flexible, and users can change them according to requirements. Moreover, each organiser or company can use them if necessary. It means you need not always include them in the records, unlike Standard fields. Hence, the final decision depends on the user, and he can add/remove Custom Fields of any given form.



Appointment

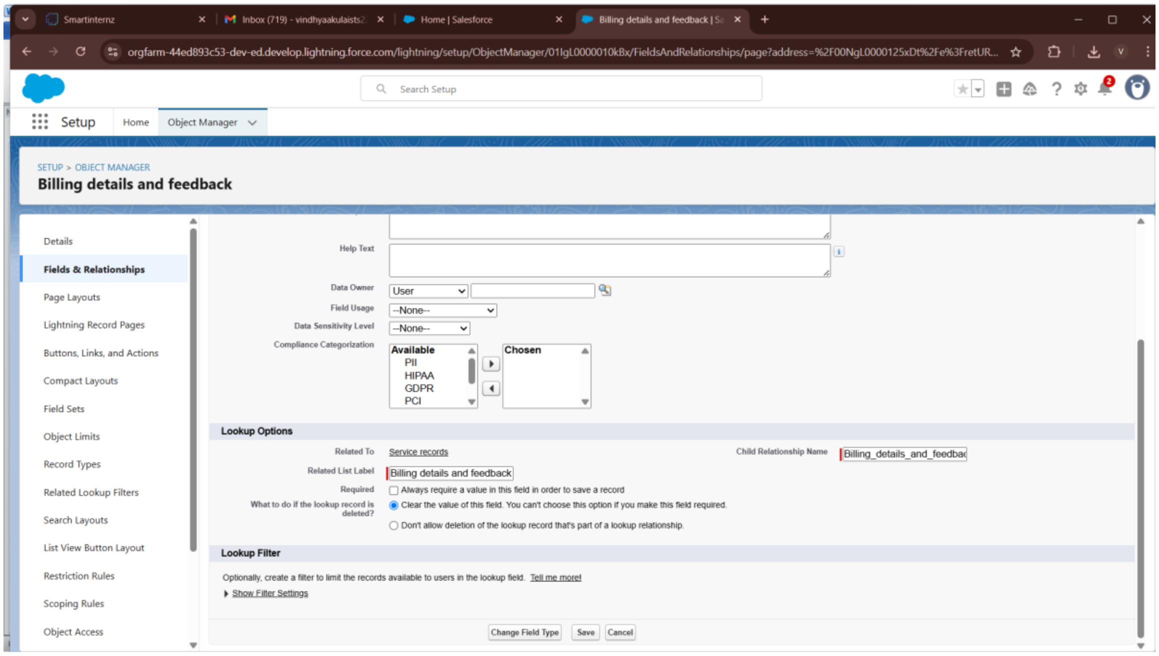
Fields & Relationships

11 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment Date	Appointment_Date__c	Date		
Appointment Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Customer Details	Customer_Details__c	Lookup(Customer Details)		✓
Customer Name	Customer_Name__c	Master-Detail(Customer Details)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Maintenance service	Maintenance_service__c	Checkbox		
Repairs	Repairs__c	Checkbox		
Replacement Parts	Replacement_Parts__c	Checkbox		
Service Amount	Service_Amount__c	Currency(18, 0)		

Relationships & Lookups

Describes lookup relationships between objects and their use in data hierarchy.



Help Text

Data Owner: User

Field Usage: --None--

Data Sensitivity Level: --None--

Compliance Categorization: Available: PII, HIPAA, GDPR, PCI; Chosen:

Lookup Options

Related To: Service records

Child Relationship Name: Billing_details_and_feedbac

Related List Label: Billing details and feedback

Required: □ Always require a value in this field in order to save a record

What to do if the lookup record is deleted?
● Clear the value of this field. You can't choose this option if you make this field required.
○ Don't allow deletion of the lookup record that's part of a lookup relationship.

Lookup Filter

Optionally, create a filter to limit the records available to users in the lookup field. Tell me more!

Show Filter Settings

Tabs & Lightning App Setup

Steps to create custom tabs and design a unified Lightning App page for easy navigation.

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:

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.

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.

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.

Lightning Component Tabs

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

Lightning Page Tabs

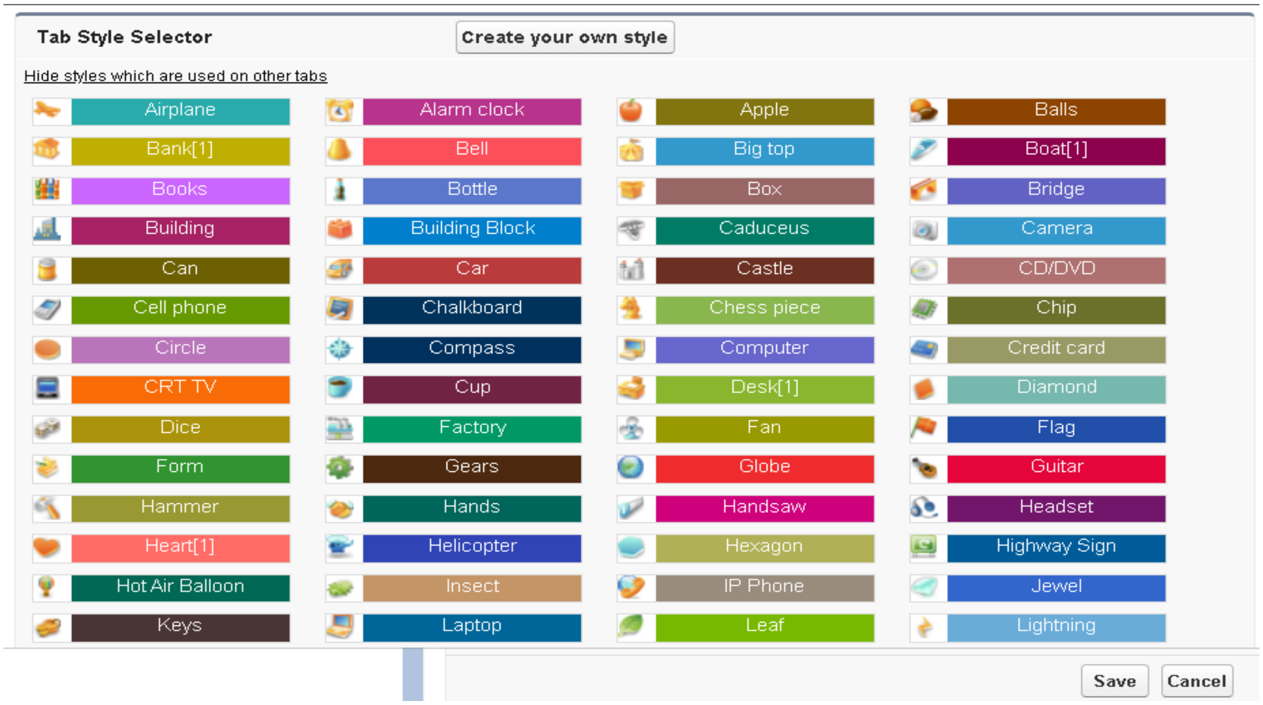
Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

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.

The Lightning App

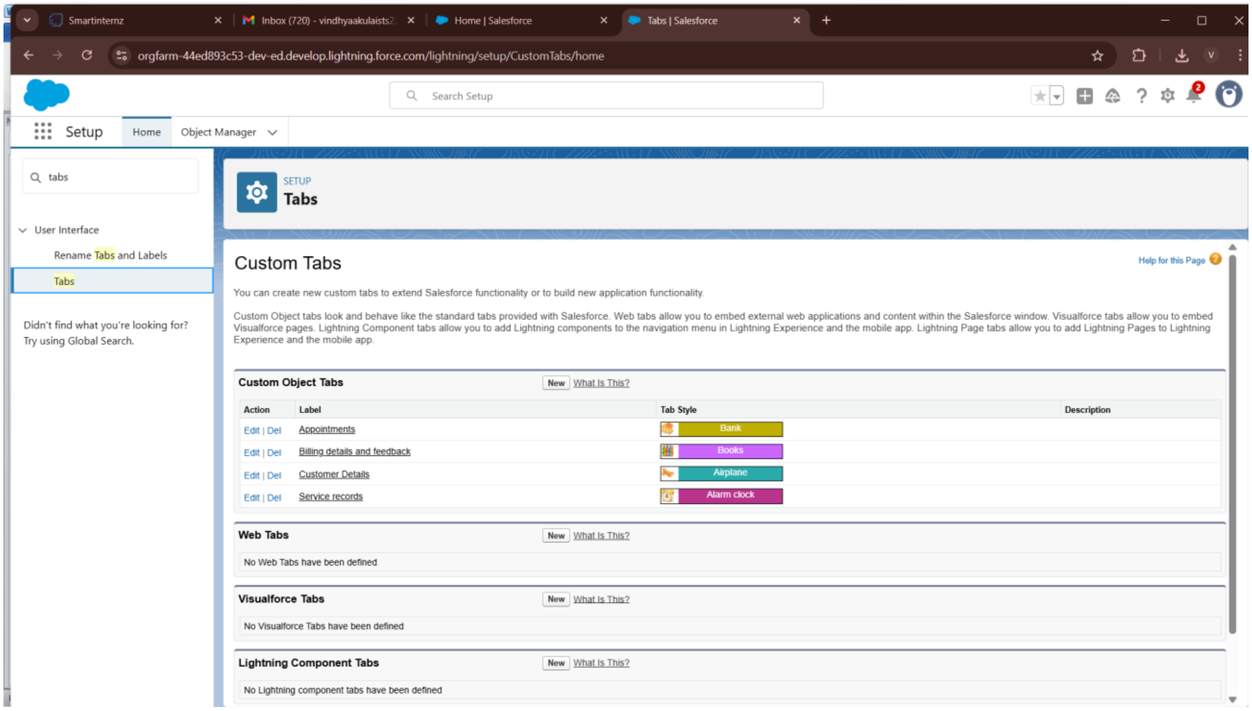
An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

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.



Tab Style Selector

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



Custom Tabs

Help for this Page

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs New What Is This?

Action	Label	Tab Style	Description
Edit | Del	Appointments	Bank	
Edit | Del	Billing details and feedback	Books	
Edit | Del	Customer Details	Airplane	
Edit | Del	Service records	Alarm clock	

Web Tabs New What Is This?

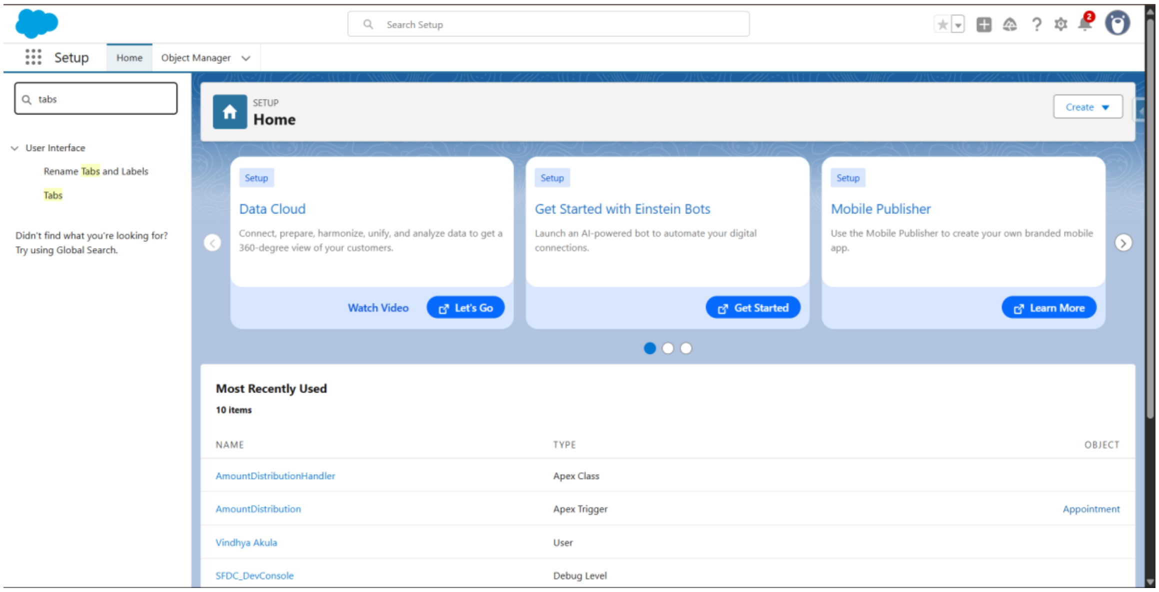
No Web Tabs have been defined

Visualforce Tabs New What Is This?

No Visualforce Tabs have been defined

Lightning Component Tabs New What Is This?

No Lightning component tabs have been defined


Search Setup
Setup
Home
Object Manager
tabs
User Interface
Rename Tabs and Labels
Tabs
Didn't find what you're looking for?
Try using Global Search.
SETUP
Home
Create
Setup
Data Cloud
Connect, prepare, harmonize, unify, and analyze data to get a 360-degree view of your customers.
Watch Video
Let's Go
Setup
Get Started with Einstein Bots
Launch an AI-powered bot to automate your digital connections.
Get Started
Setup
Mobile Publisher
Use the Mobile Publisher to create your own branded mobile app.
Learn More

Most Recently Used

10 items

NAME	TYPE	OBJECT
AmountDistributionHandler	Apex Class	
AmountDistribution	Apex Trigger	Appointment
Vindhya Akula	User	
SFDC_DevConsole	Debug Level	

Role Hierarchy & Profiles

Manager and Salesperson roles with related profiles and permissions setup.

Profile	Base Profile	Access Given
Manager	Standard User	Full access to all garage objects
Sales Person	Salesforce Platform	Read/Edit based on role hierarchy

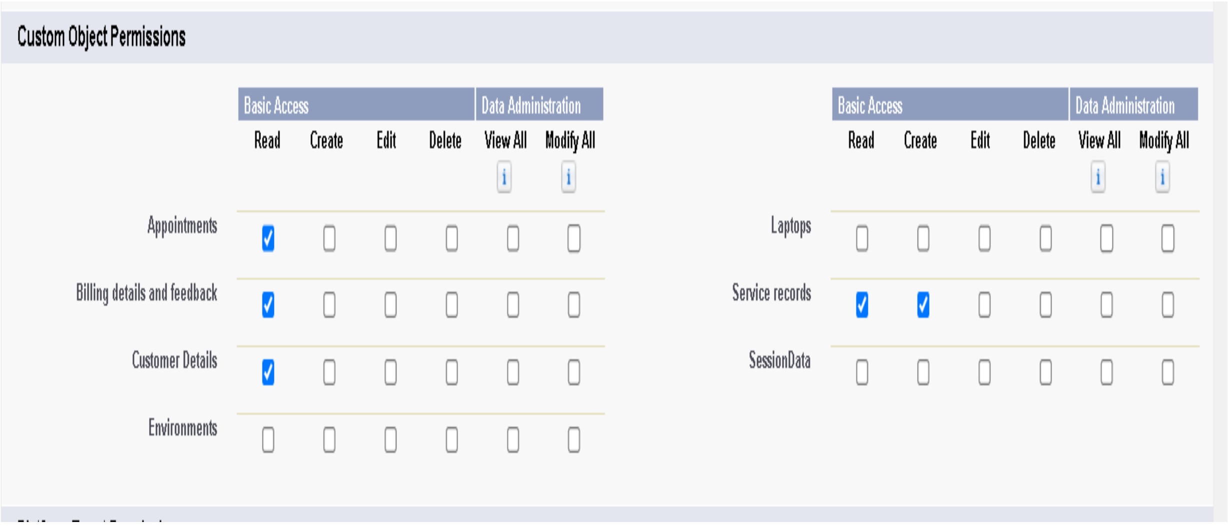
Roles

- Manager (under CEO)
- Sales Person (under Manager)

Users

At least 3 users created:

- 1 Manager Role, Manager Profile
- 2 Sales Person Role, Sales Person Profile



Custom Object Permissions

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Appointments	☑	□	□	□	□	□
Billing details and feedback	☑	□	□	□	□	□
Customer Details	☑	□	□	□	□	□
Environments	□	□	□	□	□	□

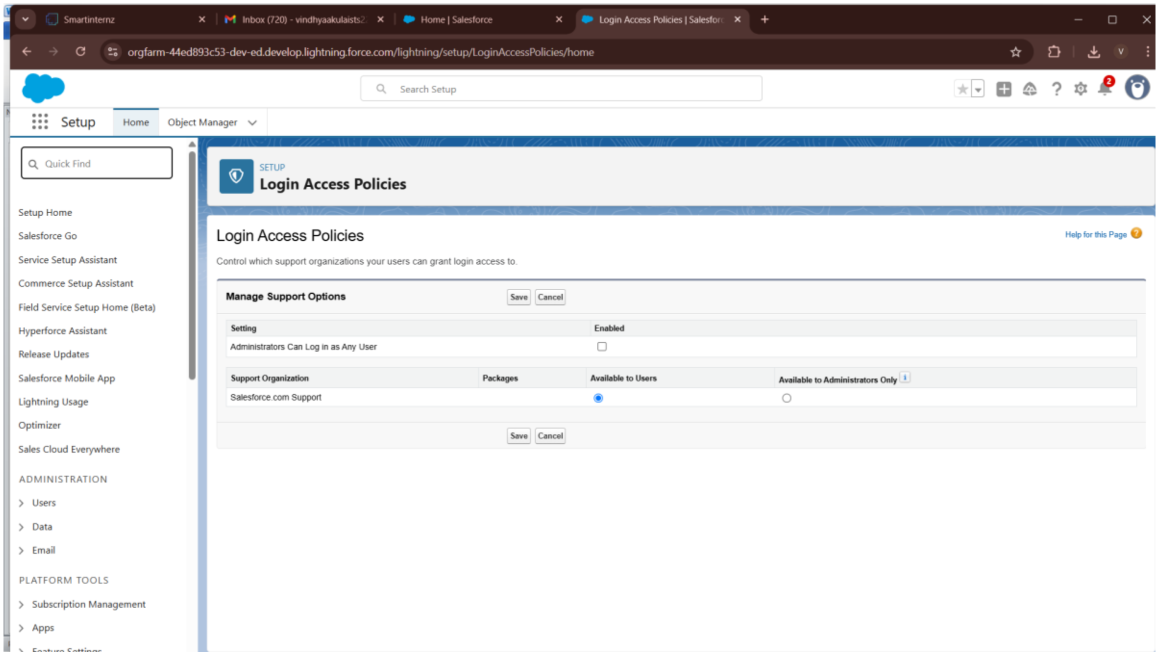
	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Laptops	□	□	□	□	□	□
Service records	☑	☑	□	□	□	□
SessionData	□	□	□	□	□	□

Security & Sharing Settings

OWD configurations, sharing rules, and public groups for secure data access.

- Assign roles (Manager, Salesperson) with appropriate permissions.

- Control access to records using sharing settings and public groups.



Login Access Policies

Login Access Policies

Help for this Page

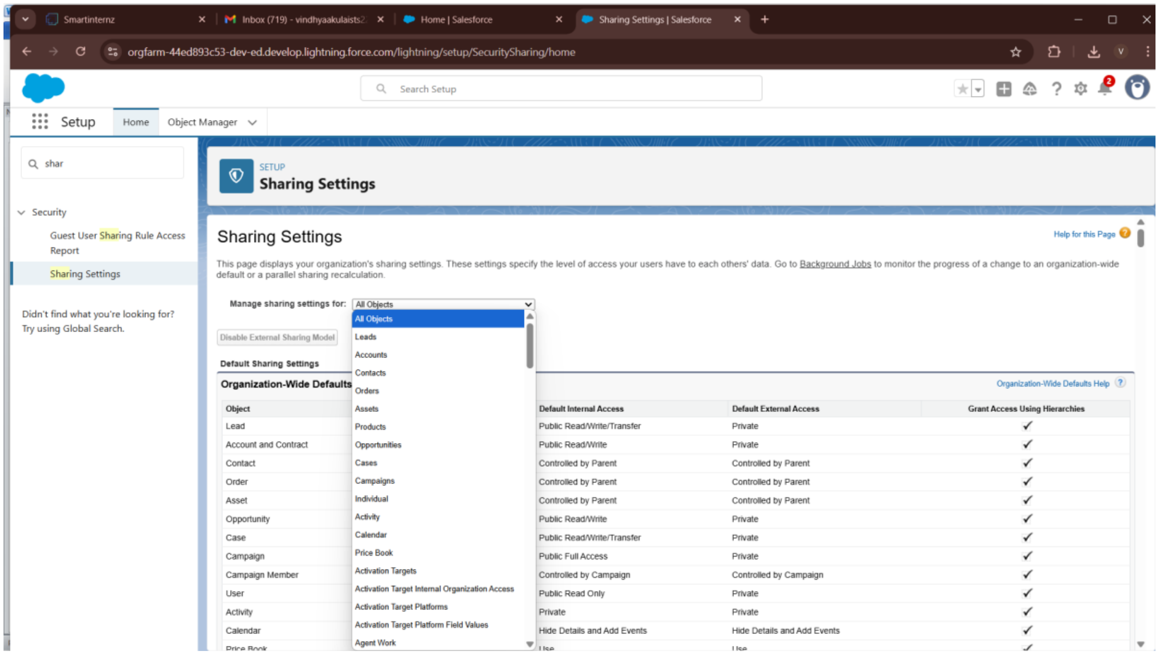
Control which support organizations your users can grant login access to.

Manage Support Options Save Cancel

Setting	Enabled
Administrators Can Log in as Any User	□

Support Organization	Packages	Available to Users	Available to Administrators Only
Salesforce.com Support		●	○

Save Cancel


Smartinternz
Inbox (719) - vindhyaakulaists2...
Home | Salesforce
Sharing Settings | Salesforce
orgfarm-44ed893c53-dev-ed.develop.lightning.force.com/lightning/setup/SecuritySharing/home
Search Setup
Setup
Home
Object Manager
shar
Security
Guest User Sharing Rule Access Report
Sharing Settings
Didn't find what you're looking for? Try using Global Search.
SETUP
Sharing Settings
Help for this Page
All Objects
Leads
Accounts
Contacts
Orders
Assets
Products
Opportunities
Cases
Campaigns
Individual
Activity
Calendar
Price Book
Activation Targets
Activation Target Internal Organization Access
Activation Target Platforms
Activation Target Platform Field Values
Agent Work
Disable External Sharing Model
Organization-Wide Defaults Help

Sharing Settings

This page displays your organization's sharing settings. These settings specify the level of access your users have to each others' data. Go to Background Jobs to monitor the progress of a change to an organization-wide default or a parallel sharing recalculation.

Manage sharing settings for: All Objects

Default Sharing Settings

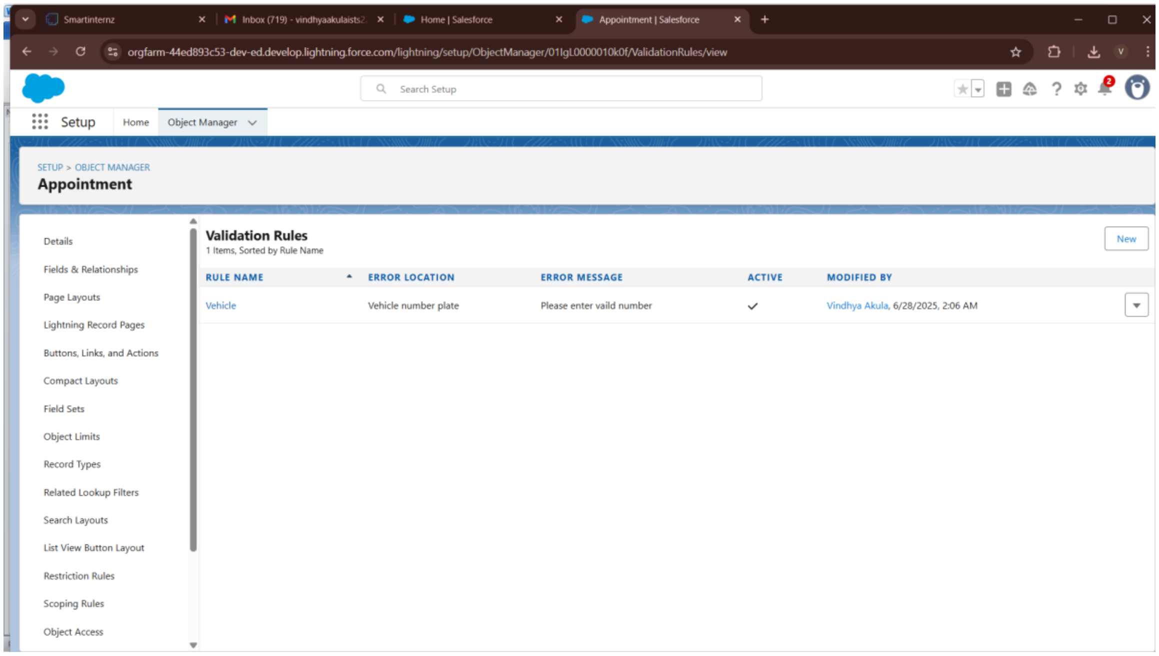
Organization-Wide Defaults

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Lead	Public Read/Write/Transfer	Private	✓
Account and Contract	Public Read/Write	Private	✓
Contact	Controlled by Parent	Controlled by Parent	✓
Order	Controlled by Parent	Controlled by Parent	✓
Asset	Controlled by Parent	Controlled by Parent	✓
Opportunity	Public Read/Write	Private	✓
Case	Public Read/Write/Transfer	Private	✓
Campaign	Public Full Access	Private	✓
Campaign Member	Controlled by Campaign	Controlled by Campaign	✓
User	Public Read Only	Private	✓
Activity	Private	Private	✓
Calendar	Hide Details and Add Events	Hide Details and Add Events	✓
Price Book	Use	Use	✓

Validation & Duplicate Rules

Enforce input patterns and prevent duplicate records via validation and matching rules.

Object	Rule Name	Condition	Error Message
Appointment	Vehicle	REGEX	"Please enter valid number"
Billing Details/Feedback	rating_should_be_less_than_5	REGEX	"Rating should be from 1 to 5"

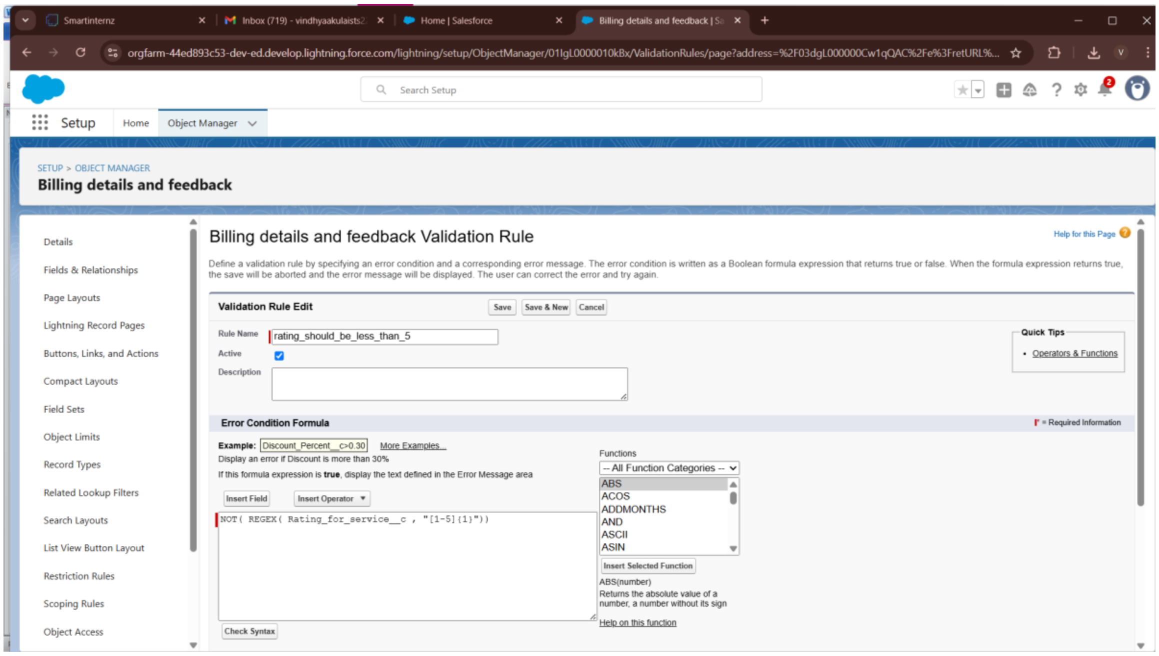

Smartinternz
Inbox (719) - vindhyaakulaists2
Home | Salesforce
Appointment | Salesforce
orgfarm-44ed893c53-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgL0000010k0f/ValidationRules/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER

Appointment

Validation Rules

1 Items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Vehicle	Vehicle number plate	Please enter vaild number	✓	Vindhya Akula, 6/28/2025, 2:06 AM



Billing details and feedback Validation Rule

Define a validation rule by specifying an error condition and a corresponding error message. The error condition is written as a Boolean formula expression that returns true or false. When the formula expression returns true, the save will be aborted and the error message will be displayed. The user can correct the error and try again.

Rule Name: rating_should_be_less_than_5

Active: ☑

Description:

Error Condition Formula

Example: Discount_Percent__c>0.30 More Examples...

Display an error if Discount is more than 30%

If this formula expression is **true**, display the text defined in the Error Message area

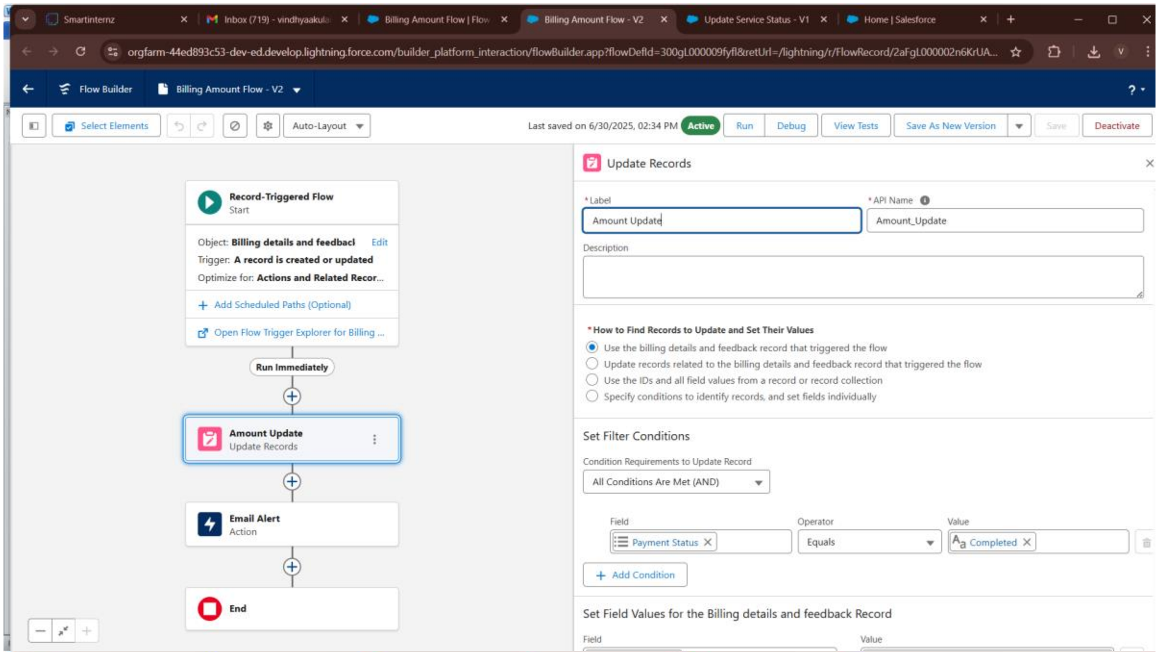
```
NOT( REGEX( Rating_for_service__c , "[1-5]{1}"))
```

ABS(number)

Returns the absolute value of a number, a number without its sign

Automation Using Flows

Two flows are created: one for auto-updating billing and emailing customers, and one for status updates.


Smartinternz
Inbox (719) - vindhyaakula
Billing Amount Flow | Flow
Billing Amount Flow - V2
Update Service Status - V1
Home | Salesforce
orgfarm-44ed893c53-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowDefId=300gL000009fyfl&retUrl=/lightning/r/FlowRecord/2aFgL000002n6KrUA...
Flow Builder
Billing Amount Flow - V2
Select Elements
Auto-Layout
Last saved on 6/30/2025, 02:34 PM
Active
Run
Debug
View Tests
Save As New Version
Save
Deactivate

Record-Triggered Flow
Start

Object: **Billing details and feedbacl** Edit
Trigger: **A record is created or updated**
Optimize for: **Actions and Related Recor...**

+ Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Billing ...

Run Immediately

Amount Update
Update Records

Email Alert
Action

End

Update Records

*Label: Amount Update
*API Name: Amount_Update
Description:

*How to Find Records to Update and Set Their Values
- ● Use the billing details and feedback record that triggered the flow
- ○ Update records related to the billing details and feedback record that triggered the flow
- ○ Use the IDs and all field values from a record or record collection
- ○ Specify conditions to identify records, and set fields individually

Set Filter Conditions

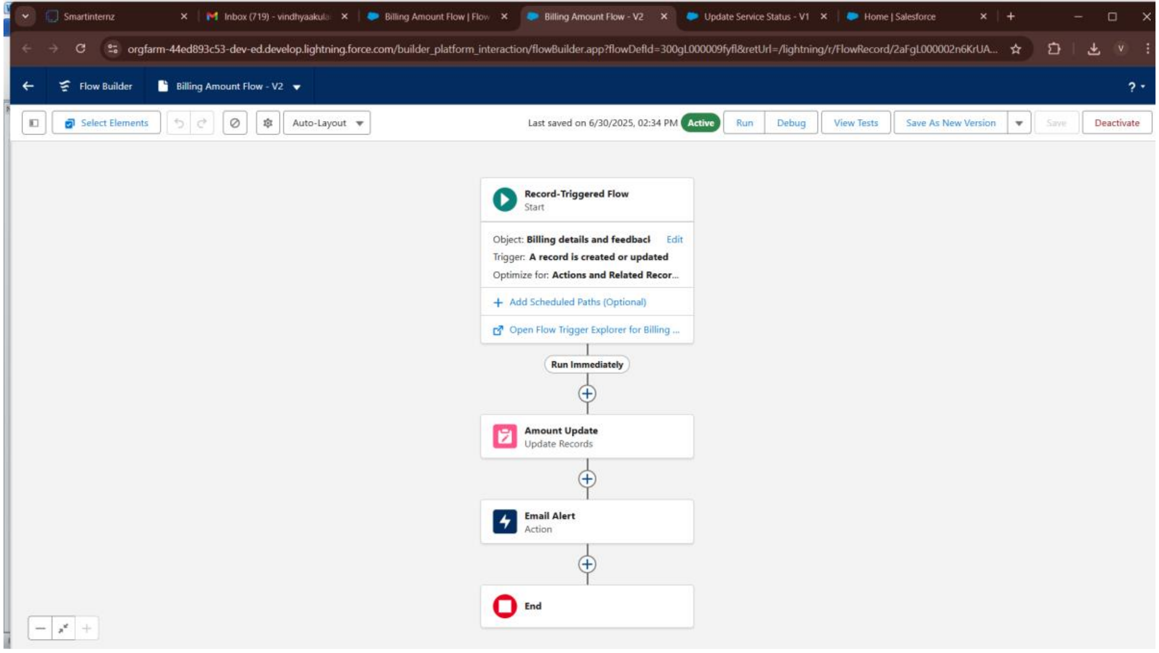
Condition Requirements to Update Record: All Conditions Are Met (AND)

Field	Operator	Value
Payment Status	Equals	Completed

+ Add Condition

Set Field Values for the Billing details and feedback Record

Field | Value


Smartinternz
Inbox (719) - vindhyaakula
Billing Amount Flow | Flow
Billing Amount Flow - V2
Update Service Status - V1
Home | Salesforce
orgfarm-44ed893c53-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowDefId=300gL000009fyfl&retUrl=/lightning/r/FlowRecord/2aFgL000002n6KrUA...
Flow Builder
Billing Amount Flow - V2
Select Elements
Auto-Layout
Last saved on 6/30/2025, 02:34 PM
Active
Run
Debug
View Tests
Save As New Version
Save
Deactivate

Record-Triggered Flow
Start

Object: **Billing details and feedbacl** Edit
Trigger: **A record is created or updated**
Optimize for: **Actions and Related Recor...**

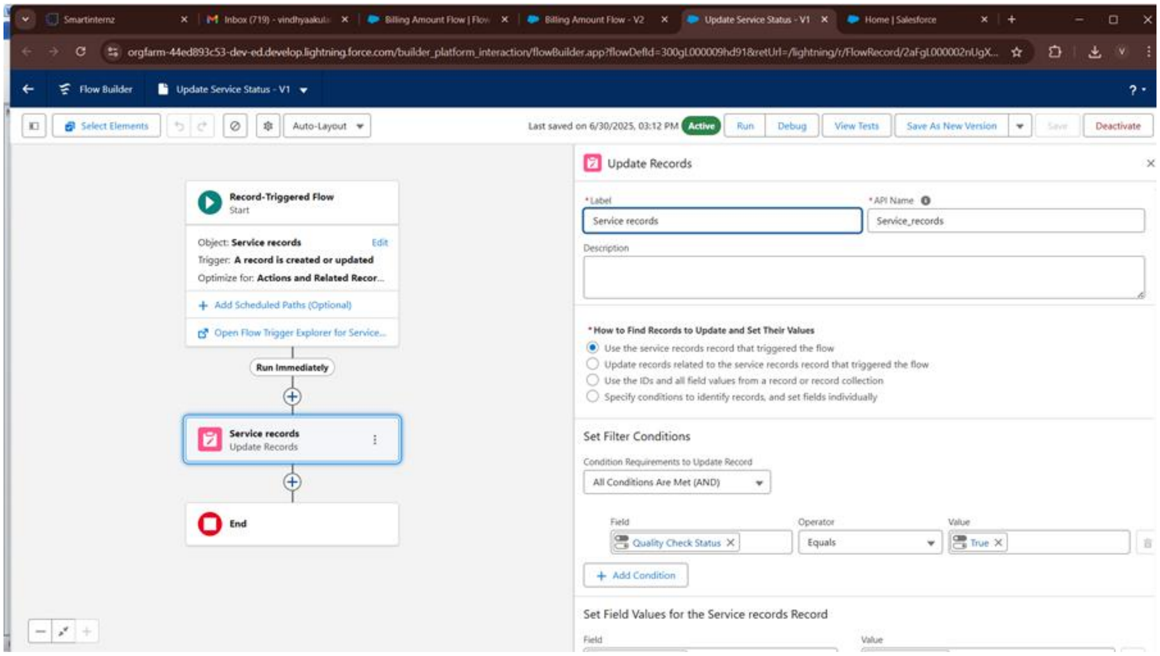
+ Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Billing ...

Run Immediately

Amount Update
Update Records

Email Alert
Action

End



Smartinternz
Inbox (719) - vindhyaakula
Billing Amount Flow | Flow
Billing Amount Flow - V2
Update Service Status - V1
Home | Salesforce
orgfarm-44ed893c53-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowDefId=300gL000009hd91&retUrl=/lightning/r/FlowRecord/2aFgL000002nUgX...
Flow Builder
Update Service Status - V1
Select Elements
Auto-Layout
Last saved on 6/30/2025, 03:12 PM
Active
Run
Debug
View Tests
Save As New Version
Save
Deactivate
Record-Triggered Flow
Start
Edit
Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Service...
Run Immediately
Service records
Update Records
End

Object: **Service records**
Trigger: **A record is created or updated**
Optimize for: **Actions and Related Recor...**

Update Records

*Label: Service records

*API Name: Service_records

Description:

*How to Find Records to Update and Set Their Values

- ● Use the service records record that triggered the flow
- ○ Update records related to the service records record that triggered the flow
- ○ Use the IDs and all field values from a record or record collection
- ○ Specify conditions to identify records, and set fields individually

Set Filter Conditions

Condition Requirements to Update Record: All Conditions Are Met (AND)

Field	Operator	Value
Quality Check Status	Equals	True

+ Add Condition

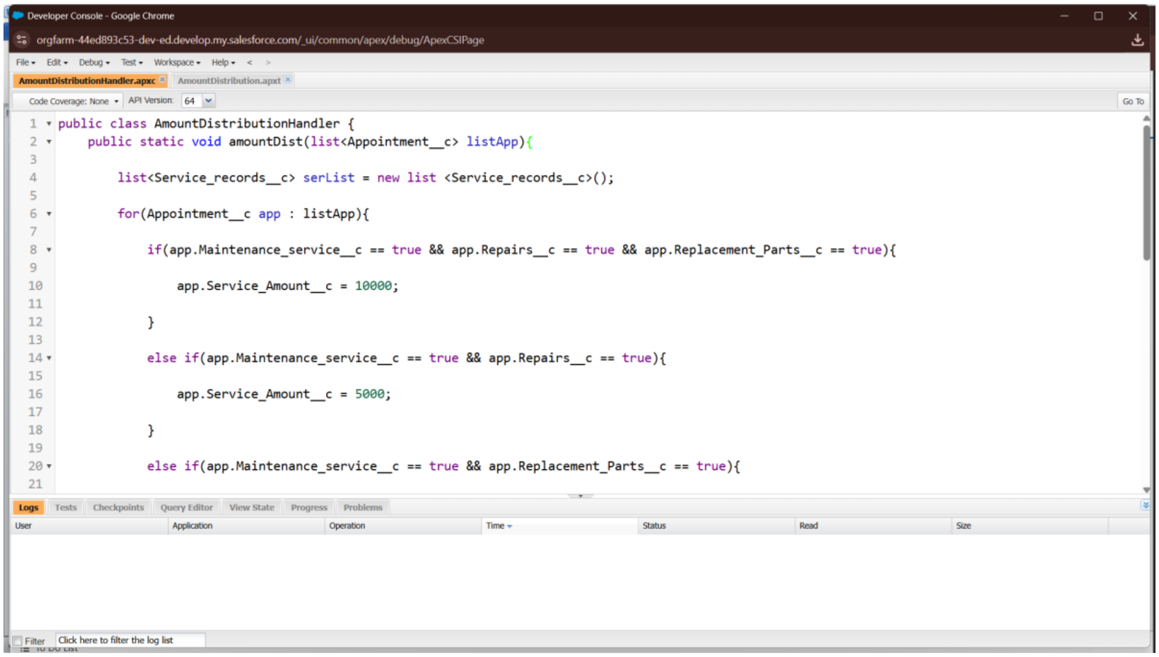
Set Field Values for the Service records Record

Field

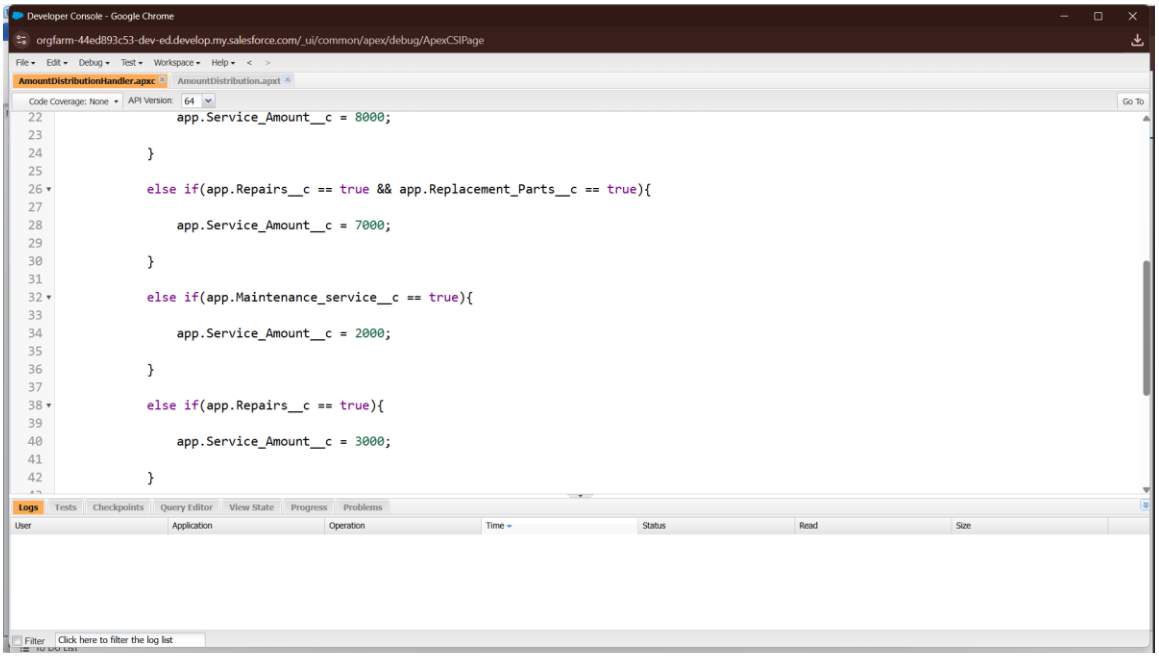
Value

Apex Programming

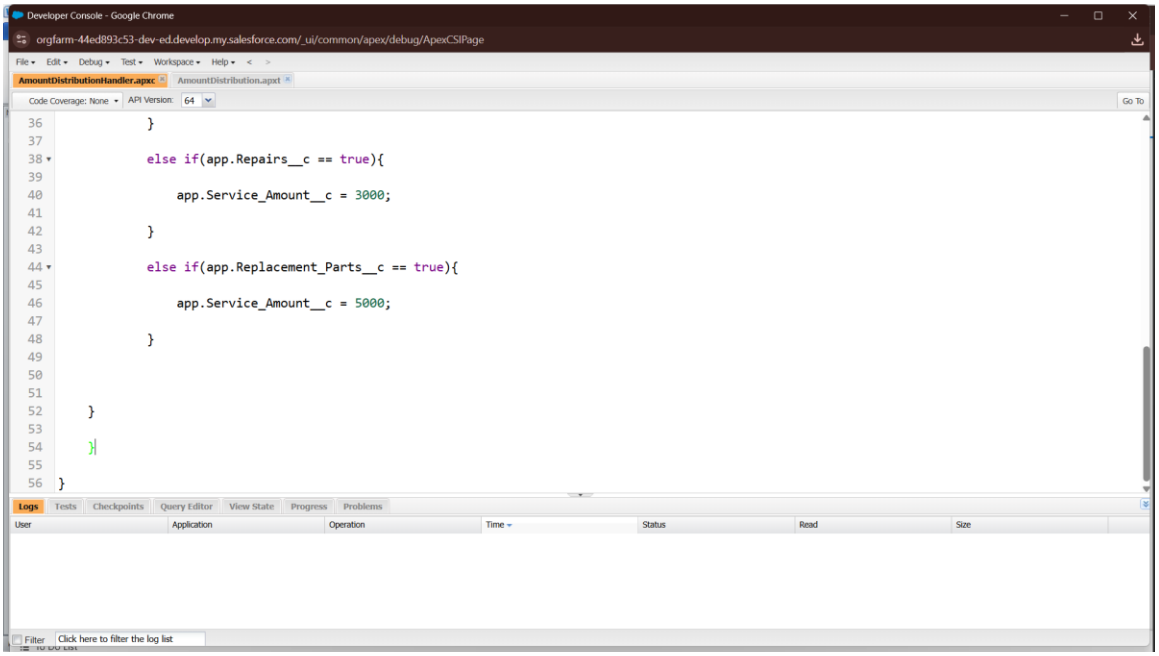
Custom Apex class and trigger to dynamically assign service cost based on selections.


Developer Console - Google Chrome
orgfarm-44ed893c53-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
AmountDistributionHandler.apxc
AmountDistribution.apxt
Code Coverage: None
API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter
Click here to filter the log list

```
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){

```


Developer Console - Google Chrome
orgfarm-44ed893c53-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
AmountDistributionHandler.apxc
AmountDistribution.apxt
Code Coverage: None
API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter
Click here to filter the log list

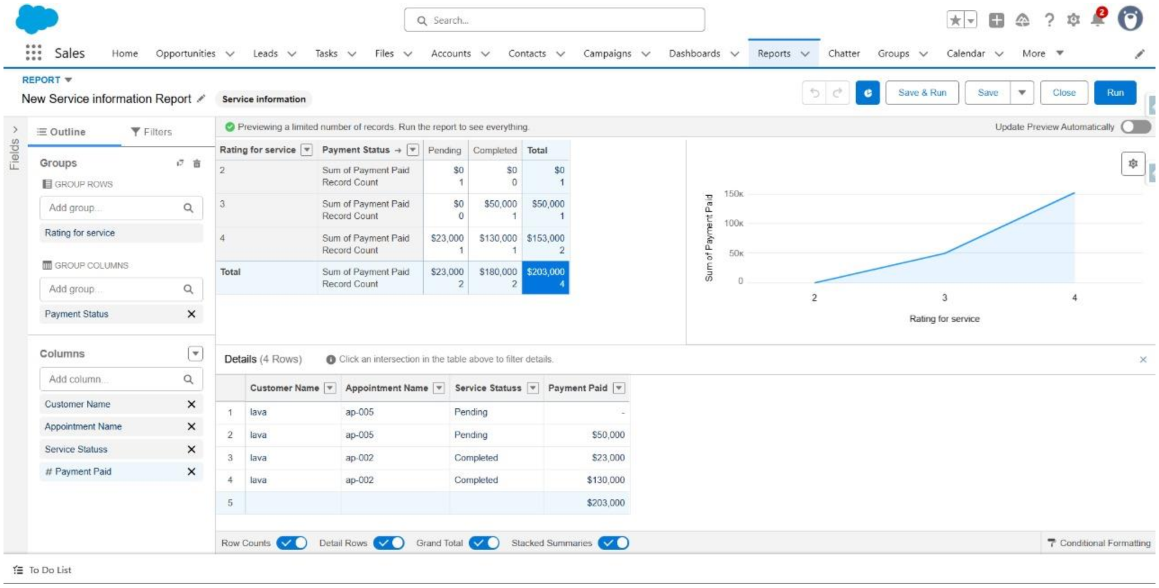
```
                app.Service_Amount__c = 8000;

            }

            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.Repairs__c == true){

                app.Service_Amount__c = 3000;

            }

            else if(app.Replacement_Parts__c == true){

                app.Service_Amount__c = 5000;

            }



    }

    }

}
```

Reports & Dashboards

Details the process of creating report types, reports, dashboards, and how to share them.



New Service information Report — Service information

Groups

GROUP ROWS: Rating for service

GROUP COLUMNS: Payment Status

Columns: Customer Name, Appointment Name, Service Statuss, # Payment Paid

Previewing a limited number of records. Run the report to see everything.

Update Preview Automatically

Rating for service	Payment Status →	Pending	Completed	Total
2	Sum of Payment Paid Record Count	$0 1	$0 0	$0 1
3	Sum of Payment Paid Record Count	$0 0	$50,000 1	$50,000 1
4	Sum of Payment Paid Record Count	$23,000 1	$130,000 1	$153,000 2
Total	Sum of Payment Paid Record Count	$23,000 2	$180,000 2	$203,000 4

Details (4 Rows) Click an intersection in the table above to filter details.

	Customer Name	Appointment Name	Service Statuss	Payment Paid
1	lava	ap-005	Pending	-
2	lava	ap-005	Pending	$50,000
3	lava	ap-002	Completed	$23,000
4	lava	ap-002	Completed	$130,000
5				$203,000

Row Counts ✓ Detail Rows ✓ Grand Total ✓ Stacked Summaries ✓

Glossary

Key terms used throughout the project with definitions.

This glossary contains important terms and concepts used throughout the **Garage Management System** Salesforce project. Understanding these will help readers and users better navigate and utilize the system effectively.

Apex

Apex is a strongly typed, object-oriented programming language provided by Salesforce. It allows developers to execute flow and transaction control statements on Salesforce servers in conjunction with calls to the API. In this project, Apex is used to calculate service charges automatically.

Auto Number

A field data type in Salesforce used to automatically generate sequential numbers for records. For example, appointments use an auto number format like `app-{000}` to generate unique identifiers.

Checkbox

A Boolean data type field that stores `true` or `false`. It is used in the project to represent whether services like repairs or maintenance were requested.

Custom Object

An object that is defined by the user to store information specific to the business. In this project, objects like `Customer Details`, `Appointments`, and `Service Records` are custom objects tailored to garage operations.

Dashboard

A visual display of key metrics and trends. In this project, dashboards are created to track service performance and customer feedback using components like charts and graphs.

Data Relationships

The connections between objects, established using lookup fields. For instance, `Service Records` are related to `Appointments`, which in turn are related to `Customer Details`.

Duplicate Rules

Salesforce feature that helps prevent users from creating duplicate records. This project includes duplicate rules for `Customer Details` based on email and phone number.

Email Alert

An automated message sent via Salesforce Flows. For example, after a customer makes a payment, an email is triggered to thank them and confirm receipt.

Flow

A point-and-click automation tool in Salesforce. It is used in this project to update records, send emails, and perform background tasks based on certain conditions.

Formula Field

A read-only field that automatically calculates a value based on a formula. Used in `Service Records` to show the service date.

Lightning App

A custom Salesforce application created using Lightning App Builder. It allows users to interact with tabs, dashboards, and objects in a unified UI. In this project, it's named **Garage Management Application**.

Lookup Relationship

A type of relationship between objects that allows users to select a record from another object. For instance, each `Appointment` record looks up to a `Customer Details` record.

Matching Rule

Used to identify duplicate records based on specified criteria. This project uses matching rules for `Customer Details` to match by email and phone.

Object Manager

A Salesforce setup area where users can create, modify, and manage objects, fields, and relationships. It is the starting point for building custom objects.

Organization-Wide Defaults (OWD)

Defines the baseline level of access users have to each object. In this system, `Service Records` are set to Private to protect service information.

Picklist

A field type that allows users to select from predefined values. Used for `Service Status` and `Payment Status`.

Profile

A collection of settings and permissions that defines what a user can do in Salesforce. Custom profiles like `Manager` and `Salesperson` were created in this project.

Public Group

A group of users that can be used in sharing rules or reports. The "Sales Team" public group was created to organize users with the `Salesperson` role.

Report

A way to view and analyze Salesforce data. Custom reports are used in this project to monitor appointments, billing, and customer ratings.

Role

Defines a user's position in the role hierarchy and their access to records owned by others. Roles like `Manager` and `Salesperson` define access in this project.

Sharing Rule

Grants specific users access to records they don't own. Used to allow managers access to service records owned by salespersons.

Validation Rule

Ensures data entered into Salesforce meets specific conditions. Examples include ensuring vehicle numbers follow a format and rating values are within 1–5.

Project Conclusion and Real-Time Usage

Conclusion

The **Garage Management System (GMS)** project built on Salesforce has successfully demonstrated how cloud-based CRM tools can be customized to serve a specific, industry-focused use case—in this case, vehicle service and garage operations. By leveraging Salesforce's declarative and programmatic capabilities, the project created a robust, scalable, and user-friendly system that manages the entire service lifecycle: from **customer onboarding**, **appointment scheduling**, and **service tracking**, to **billing**, **feedback collection**, and **reporting**.

Through the use of custom objects (`Customer Details`, `Appointments`, `Service Records`, and `Billing Details and Feedback`), the project models real-world garage workflows with high fidelity. Each component was designed to capture relevant data efficiently while ensuring data integrity and security using validation rules, matching rules, and role-based access controls.

Automation tools like **Flows** and **Apex triggers** help reduce manual work by updating payment statuses, calculating service charges, and sending email alerts. This not only improves operational efficiency but also enhances customer engagement and satisfaction. The integration of **Dashboards and Reports** further empowers the management to monitor business performance, customer trends, and revenue metrics in real-time.

This project highlights the true potential of Salesforce as a platform not only for traditional CRM but also for industry-specific custom applications.

Real-Time Usage Scenarios

The Garage Management System can be practically deployed in a variety of real-world settings, particularly in **automobile service centers**, **bike workshops**, and even **multi-brand vehicle garages**. Below are a few realistic use cases:

1. Appointment Scheduling and Service Intake

When a customer walks into a service center or books online, the service executive can quickly create an appointment in Salesforce. Based on the customer's chosen services (maintenance, repair, replacement), the system can automatically calculate the estimated charges.

2. Customer History and Service Tracking

For repeat customers, the system stores historical service data, appointment records, and previous feedback. This allows staff to offer personalized service recommendations or identify recurring issues with specific vehicles.

3. Billing and Follow-Up

Once the service is completed, the billing details are generated. The system checks if the payment is completed and, using Salesforce Flows, sends out a thank-you email with payment confirmation. This enhances customer trust and brand image.

4. Team Collaboration

Sales Managers and Service Technicians can use their respective profiles and roles to collaborate without compromising data security. Managers can oversee work done by the sales team, while salespersons focus on their assigned appointments.

5. Business Analytics

Owners or senior managers can track service volumes, payment trends, and customer satisfaction levels using real-time reports and dashboards. This helps in business planning, marketing strategy, and customer retention.

6. Franchise-Level Operations

For larger garages with multiple locations, the Salesforce-based system can be scaled and modified to manage data across branches, enforce standard operating procedures, and ensure consistency in customer experience.

Final Note

This project is a strong foundation for garage owners looking to transition from manual or spreadsheet-based operations to a fully digitized and cloud-managed ecosystem. With further integration possibilities (e.g., SMS alerts, vehicle diagnostics, customer portals), this system can evolve into a full-featured **Automobile Service CRM**.