

# A Garage Management System (GMS) Application

## 1. Project Overview

This project aims to streamline and automate the operations of automotive repair facilities through a Garage Management System (GMS). It is designed to address inefficiencies in managing customer appointments, repair tasks, inventory, and billing, delivering a scalable and user-friendly solution. GMS empowers garages to enhance operational efficiency, deliver top-notch service, and build lasting customer relationships, aligning with long-term objectives to thrive in a competitive market while ensuring a seamless and satisfying experience for both customers and staff.

## 2. Objectives

Business Goals:

- Increase operational efficiency in garage management.
- Enhance customer satisfaction through a streamlined repair process.
- Ensure accurate tracking of inventory, repair history, and billing.

Specific Outcomes:

- Centralized database for customer details, repair history, and inventory management.
- Automated workflows for appointment scheduling, task allocation, and reminders.
- Real-time dashboards to monitor garage performance and inventory usage.

## 3. Salesforce Key Features and Concepts Utilized

- **Customer Management:** Efficiently handle customer records, service history, and communication.
- **Appointment Scheduling:** Automated booking system with reminders.
- **Billing and Invoicing:** Simplified and accurate billing processes.
- **Reports and Dashboards:** Insights into operational metrics, inventory trends, and customer satisfaction.

## 4. Detailed Steps to Solution Design

- **Data Models:** Define custom objects for customers details , appointments, and service records and billing and feedback . Establish relationships between these objects to ensure data integrity.Establish relationships between these entities to maintain data integrity.
- **User Interface:** Create intuitive layouts and pages for managing appointments, repairs, and dashboards. Include visual aids for seamless navigation.
- **Business Logic:** Implement workflows and automation to handle repair task assignments, overdue notifications, and inventory updates efficiently.

## 5. Testing and Validation

- **Unit Testing:** Validate the system's logic and automated processes to ensure they meet business requirements.
- **User Interface Testing:** Test all pages and features for seamless functionality across devices.

## 6. Key Scenarios Addressed by Salesforce in the Implementation Project

- **Customer Service:**
  - Centralized customer details for seamless management of contact information, vehicle history, and preferences.
  - Automated appointment scheduling linked with service records, ensuring smooth communication and task allocation.
  - Integrated feedback collection to measure customer satisfaction and address concerns promptly.
- **Appointments:**
  - Streamlined scheduling process with automated reminders sent to customers and staff.
  - Clear linkage between appointments and service records, enabling efficient planning and resource allocation.

- **Service Records:**
  - Comprehensive tracking of past and ongoing repair tasks, ensuring transparency and accountability.
  - Easy access to vehicle history for staff, allowing better diagnostics and service recommendations.
- **Inventory Updates:**
  - Real-time management of parts and tools usage linked to service records, reducing downtime and ensuring availability.
  - Alerts for low stock, enabling timely reordering and avoiding delays.
- **Billing:**
  - Simplified and accurate billing process directly linked to service records and inventory usage.
  - Clear and detailed invoices generated automatically, enhancing customer trust and reducing errors.
- **Feedback Management:**
  - Feedback collection integrated into the system, allowing customers to provide input post-service.
  - Reports generated from feedback data to identify areas for improvement and boost overall service quality.
- **Data Accuracy:**
  - Robust validation processes across customer details, service records, billing, and feedback, ensuring reliable and error-free information.
  - Centralized database structure to eliminate data redundancy and discrepancies.

## 7. Conclusion

**Summary of Achievements:** The Garage Management System successfully automates the management of automotive repair facilities, offering centralized data management, automated workflows, and enhanced customer communication. These features collectively contribute to improved operational efficiency and customer satisfaction, laying the foundation for scalable growth in the automotive service industry.

Labels

Looking for?

## Custom Tabs

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
<a href="#">Edit</a>   <a href="#">Del</a>	Appointments	Bell	
<a href="#">Edit</a>   <a href="#">Del</a>	Billing details and feedback	Cell phone	
<a href="#">Edit</a>   <a href="#">Del</a>	Customer Details	Books	
<a href="#">Edit</a>   <a href="#">Del</a>	Service records	Bottle	

Web Tabs

[New](#) [What Is This?](#)

No Web Tabs have been defined

Visualforce Tabs

[New](#) [What Is This?](#)

No Visualforce Tabs have been defined

## App Launcher

[Visit AppExchange](#)

profiles and feeds

Salesforce CRM Content

multiple records on one screen

**Service Console**  
(Lightning Experience) Lets support agents work with multiple records across customer service...

**Sales**  
Manage your sales process with accounts, leads, opportunities, and more

**Lightning Usage App**  
View Adoption and Usage Metrics for Lightning Experience

**Digital Experiences**  
Manage content and media for all of your sites.

**Business Rules Engine**  
Create and maintain business rules that perform complex lookups and calculations.

**Salesforce Scheduler Setup**  
Set up personalized appointment scheduling.

**Bolt Solutions**  
Discover and manage business solutions designed for your industry.

**Automation**  
Automate business processes and repetitive tasks.

**Garage Management Application**

[All Items](#)

[Accounts](#)
[Communication Subscription Channel T...](#)
[Expression Set Templates](#)
[Messaging Sessions](#)
[Refunds](#)

Flow Builder Update Service Status - V1

Last saved on 3/1/2025, 02:01 pm **Active** Run Debug View Tests Save As New Version Save Deactivate

Select Elements

Record-Triggered Flow Start

Run Immediately

Update Service Update Records

End

**Update Records**

\*Label Update Service \*API Name Update\_Service

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 Quality\_Check\_Status\_\_c Operator Equals Value True

+ Add Condition

Set Field Values for the Service records Record

```

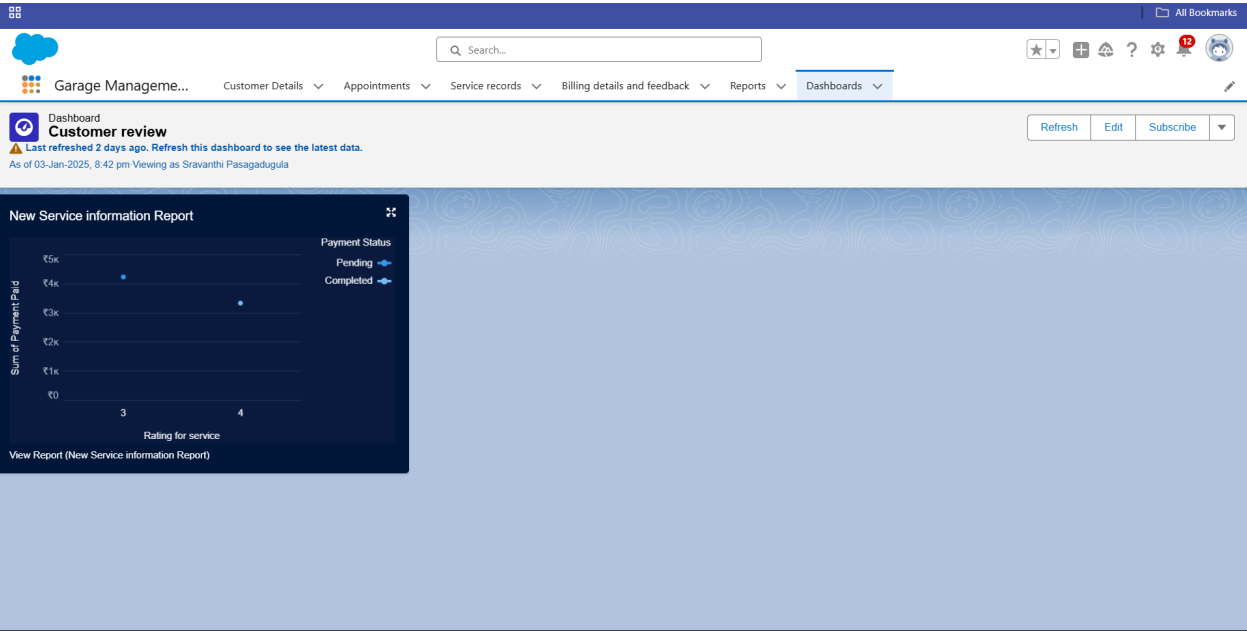
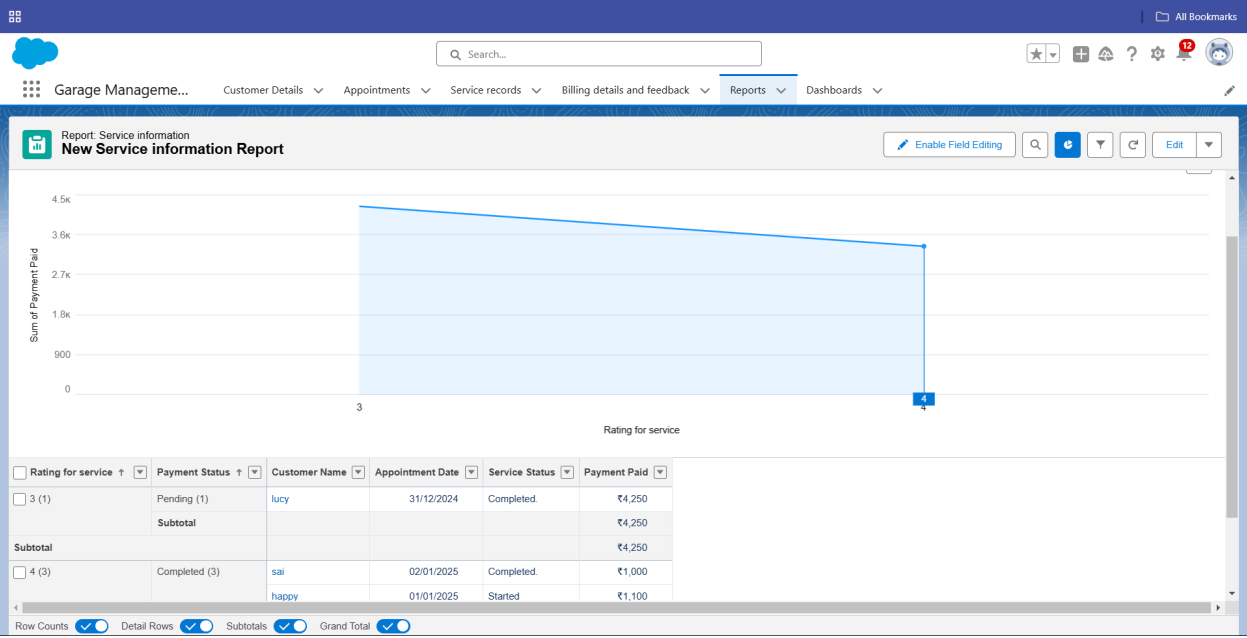
File Edit Debug Test Workspace Help < >
AmountDistributionHandler.apxc
Code Coverage: None API Version: 62 Go To
1 public class AmountDistributionHandler {
2
3
4
5 public static void amountDist(list<Appointment__c> listApp){
6
7     list<Service_records__c> serList = new list<Service_records__c>();
8
9
10
11 for(Appointment__c app : listApp){
12
13     if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
14
15         app.Service_Amount__c = 10000;
16
17     }
18
19     else if(app.Maintenance_service__c == true && app.Repairs__c == true){

```

```

File Edit Debug Test Workspace Help < >
AmountDistributionHandler.apxc AmountDistribution.apxt
Code Coverage: None API Version: 62 G
1 trigger AmountDistribution on Appointment__c (before insert, before update) {
2
3
4
5 if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
6
7     AmountDistributionHandler.amountDist(trigger.new);
8
9
10
11 }
12
13
14 }

```



☰

☁

🔍 Search...

★

+

🏠

?

⚙️

🔔 12

👤

📄 All Bookmarks

☰

Garage Manageme...

Customer Details ▾

Appointments ▾

Service records ▾

Billing details and feedback ▾

Reports ▾

Dashboards ▾

✎

📄

Customer Details

Recently Viewed ▾

👤

New

Import

Change Owner

Assign Label

🔍 Search this list...

⚙️ ▾

📄 ▾

🔄

✎

🗑️

⏏️

16 items • Updated a few seconds ago

	<input type="checkbox"/> Customer Name	
1	<input type="checkbox"/> saddy	▾
2	<input type="checkbox"/> happy	▾
3	<input type="checkbox"/> lucy	▾
4	<input type="checkbox"/> park	▾
5	<input type="checkbox"/> sai	▾
6	<input type="checkbox"/> dog	▾
7	<input type="checkbox"/> micky	▾
8	<input type="checkbox"/> chinnu	▾
9	<input type="checkbox"/> zoo	▾
10	<input type="checkbox"/> durga	▾
11	<input type="checkbox"/> satya	▾
12	<input type="checkbox"/> sravas	▾
13	<input type="checkbox"/> seeta	▾
14	<input type="checkbox"/> shiva	▾
15	<input type="checkbox"/> raju	▾
16	<input type="checkbox"/> ram	▾