Garage Management System

By

Nivas Parupati

nivasreddyparupati@gmail.com

Project Abstract

The Salesforce-based Garage Management System (GMS) is a sophisticated solution tailored for automotive repair facilities to elevate service quality, streamline operations, and cultivate enduring customer relationships. Leveraging Salesforce's powerful CRM capabilities, GMS offers an intuitive interface and comprehensive features, allowing garages to thrive in a competitive marketplace. The system automates critical processes such as appointment scheduling, inventory management, billing, and customer communications. By integrating these functionalities within Salesforce, GMS ensures a seamless and satisfying experience for both customers and staff. This advanced solution empowers automotive repair businesses to deliver top-notch service, optimize workflows, and maintain a competitive edge

INDEX PAGE

INTRODUCTION

Running an automotive repair shop involves juggling many tasks, from scheduling appointments to managing inventory and billing customers. The Salesforce-based Garage Management System (GMS) is designed to make these tasks easier and more efficient.

This system uses Salesforce's powerful tools to help garages deliver better service and keep operations running smoothly. With GMS, repair shops can automate important tasks like booking appointments, tracking inventory, sending bills, and communicating with customers.

The user-friendly design of GMS ensures that both staff and customers have a pleasant experience. By using this system, garages can stay ahead of the competition, provide exent service, and manage their business more effectively. GMS not only simplifies daily operations but also provides valuable insights to help make better business decisions.

TASK 1:

1.1 Creating The Developer Account:

1)To create a developer account in Salesforce, follow these steps:

2)Go to Salesforce Developer Signup.

Fill in the signup form with the following details:

First name & Last name

Email

Role: Developer

Company: College Name

Country: India

Postal Code: Pin code

Username: A combination of your name and company (format: username@organization.com, not an actual email id)

3)Click Sign me up.

1.2 Activating the Account:

1. Check the inbox of the email you used for signup.
2. Click on the verification link to activate your account (the email may take 5-10 minutes to arrive).
3. Click on **Verify Account**.
4. Set a password and answer a security question.
5. Click on **Change Password**.
6. You will be redirected to your Salesforce setup page.

TASK 2:

Creating the Objects:

This report outlines the steps to create various custom objects in Salesforce, specifically for a Garage Management System. The objects include Customer Details, Appointments, Service Records, and Billing Details and Feedback. These objects will help streamline operations and improve data management within the system.

1)Creating Custom Objects

Customer Details Object

Navigation Path: From the setup page, click on Object Manager >> Create >> Custom Object.

Details:

Label Name: Customer Details

Plural Label Name: Customer Details

Record Name: Customer Name

Data Type: Text

Options: Allow reports, Track Field History, Allow search

Save: Click Save to create the object.

2)Appointment Object

Navigation Path: From the setup page, click on Object Manager >> Create >> Custom Object.

Details:

Label Name: Appointment

Plural Label Name: Appointments

Record Name: Appointment Name

Data Type: Auto Number

Display Format: app-{000}

Starting Number: 1

Options: Allow reports, Track Field History, Allow search

Save: Click Save to create the object.

3)Service Records Object

Navigation Path: From the setup page, click on Object Manager >> Create >> Custom Object.

Details:

Label Name: Service Records

Plural Label Name: Service Records

Record Name: Service Records Name

Data Type: Auto Number

Display Format: ser-{000}

Starting Number: 1

Options: Allow reports, Track Field History, Allow search

Save: Click Save to create the object.

4)Billing Details and Feedback Object

Navigation Path: From the setup page, click on Object Manager >> Create >> Custom Object.

Details:

Label Name: Billing Details and Feedback

Plural Label Name: Billing Details and Feedback

Record Name: Billing Details and Feedback Name

Data Type: Auto Number

Display Format: bill-{000}

Starting Number: 1

Options: Allow reports, Track Field History, Allow search

Save: Click Save to create the object.

Task 3:

Creating Tabs

1)Creating a Custom Tab for Customer Details

Navigation Path:

Go to the setup page.

Type "Tabs" in the Quick Find bar.

Click on "Tabs".

Click on "New" under the Custom Object Tabs .

Details:

Select Object: Customer Details

Select Tab Style: Choose a style that represents Customer Details.

Click Next.

Add to Profiles Page: Keep the default settings.

Click Next.

Add to Custom App: Uncheck "Include Tab".

Ensure "Append tab to users' existing personal customizations" is checked.

Click Save.

Creating Remaining Tabs (Appointments, Service Records, Billing Details and Feedback)

Repeat the above steps for each remaining object: Appointments, Service Records, and Billing Details and Feedback.

Navigation Path:

Go to the setup page.

Type "Tabs" in the Quick Find bar.

Click on "Tabs".

Click on "New" under the Custom Object Tabs .

Details for Each Object:

Select Object: Choose the respective object (Appointments, Service Records, Billing Details and Feedback).

Select Tab Style: Choose a sui style for each object.

Click Next.

Add to Profiles Page: Keep the default settings.

Click Next.

Add to Custom App: Uncheck "Include Tab".

Ensure "Append tab to users' existing personal customizations" is checked.

Click Save.

Task 4:

Building Lignting App

Creating a Lightning App in Salesforce for managing garage operations allows for streamlined processes and efficient management of various business aspects. This guide outlines the steps to create the Garage Management Application, including adding essential navigation items and user profiles.

Steps to Create a Lightning App

Accessing App Manager:

Begin by navigating to the setup page. In the Quick Find bar, search for "App Manager" and select it. Click on "New Lightning App" to start the creation process.

App Details:

Enter the name "Garage Management Application" in the app details . Proceed to the next steps, keeping all settings on the App Options and Utility Items pages as default.

Adding Navigation Items:

To enhance navigation within the app, select essential items such as Customer Details, Appointments, Service Records, Billing Details and Feedback, Reports, and Dashboards from the search bar. Move these items using the ar button to include them in the app's navigation menu.

Assigning User Profiles:

To ensure appropriate access and functionality, search for the "System Administrator" profile in the search bar. Add this profile by clicking the ar button. Complete the setup by saving and finishing the configuration.

Task 5:

Creating Fields

1)Customer Details Object Fields

Phone Field:

Created a phone field labeled "Phone Number" to store customer contact numbers.

Email Field:

Added an email field labeled "Gmail" to capture customer email addresses.

2)Lookup Fields

Appointment Object:

Established a lookup relationship field to link appointments to customer details, ensuring appointments are associated with the correct customer.

Service Records Object:

Created a lookup relationship field to link service records to appointments. Added a filter to ensure the appointment date is less than the created date, making it a required field with error validation.

Billing Details and Feedback Object:

Added a lookup relationship field to connect billing details and feedback with service records.

3)Checkbox Fields

Appointment Object:

Added multiple checkbox fields to indicate different services, including "Maintenance Service," "Repairs," and "Replacement Parts," all with default values set to unchecked.

Service Records Object:

Created a checkbox field labeled "Quality Check Status" to track the quality check status of the service records.

4)Date Fields

Appointment Object:

Added a date field labeled "Appointment Date" and marked it as required to ensure each appointment has a specified date.

5)Currency Fields

Appointment Object:

Created a currency field labeled "Service Amount" to record the cost of services provided, with read-only access for all profiles.

*Billing Details and Feedback Object:

Added a currency field labeled "Payment Paid" to track payments made by customers.

6)Text Fields

Appointment Object:

Created a text field labeled "Vehicle Number Plate" with a length of 10 characters, marked as required and unique for vehicle identification.

Billing Details and Feedback Object:

Added a text field labeled "Rating for Service" with a length of 1 character, required and unique to capture customer ratings.

7)Picklist Fields

Service Records Object:

Created a picklist field labeled "Service Status" with values "Started" and "Completed" to track the progress of services.

Billing Details and Feedback Object:

Added a picklist field labeled "Payment Status" with values "Pending" and "Completed" to monitor payment statuses.

8)Formula Fields

Service Records Object:

Created a formula field labeled "Service Date" to automatically populate with the created date of the record.

Task 6:

Validation Rules:

1)Validation Rule for Appointment Object

Rule Name: Vehicle

Purpose: Ensure that the vehicle number plate follows a specific format.

Error Condition Formula:

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

Error Message: "Please enter a valid number"

Error Location: Field - Vehicle number plate

2)Validation Rule for Service Records Object

Rule Name: service_status_note

Purpose: Ensure that the service status is marked as "Completed" before the record can be saved.

Error Condition Formula:

NOT(ISPICKVAL(Service_Status__c, "Completed"))

Error Message: "Still it is pending"

Error Location: Field - Service status

3)Validation Rule for Billing Details and Feedback Object

Rule Name: rating_should_be_less_than_5

Purpose: Ensure that the rating for service is between 1 and 5.

Error Condition Formula:

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

Error Message: "Rating should be from 1 to 5"

Error Location: Field - Rating for Service

Task 7:

Duplicate Rules:

Creating a Matching Rule for Customer Details Object

1. **Rule Name**: Matching Customer Details

    a. **Object**: Customer Details

    b. **Matching Criteria**:

        i. **Gmail**: Exact match

        ii. **Phone Number**: Exact match

    a. **Activation**: Once the rule is saved, it is activated to start matching customer details based on the specified criteria.

Creating a Duplicate Rule for Customer Details Object

1. **Rule Name**: Customer Detail Duplicate

    a. **Object**: Customer Details

    b. **Matching Rule**: Utilizes the previously created "Matching Customer Details" rule.

    c. **Activation**: After saving the duplicate rule, it is activated to begin identifying duplicate customer details based on the matching criteria.

Task 8:

Profiles:

Creating the Manager Profile

1. **Profile Cloning**:
    a. Clone the Standard User profile to create a new profile named "Manager."
2. **Custom App Settings**:
    a. Set the Garage Management Application as the default custom app for the Manager profile.
3. **Custom Object Permissions**:
    a. Grant access permissions for the following objects:
        i. Appointments
        ii. Billing Details and Feedback
        iii. Service Records
        iv. Customer Details
4. **Session Timeout**:
    a. Configure the session timeout to occur after 8 hours of inactivity.
5. **Password Policies**:
    a. Set user passwords to never expire.
    b. Establish a minimum password length of 8 characters.

Creating the Sales Person Profile

1. **Profile Cloning**:
    a. Clone the Salesforce Platform User profile to create a new profile named "Sales Person."
2. **Custom App Settings**:
    a. Set the Garage Management Application as the default custom app for the Sales Person profile.
3. **Custom Object Permissions**:
    a. Grant access permissions for the following objects:
        i. Appointments
        ii. Billing Details and Feedback
        iii. Service Records
        iv. Customer Details

Task 9:

Role And Role Hierarch

Creating the Manager Role

1. **Access Roles Setup**:
    a. Navigate to the roles setup section in Salesforce.
2. **Expand Role Hierarchy**:
    a. Expand all roles and add a new role under the appropriate superior role.
3. **Define Role Details**:
    a. Label the new role as "Manager" and save the configuration.

Creating Sales Person Role

1. **Access Roles Setup**:
    a. Navigate to the roles setup section in Salesforce.
2. **Add Role Under Manager**:
    a. Expand the CEO role and add a new role under the Manager role.
3. **Define Role Details**:
    a. Label the new role as "Sales Person" and save the configuration.

Task 10:

Users:

Creating a Manager User

1. **Access User Setup**:
    a. Navigate to the users section in Salesforce setup.
2. **New User Details**:
    a. Fill in the required fields with the user's details, such as first name, last name, alias, email, username, and nickname.
    b. Assign the role of "Manager."
    c. Set the user license to "Salesforce."
    d. Assign the profile of "Manager."
3. **Save the Configuration**:
    a. Save the user details to create the new Manager user.

Creating Sales Person Users

1. **Access User Setup**:
    a. Navigate to the users section in Salesforce setup.
2. **New User Details**:
    a. Fill in the required fields with the user's details.
    b. Assign the role of "Sales Person."
    c. Set the user license to "Salesforce Platform."
    d. Assign the profile of "Sales Person."
3. **Save and Repeat**:
    a. Save the user details and repeat the process to create at least three users with the Sales Person role and profile.

Task 11:

Public Groups:

Creating the Sales Team Public Group

1. **Access Public Groups Setup**:

    a. Navigate to the public groups section in Salesforce setup.

2. **Define Group Details**:

    a. Provide a label for the new group, such as "Sales Team." The group name is automatically generated based on the label.

3. **Add Members**:

    a. Search for roles in the available members section.

    b. Select the "Sales Person" role and add it to the selected members list.

4. **Save the Group**:

    a. Save the new public group configuration

Task 12:

Sharing Settings

Configuring Sharing Settings for Service Records

1. **Access Sharing Settings**:

    Navigate to the Sharing Settings section in Salesforce setup.

2. **Set Object-Wide Default (OWD)**:

    a. Change the OWD setting for the Service Records object to "Private" to restrict access to records by default.

3. **Create Sharing Rules**:

    a. Define a new sharing rule to grant access to specific roles.

4. **Define Sharing Rule Details**:

    a. Label the sharing rule appropriately, such as "Sharing Setting."

    b. Specify the members of the role to be shared, selecting "Sales Person."

    c. Determine who will receive the access, selecting "Manager."

    d. Set the access level to "Read/Write."

5. **Save and Refresh**:

    a. Save the sharing rule and refresh the settings to apply changes.

Task 13:

Flows:

Creating the Flow

1. **Initiate Flow Creation**:
    a. Access the Flow setup page and start a new Record-Triggered Flow.
2. **Configure Flow Trigger**:
    a. Select "Billing Details and Feedback" as the object.
    b. Set the flow to trigger when a record is created or updated.
    c. Choose "Actions and Related Records" to optimize the flow.
3. **Add Update Records Element**:
    a. Label the update element as “Amount Update.”
    b. Configure it to update records where the field Payment_Status__c equals "Completed."
    c. Set the field value for Payment_Paid__c to the service amount from the related Appointment record.
4. **Create and Configure Email Alert**:
    a. Define a new resource as a text template to construct the email body.
    b. Use rich text format for clarity and include placeholders for dynamic content.
    c. The email body should express gratitude for the payment and include the payment amount.
5. **Add Email Action**:
    a. Insert an "Action" element to send an email.
    b. Utilize the previously created text template for the email body.
    c. Set the recipient address dynamically from the related record.
    d. Define the subject of the email as "Thank You for Your Payment - Garage Management."
6. **Save and Activate the Flow**:
    a. Save the flow with an appropriate label and auto-populated API name.
    b. Activate the flow to make it operational.

Task 14:

Apex Triggers

Apex Handler: AmountDistributionHandler

Purpose: This Apex class calculates and updates the service amount based on selected services in the Appointment__c object.

Class Definition:

1. **Class Name**: AmountDistributionHandler
2. **Method**: amountDist
3. **Parameters**: List<Appointment__c> – A list of appointment records to process.

Functionality:

1. Iterates through the list of appointments.
2. Updates the Service_Amount__c field based on the combination of services selected:
    a. All three services: 10000
    b. Maintenance and Repairs: 5000
    c. Maintenance and Replacement Parts: 8000
    d. Repairs and Replacement Parts: 7000
    e. Maintenance only: 2000
    f. Repairs only: 3000
    g. Replacement Parts only: 5000

Trigger: AmountDistribution

Purpose: This trigger invokes the AmountDistributionHandler to update the Service_Amount__c field whenever an appointment record is inserted or updated.

Trigger Definition:

1. **Trigger Name**: AmountDistribution
2. **sObject**: Appointment__c
3. **Events**: before insert, before update

Trigger Logic:

1. Checks if the trigger event is before insert or before update.
2. Calls the amountDist method from AmountDistributionHandler to perform the required updates on the Appointment__c records.

Code:

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

```
  }
}
```

Task 15:

Reports:

Creating a Report Folder

1. **Objective**: Organize reports by creating a dedicated folder.
2. **Process**:
    a. Access the Reports tab through the app launcher.
    b. Click on **New Folder** and name it **"Garage Management Folder"**. The folder's unique name is automatically generated.
    c. Save the new folder to store and manage related reports.

Sharing the Report Folder

1. **Objective**: Control access to the report folder.
2. **Process**:
    a. Navigate to the Reports tab and locate the **"Garage Management Folder"**.
    b. Click on the dropdown menu next to the folder name and select **Share**.
    c. Choose **Roles** as the sharing option, search for the **"Manager"** role, and assign **View**access.
    d. Confirm the sharing settings and finalize by clicking **Done**.

Creating a Custom Report Type

1. **Objective**: Define a report type to include multiple related objects.
2. **Process**:
    a. Access **Setup** and search for **Report Types**.
    b. Click on **New Custom Report Type**.
    c. Set the **Primary Object** to **Customer Details**.
    d. Name the report type **"Service Information"** and confirm the auto-populated name.
    e. Categorize under **Other Reports** and set the deployment status to **Deployed**.
    f. Click **Next** to define related objects:
        i. **Appointment Object**
        ii. **Service Records**
        iii. **Billing Details and Feedback**
    a. Save the new report type.

Creating a Report

1. **Objective**: Generate a detailed report based on the created report type.
2. **Preparation**:
    a. Ensure to create at least 10 records for each object to ensure comprehensive data representation.

3. **Process**:

    a. Access the Reports tab and click on **New Report**.

    b. Select the category **Other Reports** and choose **Service Information** as the report type.

    c. Start the report builder and configure the fields:

        i. **Columns**: Customer Name, Appointment Date, Service Status, Payment Paid.

        ii. **Group Rows**: Rating for Service, Payment Status.

    a. Add a Line Chart to visualize the data.

    b. Save the report with the name **"New Service Information Report"**, ensuring the unique name is auto-populated and the report is saved in the **"Garage Management Folder"**.

Task 16:

Dashboards:

Creating a Dashboard Folder

1. **Objective**: Organize dashboards by creating a specific folder.
2. **Process**:
    a. Navigate to the **Dashboards** tab via the app launcher.
    b. Click **New Folder** and name it **"Service Rating Dashboard"**. The unique name is auto-generated.
    c. Save the folder to group and manage related dashboards efficiently.

Sharing the Dashboard Folder

1. **Objective**: Set access permissions for the dashboard folder.
2. **Process**:
    a. Locate the **"Service Rating Dashboard"** folder in the Dashboards tab.
    b. Click on the dropdown menu for the folder and select **Share**.
    c. Assign appropriate permissions based on user roles, ensuring that the relevant team members have access to view or edit the dashboards as needed.
    d. Confirm the sharing settings to finalize.

Creating a Dashboard

1. **Objective**: Build a dashboard to visualize data and insights.
2. **Process**:
    a. Access the **Dashboards** tab and click **New Dashboard**.
    b. Enter a name for the dashboard and select the **"Service Rating Dashboard"** folder created earlier.
    c. Click **Create** to start building the dashboard.
    d. Add a new component by selecting **Add Component**.
    e. Choose a report to base the dashboard on, and select a **Line Chart** for visualization.
    f. Customize the chart theme as desired.
    g. Click **Add** to include the component in the dashboard, then click **Save** and **Done**.

Subscribing to the Dashboard

1. **Objective**: Automate the delivery of dashboard updates.
2. **Process**:
    a. After saving the dashboard, click **Subscribe** at the top right.
    b. Set the subscription frequency to **Weekly**.
    c. Choose **Monday** as the day for the report to be sent.

d. Click **Save** to activate the subscription and ensure regular updates.