

SALESFORCE DEVELOPER PROJECT



By  SmartInternz SmartInternz

Garage Management System-Salesforce CRM Solution for Automotive Service Centers

By

Kella. Chiranjeevi Chinmai Sai

Email: 21131a0592@gvpce.ac.in

Computer Science and Engineering (2021-2025)

Gayatri Vidya Parishad College of Engineering (Autonomous)

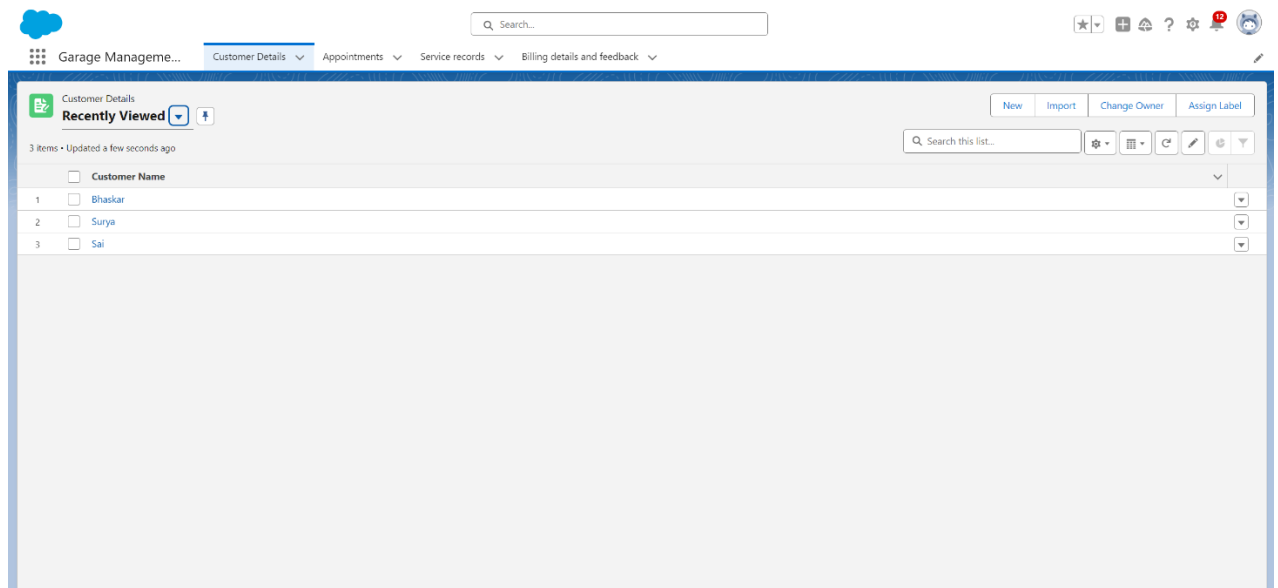
Garage Management System - Salesforce CRM Solution for Automotive Service Centers

Project Overview

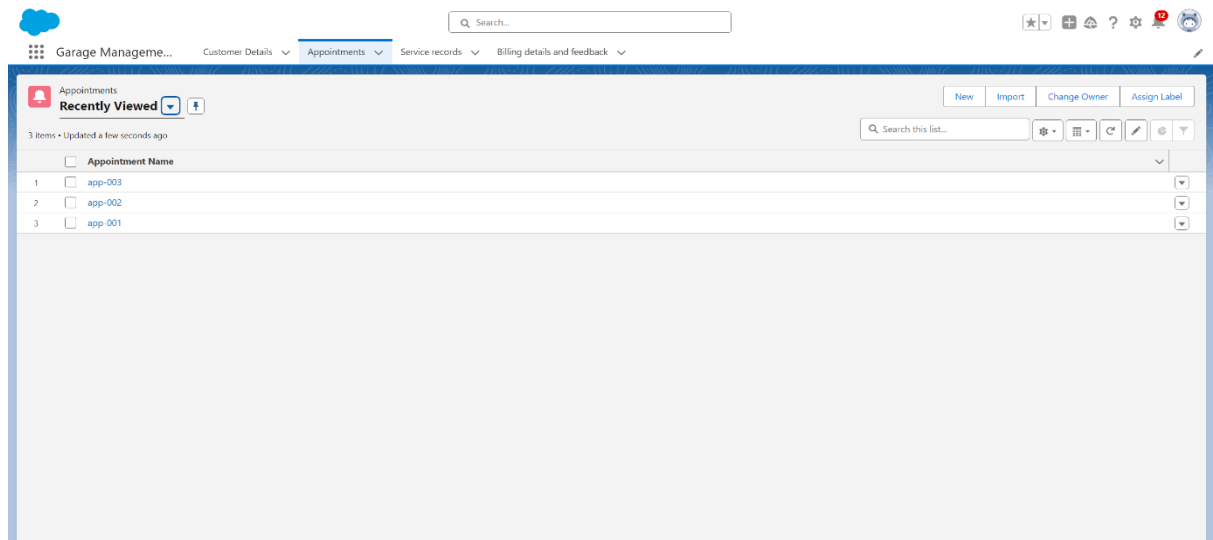
The **Garage Management System (GMS)** is a comprehensive CRM solution built on Salesforce, designed specifically for automotive service centers. This system streamlines customer management, appointment scheduling, service tracking, and billing, creating a centralized platform that efficiently handles the customer lifecycle—from initial booking to post-service feedback.

Key Features

- **Customer details:**
 - Store and manage customer details including contact information, service history, and preferences.
 - Access comprehensive service records for customer insights and tailored service.

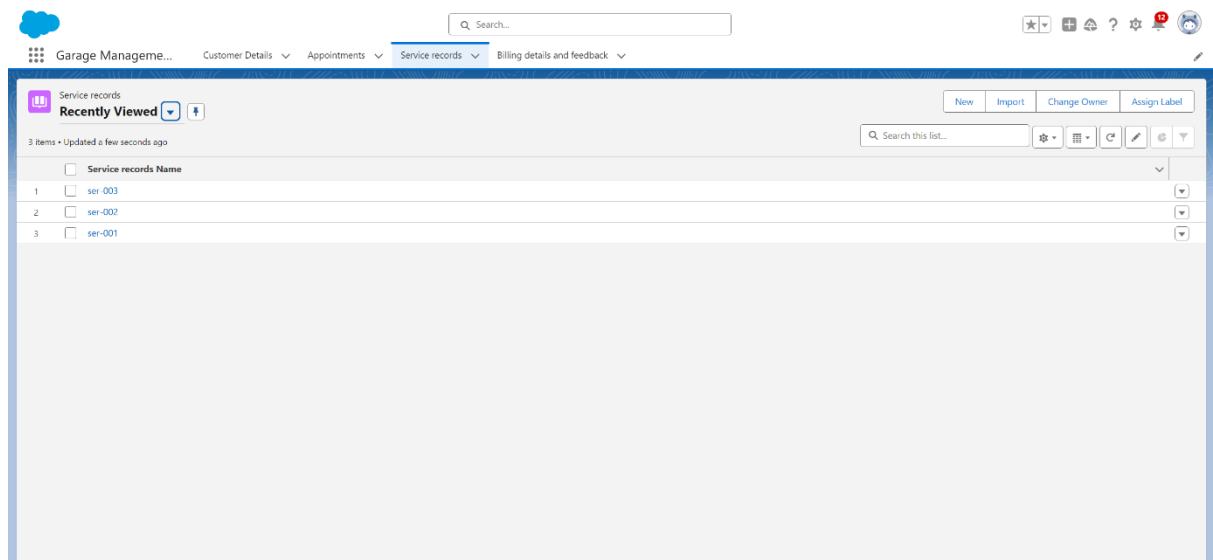


- **Appointment:**
 - Schedule and manage appointments with mechanics.
 - Automated reminders and notifications to reduce no-shows.
 - Integrated calendar to optimize working time.



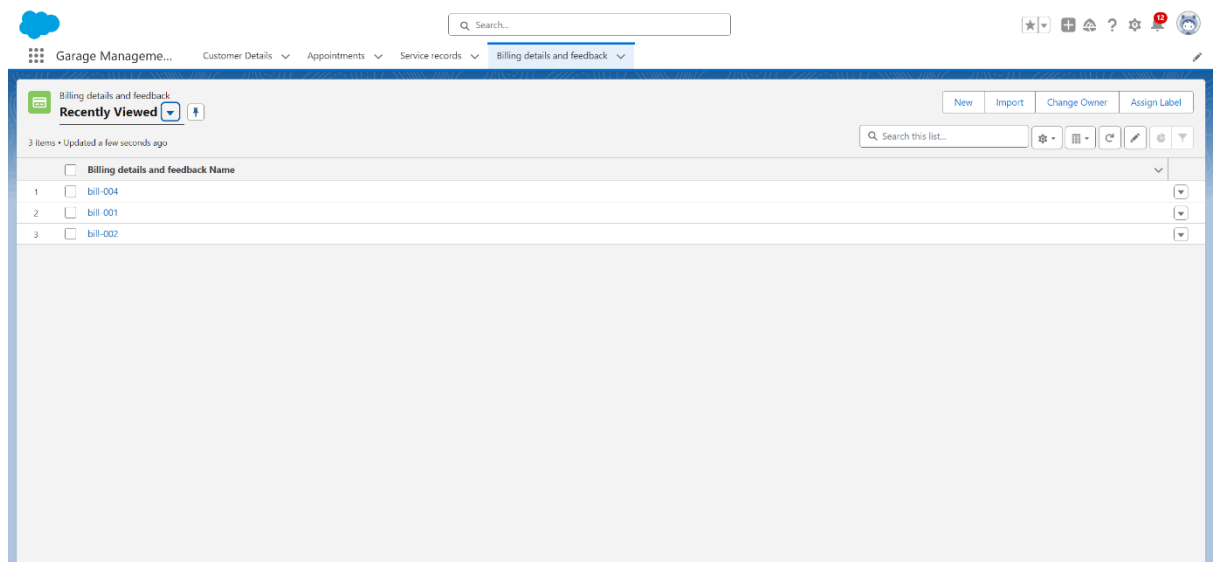
- **Service Records:**

- Track detailed service information including service type, parts used, and associated costs.
- Link service records to specific customer appointments for easy tracking and reference.



- **Billing details:**

- Generate and manage billing details, linking them to service records.
- Track payment status and send timely payment reminders.
- Automated invoice generation and email notifications for a streamlined payment process.



- **Feedback:**
 - Collect and manage customer feedback post-service.
 - Track ratings and comments to enhance service quality.
 - Analyze feedback trends for continuous improvement.

Technical Architecture

Custom Objects

- Customer details
- Appointment
- Service Records
- Billing details & Feedback

Object Relationships

- Appointment → Customer details (Lookup)
- Service Records → Appointment (Lookup)
- Billing details and feedback → Service Records (Lookup)

Automation

- **Flows:**
 - **Record-trigger Flow:** Manages customer verification, appointment, and reminder notifications.
 - **Amount Update Flow:** Automatically generates a billing record upon service completion and sends payment reminders.
 - **Email Alert Flow:** Sends a feedback request to customers post-service.

- **Approval Processes:** Approvals for high-value repairs or additional services, with manager-based routing and email notifications for approvals and rejections.
- **Apex Triggers:**
 - **AmountDistributionTrigger:** Validates business hours for appointment and Automatically calculates service costs and updates billing status upon payment.

User Interface

- **Lightning Components:**
 - Custom Lightning App: "Garage Management System"
 - Custom Home Page for easy navigation and quick access to customer and appointment details.
 - Streamlined navigation with essential tabs and dashboards for Sales person and managers.

Setup Instructions

1. **Object Creation:**
 - Create Customer details and Service Records objects and data.
 - Create Appointment, Billing details, and Feedback objects.
 - Create relationship fields to link objects as needed.
2. **User Configuration:**
 - Set up profiles for Sales person and Managers.
 - Configure user permissions and hierarchies for different user roles.
3. **Flow Deployment:**
 - Deploy the Record-trigger Flow.
 - Deploy the Amount update Flow.
 - Deploy the Email alert Flow.
4. **Lightning App Setup:**
 - Deploy the Garage Management System Lightning App.
 - Configure home page layouts to suit user roles.
 - Assign app permissions to profiles.

Process

Activity 1: Creating a Salesforce Developer Account and Activation

1. **Sign Up:** Visit [Salesforce Developer Sign Up](#).
2. **Fill Out the Form:** Enter the required details in the sign-up form and submit.
3. **Email Verification:** Check your email inbox (wait 5-10 minutes if needed) and click "Verify Account" to activate.
4. **Set Up Password:** Create a password and select a security question, then click "Change Password."
5. **Access Salesforce Setup:** You will be redirected to your Salesforce setup page.

Activity 2: Creating Custom Objects

1. **Navigate to Custom Object Creation:**
2. From the Setup page, go to **Object Manager > Create > Custom Object**.
3. **Enter Object Details:**
4. **Label Name and Plural Label:**
 - "Customer Details" (singular and plural: Customer Details)
 - "Appointment" (singular: Appointment, plural: Appointments)
 - "Service Records" (singular and plural: Service Records)
 - "Billing Details and Feedback" (singular and plural: Billing Details and Feedback)
5. **Set Record Name and Format:**
6. Enter the **Record Name** (e.g., "Customer Name," "Appointment Name," "Service Records Name," "Billing Details and Feedback Name").
7. Choose **Data Type**:
 - Text for Customer Details
 - Auto Number for others
 - Specify **Display Format** (e.g., "app-{000}", "ser-{000}", "bill-{000}")
 - **Starting Number:** 1
8. **Configure Settings:**
9. Enable **Allow Reports**, **Track Field History**, and **Allow Search**.

10. Click **Save**.

SETUP

New Custom Object

Custom Object Definition Edit

SaveSave & NewCancel

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label

Customer Details

Example: Account

Plural Label

Customer Details

Example: Accounts

Starts with vowel sound

☐

The Object Name is used when referencing the object via the API.

Object Name

Customer_Details

Example: Account

Description

Context-Sensitive Help Setting

☒ Open the standard Salesforce.com Help & Training window

☐ Open a window using a Visualforce page

Context Name

[None]

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name

Customer Name

Example: Account Name

Data Type

Text

Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Optional Features

SETUP > OBJECT MANAGER

Appointment

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Details

EditDelete

Description

API Name

Appointment__c

Custom

☒

Singular Label

Appointment

Plural Label

Appointments

Enable Reports

☒

Track Activities

☐

Track Field History

☒

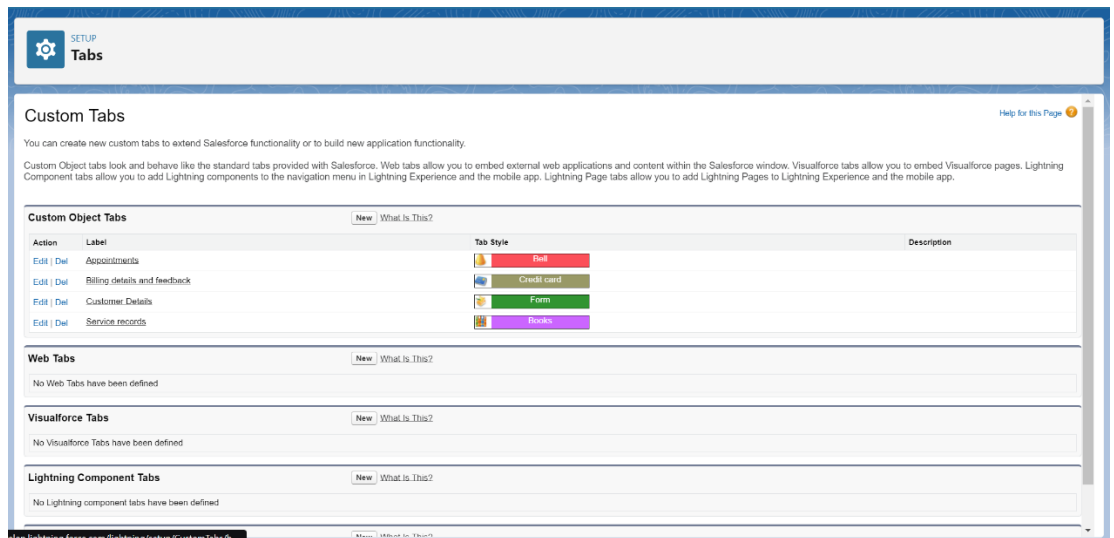
Deployment Status

Deployed

Help Settings

Standard salesforce.com Help Window

- Repeat these steps for **Appointments**, **Service Records**, and **Billing Details and Feedback** objects.



2. Create a Lightning App Page:

- Go to the **Setup** page > search **App Manager** in Quick Find > select **App Manager**.
- Click on **New Lightning App** and follow the prompts to create the app.
- Click **Save** when done.

Activity 4: Setting Up Fields and Relationships

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

1. Standard Fields
2. 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,

- Created By
- Owner
- Last Modified
- Field Made During object Creation

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.

We setup various fields in custom Salesforce objects (e.g., Customer Details, Appointments, Service Records, Billing Details, and Feedback) to capture and organize specific data types, such as phone numbers, dates, and statuses. This setup includes creating relationships, defining data entry requirements, and configuring display formats to streamline data tracking and reporting.

SETUP > OBJECT MANAGER

Customer Details

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Object Access

Triggers

Fields & Relationships

6 Items, Sorted by Field Label

Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

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			

SETUP > OBJECT MANAGER

Appointment

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Object Access

Triggers

Fields & Relationships

11 Items, Sorted by Field Label

Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	
Appointment Date	Appointment.Date__c	Date			
Appointment Name	Name	Auto Number		✓	
Created By	CreatedById	Lookup(User)			
Customer Details	Customer_Details__c	Lookup(Customer Details)		✓	
Last Modified By	LastModifiedById	Lookup(User)			
Maintenance service	Maintenance.service__c	Checkbox			
Owner	OwnerId	Lookup(User,Group)		✓	
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)		✓	

SETUP > OBJECT MANAGER

Service records

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Object Access

Triggers

Details

Description

API Name

Service_records__c

Custom

✓

Singular Label

Service records

Plural Label

Service records

Enable Reports

✓

Track Activities

✓

Track Field History

✓

Deployment Status

Deployed

Help Settings

Standard salesforce.com Help Window

Edit

Delete

SETUP > OBJECT MANAGER

Billing details and feedback

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Object Access

Triggers

Fields & Relationships

8 Items. Sorted by Field Label

Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

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

Activity 5: Creating Validation and Duplicate Rules

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

Service records Validation Rule

[Help for this Page](#) ?

[Back to Service records](#)

Validation Rule Detail

[Edit](#) [Clone](#)

Rule Name	service_status_note	Active	<input checked="" type="checkbox"/>
Error Condition Formula	NOT(ISPICKVAL(Service_Status__c , "Completed"))		
Error Message	still it is pending	Error Location	Service Status
Description			
Created By	Sai Kella , 29/10/2024, 1:18 pm	Modified By	Sai Kella , 29/10/2024, 1:18 pm

[Edit](#) [Clone](#)

Billing details and feedback Validation Rule

[Help for this Page](#) ?

[Back to Billing details and feedback](#)

Validation Rule Detail

[Edit](#) [Clone](#)

Rule Name	rating_should_be_less_than_5	Active	<input checked="" type="checkbox"/>
Error Condition Formula	NOT(REGEX(Rating_for_service__c , "[1-5]({1})"))		
Error Message	rating should be from 1 to 5	Error Location	Rating for service
Description			
Created By	Sai Kella , 29/10/2024, 1:19 pm	Modified By	Sai Kella , 29/10/2024, 1:19 pm

[Edit](#) [Clone](#)

Billing details and feedback Validation Rule ~ Salesforce - Developer Edition

3. Create a Matching Rule:

- In **Setup**, search for **Matching Rule** > click **New Rule** > configure for the Customer Details object.

4. Create a Duplicate Rule:

- In **Setup**, search for **Duplicate Rules** > click **New Rule** > select Customer Details object.
- Name the rule "Customer Detail duplicate," select the **Matching Rule** ("Matching customer details") > Save and Activate.

Customer Details Duplicate Rule

[Help for this Page](#)

Customer Detail duplicate

Duplicate Rule Detail

Edit

Delete

Clone

Deactivate

Rule Name	Customer Detail duplicate	Order	1 of 1 [Reorder]
Description			
Object	Customer Details		
Record-Level Security	Enforce sharing rules		
Action On Create	Allow	Operations On Create	<input checked="" type="checkbox"/> Alert <input checked="" type="checkbox"/> Report
Action On Edit	Allow	Operations On Edit	<input type="checkbox"/> Alert <input type="checkbox"/> Report
Alert Text	Use one of these records?		
Active	<input checked="" type="checkbox"/>		
Matching Rule	Matching customer details Mapped	Matching Criteria	(Customer Details: Gmail EXACT MatchBlank = FALSE) AND (Customer Details: Phone_number EXACT MatchBlank = FALSE)
Conditions			
Created By	Sai Kella , 29/10/2024, 1:22 pm	Modified By	Sai Kella , 29/10/2024, 1:22 pm

Edit

Delete

Clone

Deactivate

Activity 6: Configuring Profiles and Roles

A profile is a group/collection of settings and permissions that define what a user can do in salesforce. Profile controls “Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges. You can define profiles by the user's job function. For example System Administrator, Developer, Sales Representative.

Types of profiles in salesforce

1. Standard profiles:

By default salesforce provides below standard profiles.

- Contract Manager
- Read Only
- Marketing User

Solutions Manager

- Standard User
- System Administrator.

We cannot delete standard ones

Each of these standard ones includes a default set of permissions for all of the standard objects available on the platform.

2. Custom Profiles:

Custom ones defined by us.

They can be deleted if there are no users assigned with that particular one.

Profile Manager

Help for this Page ?

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Login IP Ranges [0] | Enabled Apex Class Access [0] | Enabled Visualforce Page Access [0] | Enabled External Data Source Access [0] | Enabled Named Credential Access [0] | Enabled External Credential Principal Access [0] | Enabled Custom Metadata Type Access [0] | Enabled Custom Setting Definitions Access [0] | Enabled Flow Access [0] | Enabled Service Presence Status Access [0] | Enabled Custom Permissions [0]

Profile Detail

EditCloneDeleteView Users

Name	Manager		
User License	Salesforce	Custom Profile	<input checked="" type="checkbox"/>
Description			
Created By	Sai Kella, 29/10/2024, 1:24 pm	Modified By	Sai Kella, 29/10/2024, 1:26 pm

Page Layouts

Standard Object Layouts

Global	Global Layout [View Assignment]	Invoice	Invoice Layout [View Assignment]
Email Application	Not Assigned [View Assignment]	Invoice Line	Invoice Line Layout [View Assignment]
Home Page Layout	DE Default [View Assignment]	Lead	Lead Layout [View Assignment]
Account	Account Layout	Legal Entity	Legal Entity Layout

sales person



Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

[Login IP Ranges \[0\]](#) |
 [Enabled Apex Class Access \[0\]](#) |
 [Enabled Visualforce Page Access \[0\]](#) |
 [Enabled External Data Source Access \[0\]](#) |
 [Enabled Named Credential Access \[0\]](#) |
 [Enabled External Credential Principal Access \[0\]](#) |
 [Enabled Custom Metadata Type Access \[0\]](#) |
 [Enabled Custom Setting Definitions Access \[0\]](#) |
 [Enabled Flow Access \[0\]](#) |
 [Enabled Service Presence Status Access \[0\]](#) |
 [Enabled Custom Permissions \[0\]](#)

Edit

Delete

[Edit](#) [Clone](#) [Delete](#) [View Users](#)

Name	sales person		
User License	Salesforce Platform	Custom Profile	<input checked="" type="checkbox"/>
Description			
Created By	<u>Sai Kella</u> , 29/10/2024, 1:28 pm	Modified By	<u>Sai Kella</u> , 29/10/2024, 1:29 pm

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

[illegible]

Activity 7: Creating Users, Public Groups and Sharing Settings

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

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

Sharing Setting

Salesforce allows you to configure sharing settings to control how records are accessed and shared within your organization. These settings are crucial for maintaining data security and privacy. Salesforce provides a variety of tools and mechanisms to define and enforce sharing rules, such as:

Organization-Wide Default (OWD) Settings:

These settings define the default level of access for all objects within your Salesforce org.

OWD settings include Private, Public Read-Only, Public Read/Write, and Controlled by Parent.

OWD settings can be configured for each standard and custom object.

Role Hierarchy:

Salesforce uses a role hierarchy to determine record access.

Users at higher levels in the hierarchy have greater access to records owned by or shared with users lower in the hierarchy.

The role hierarchy is often used in combination with OWD settings to grant different levels of access.

Profiles and Permission Sets:

Profiles and permission sets allow administrators to specify object-level and field-level permissions for users.

Profiles are typically used to grant general object and field access, while permission sets can be used to extend those permissions to specific users.

Sharing Rules:

Sharing rules are used to extend access to records for users who meet specific criteria.

They can be used to grant read-only or read-write access to records owned by other users.

Manual Sharing:

Administrators and record owners can manually share specific records with other users or groups.

User

Niklaus Mikaelson

User ProfileHelp for this Page ?

Permission Set Assignments [0] | Permission Set Assignments: Activation Required [0] |
Permission Set Group Assignments [0] | Permission Set License Assignments [0] | Personal Groups [0] |
Public Group Membership [0] | Queue Membership [0] | Team [0] | Managers in the Role Hierarchy [0] |
OAuth Apps [0] | Third-Party Account Links [0] | Installed Mobile Apps [0] |
Authentication Settings for External Systems [0] | Login History [0+] | User Provisioning Accounts [0]

User Detail

Edit Sharing Reset Password Freeze
View Summary

Name	Niklaus Mikaelson	Role	Manager
Alias	nmika	User License	Salesforce
Email	linkchinmai@gmail.com [Verify] i	Profile	Manager
Username	linkchinmai@gmail.com	Active	<input checked="" type="checkbox"/>
Nickname	User17301889022503304337 i	Marketing User	<input type="checkbox"/>
Title		Offline User	<input type="checkbox"/>
Company		Knowledge User	<input type="checkbox"/>
Department		Flow User	<input type="checkbox"/>
Division		Service Cloud User	<input type="checkbox"/>
Address		Site.com Contributor User	<input type="checkbox"/>
Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)	Site.com Publisher User	<input type="checkbox"/>
Locale	English (India)	WDC User	<input type="checkbox"/>

Public Groups

Help for this Page ?

A public group is a set of users. It can contain individual users, other groups, the users in a particular role or territory, or the users in a role or territory plus all of the users below that role or territory in the hierarchy.

View: All v Edit | Create New View

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other All

New				
Action	Label ↑	Group Name	Created By	Created Date
Edit Del	sales team	sales_team	Kella, Sai	29/10/2024, 1:39 pm

Service records Sharing Rules			
		New	Recalculate
Service records Sharing Rules Help ?			
Action	Criteria	Shared With	Access Level
Edit Del	Owner in <u>Role: sales_person</u>	<u>Role: Manager</u>	Read/Write

Activity 8: Creating a Flow

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

1. Create a New Flow:

- In **Setup**, search for **Flow** > select **New Flow** > choose **Record-triggered flow** and click **Create**.
- Select **Billing Details and Feedback** as the object, trigger on **Created or Updated**, and optimize for **Actions and Related Records** > click **Done**.

2. Add Elements to the Flow:

- **Update Records Element:**
 - Label: "Amount Update"
 - Set conditions where **Payment_Status__c** equals **Completed**.
 - Set **Payment_Paid__c** to reference
{!\$Record.Service_records__r.Appointment__r.Service_Amount__c}.

Click **Done**.

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

Payment_Status__c

Operator

Equals

Value

Completed

+ Add Condition

Set Field Values for the Billing details and feedback Record

Field

Payment_Paid__c

Value

{!\$Record > Service records > Appointment > S...

- **Create a New Resource:**

- Go to **Toolbox > New Resource** > select **Variable** > type: **Text Template**.
- API Name: "alert"; paste a message template to thank the customer for payment.

Edit Text Template

* API Name ⓘ
alert

Description

* Body ⓘ
Insert a resource... 🔍 View as Plain Text ▼

Dear {!\$Record.Service_records__r.Appointment__r.Customer_Details__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 .

Cancel Done

- **Add an Action Element:**

- Select **Send Email** > Label: "Email Alert" > enable body > use **alert** text template as Body.
- Set **RecipientAddressList** to
{!\$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail__c
}.
- Add **Subject**: "Thank You for Your Payment - Garage Management".


Send Email


* Label

Email Alert

* API Name ⓘ

Email_Alert

Description


Send Email ⓘ

emailSimple-emailSimple

Use values from earlier in the flow to set the inputs for the "Send Email" core action. To use its outputs later in the flow, store them in variables.

Set Input Values for the Selected Action

Add Threading Token to Body

Not Included


Add Threading Token to Subject

Not Included

A_B BCC Recipient Address List

Not Included

A_B Body ⓘ


alert

X

Included

3. Save and Activate:

- Give the Flow a label, save it, and click **Activate**

Activity 9: Creating Apex triggers

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

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.

1. Create Apex Class:

- In **Developer Console**, create a new Apex class named **AmountDistributionHandler**.
- Code logic in AmountDistributionHandler assigns a **Service_Amount__c** value to each Appointment__c record based on selected services (e.g., Maintenance, Repairs, Replacement Parts).



```
1 public class AmountDistributionHandler {
2
3     public static void amountDist(list<Appointment__c> listApp){
4
5         list<Service_records__c> serList = new list<Service_records__c>();
6
7         for(Appointment__c app : listApp){
8
9             if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
10
11                 app.Service_Amount__c = 10000;
12
13             }
14
15             else if(app.Maintenance_service__c == true && app.Repairs__c == true){
16
17                 app.Service_Amount__c = 5000;
18
19             }
20
21             else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
22
23                 app.Service_Amount__c = 8000;
24
25             }
26
27             else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
28
29                 app.Service_Amount__c = 7000;
30
31             }
32
33             else if(app.Maintenance_service__c == true){
34
35                 app.Service_Amount__c = 2000;
36
37             }
38
39             else if(app.Repairs__c == true){
40
41                 app.Service_Amount__c = 3000;
42
43             }
44
45             else if(app.Replacement_Parts__c == true){
46
47                 app.Service_Amount__c = 5000;
48
49             }
50
51         }
52     }
```

2. Create Apex Trigger:

- In **Developer Console**, create a trigger named **AmountDistribution** for the **Appointment__c** object.

- The trigger runs **before insert** and **before update** events, calling the AmountDistributionHandler.amountDist method when records exceed a defined threshold.

```

File • Edit • Debug • Test • Workspace • Help • <
AmountDistribution.apex | AmountDistributionHandler.apex
Code Coverage: None | API Version: 62 | Go To

1 trigger AmountDistribution on Appointment__c (before insert, before update) {
2
3     if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
4
5         AmountDistributionHandler.amountDist(trigger.new);
6
7     }
8 }

```

○

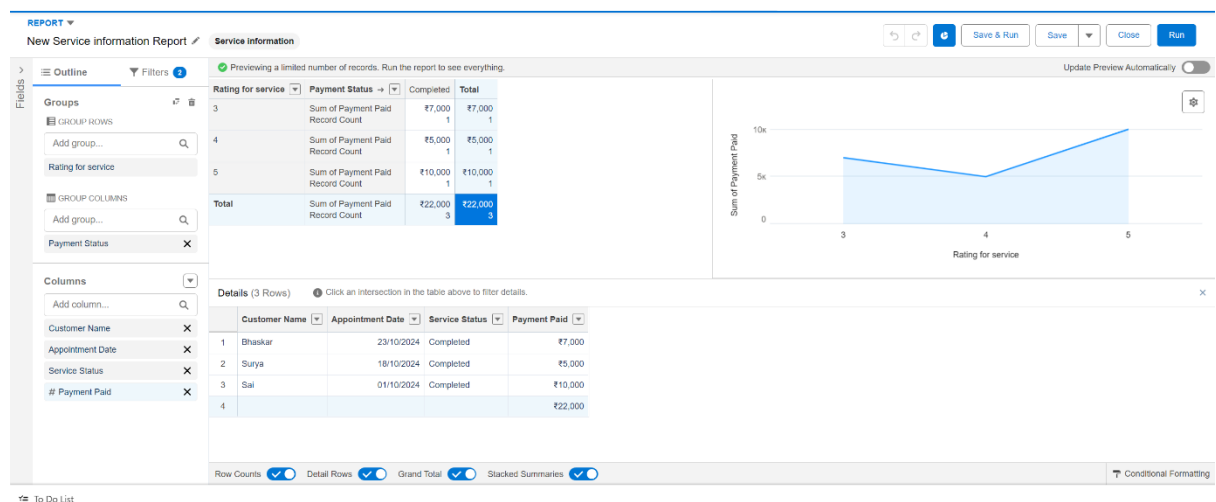
This handler and trigger automate amount distribution based on customer service selections.

Activity 10: Configuring Reports and Dashboard

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

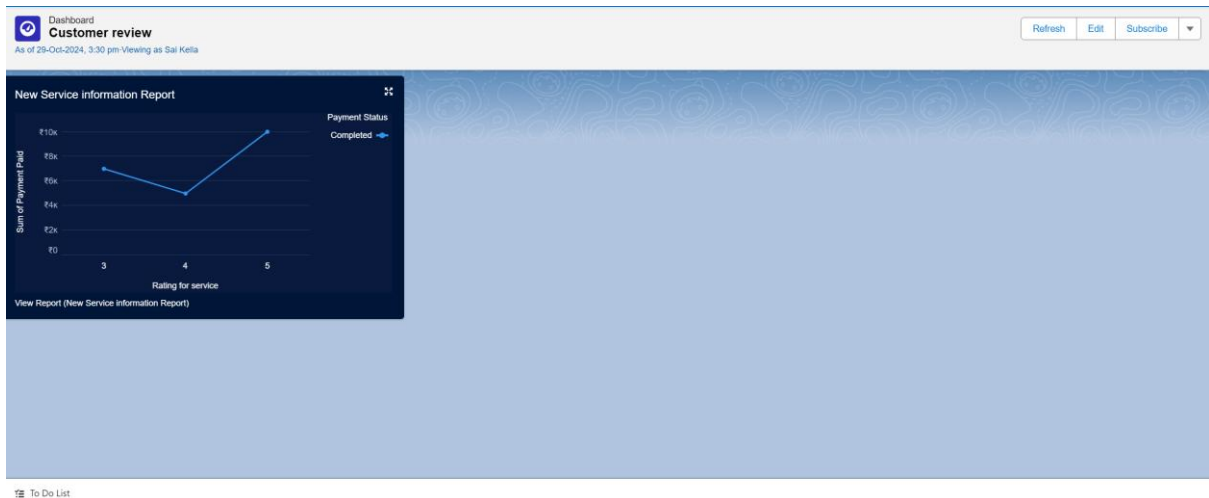
Types of Reports in Salesforce

1. Tabular
2. Summary
3. Matrix
4. Joined Reports



Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you've gathered with reports. Use dashboards to help users identify

trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.



Features in Detail

Appointment

1. Customer requests an appointment, entering their preferred date and time.
2. also provides the type of service required and provide details like vehicle number plate and other customer details.
3. Appointment is confirmed, and a reminder email is sent to the customer.

Service details and Billing

1. Here all services provided, parts replaced and others will be recorded.
2. Billing record is generated based on service type and labor, with costs calculated automatically.
3. Payment reminders are sent if the billing status is pending.

Feedback Collection and Analysis

1. Feedback request is sent to the customer post-service.
2. Customer submits their feedback, which is recorded and linked to their profile.
3. Negative feedback triggers a follow-up case for management review.

System Requirements

- Salesforce Enterprise Edition or higher
- System Administrator profile for initial setup
- Service records and sales person licenses for end user

Author

Kella Chiranjeevi Chinmai Sai

Gayatri Vidya Parishad College of Engineering (A), Visakhapatnam

Roll Number: 21131A0592

Email: 21131A0592@gvpce.ac.in