**Garage Management System**

## Project Overview

This project is focused on the development of a Garage Management System (GMS), designed to address the primary challenge of inefficient management of customer data, service scheduling, and vehicle repair tracking in auto repair shops.

The system includes features for managing job scheduling, inventory, and employee productivity. Mechanics and technicians can view their assigned tasks, update job statuses, and keep track of parts used, while managers monitor inventory levels and replenish stock as needed. This improves workflow by reducing wait times for parts and minimizing delays in repairs. Integrated billing and invoicing features further streamline operations by generating estimates and final invoices automatically, improving cash flow and ensuring customers receive transparent, itemized bills.

The goal is to deliver a comprehensive solution by leveraging Salesforce’s CRM platform, custom objects, process automation, and mobile integration. Through this project, we aim to enhance operational efficiency, customer satisfaction, and data accuracy and support the long-term goals of optimizing garage operations, improving customer service, and scaling business growth in the automotive repair industry.

**Project Goals:**

* ****Strengthen Data Security and Privacy**:**

Protect sensitive customer, vehicle, and financial information by

implementing robust security measures, including role-based access, encryption, and data backups.

* **Optimize Resource Allocation and Staff Productivity**:

Allocate jobs efficiently based on mechanic availability, skills, and workload to ensure balanced workloads and high productivity.

* **Enhance Operational Efficiency**:

Automate and streamline core tasks, such as appointment scheduling, inventory tracking, and invoicing, to reduce manual work, improve accuracy, and optimize daily operations.

* **Improve Customer Experience**:

Provide a seamless experience for customers by offering timely updates, clear communication, and easy access to information about their vehicle’s status.

# Objectives

The **objective of a Garage Management System project** is to create a comprehensive software platform that automates and simplifies all aspects of garage operations, from customer scheduling and service tracking to inventory management and invoicing. By centralizing these functions in a single, easy-to-use system, the project aims to improve efficiency, reduce errors, and enhance customer service.

**Business Goals:**

* **Improve Operational Efficiency:**

Streamline the management of customer data, service scheduling, and vehicle repair tracking to reduce administrative overhead.

* **Enhance Customer Satisfaction:**

Provide a seamless, user-friendly experience for customers, improving

communication and reducing service delays.

* **Increase Business Scalability:**

Build a system that can scale as the business grows, allowing easy expansion to multiple locations.

**Specific Outcomes:**

* **Centralized Customer Data Management:**

A unified system to store and access detailed customer and vehicle information.

* **Automated Service Scheduling & Reminders:**

Customers can book appointments easily, and automated reminders will reduce missed services.

* **Real-Time Repair Tracking:**

Technicians and service managers can track repair progress, parts usage, and job assignments in real-time.

* **Mobile Access for Technicians:**

A mobile app for technicians to update service statuses and communicate with service managers, improving efficiency.

* **Insightful Reporting & Analytics:**

Real-time reports on service performance, technician productivity, and customer satisfaction.

# Salesforce Key Features and Concepts Utilized

In the Garage Management System (GMS) project, we use several key features and concepts of Salesforce to make the garage operations more efficient, organized, and customer-friendly. Here’s a simplified explanation of the main functionalities:

**Custom Objects**

* **Custom Objects and Fields:**

Custom objects allow us to create specific data structures for things like Customer Profiles, Vehicles, Service Records, and Appointments. These objects store all the important details about each customer, vehicle, service history, and scheduled repairs.

* **Process Automation (Process Builder & Flows):**

Process Builder automates repetitive tasks like sending appointment reminders, service notifications, and follow-up emails to customers, saving time and reducing the chances of human error.

Flows help automate more complex tasks. For example, when a service is completed, a flow could automatically update the system, notify the customer, and trigger a feedback request.

* **Salesforce Lightning Experience:**

Lightning Experience is the modern user interface for Salesforce. It makes it easier for staff to access and manage customer and service data. The clean, intuitive design helps technicians and managers find the information they need quickly.

Custom pages are created in Lightning to show important data, like service history or vehicle status, in a simple and organized way.

* **Salesforce Mobile App:**

The Salesforce mobile app allows technicians to access customer details, update service progress, and log information directly from their smartphones or tablets. This helps them stay connected and manage repairs while on the go.

* **Reports and Dashboards:**

Reports in Salesforce help track key metrics, such as how quickly services are completed, how happy customers are, or how much inventory is used.

Dashboards provide visual summaries of this data, so managers can easily see how the garage is performing and identify areas for improvement.

* **Workflow Rules:**

Workflow rules automatically perform tasks based on certain conditions. For example, a workflow could send a reminder email when a service appointment is approaching, or update the service status once a technician marks a job as done. This helps keep operations running smoothly without manual effort.

* **Chatter:**

Chatter is a tool for communication within Salesforce. It allows the garage team (like technicians and service managers) to chat in real time about job progress, share updates, or ask questions, keeping everyone on the same page.

* **Email and Calendar Integration:**

Salesforce can connect with email and calendar systems, so appointment confirmations and service reminders are automatically sent to customers, and scheduling is synced up with their calendar.

* **Security and Permissions:**

Salesforce ensures data security by giving different users (like technicians, managers, or customer service reps) specific permissions to access only the information they need. This helps protect sensitive customer data and ensures that team members can work effectively without compromising security.

# 4 . Detailed Steps to Solution Design:

# To build the Garage Management System (GMS) using Salesforce, we will design the solution in a structured way. The design process includes defining the data models, crafting the user interface (UI), developing business logic for automation, and ensuring the system is scalable and efficient. Below is a breakdown of the key steps involved in designing the solution.

# **Data Model Design:**

# The data model is the foundation of the system, determining how data will be stored, organized, and accessed in Salesforce. We will create custom objects and fields to track key information about customers, vehicles, services, and appointments.

# **Key Data Entities:**

# Customer Profile: Stores customer contact details and their vehicle information.

# Fields: Name, Email, Phone Number, Vehicle Make/Model, Vehicle VIN, Service History.

# Service Record: Captures repair details, such as service type, technician, and parts used.

# Fields: Service Type (Oil Change, Tire Repair), Service Date, Status (In Progress, Completed), Technician Assigned, Parts Used.

# Appointments: Manages service appointments for customers.

# Fields: Appointment Date, Time, Service Type, Customer, Status.

# Technician Profile: Tracks technician information and availability.

# Fields: Name, Skills, Work Orders Assigned, Availability.

# Relationships:

# Customer Profile → Appointments (1-to-many)

# Service Record → Appointments (1-to-1)

# Service Record → Technician Profile (many-to-1)

# **Action Steps:**

# 1.Create Custom Objects for Customer Profile, Service Record, Appointment, and Technician Profile.

# Define Custom Fields for each object.

# Set up Object Relationships in Salesforce (Lookups, Master-Detail).

# 2. User Interface (UI) Design

# The user interface (UI) should be intuitive and easy to use for both staff (technicians, managers) and customers. Using Salesforce Lightning, we can create a modern and responsive interface for various devices.

# **UI Components:**

# Customer Profile Page: Displays customer contact info, vehicle details, and service history.

# Appointment Scheduling Page: Allows customers and managers to schedule service appointments.

# Service Record Page: Shows detailed information about the service, including status, technician, and parts used.

# Technician Mobile Interface: A mobile app layout for technicians to view and update their assigned tasks and service records.

# Action Steps:

# Design Page Layouts for each custom object (Customer Profile, Service Record, Appointment).

# Use Salesforce Lightning App Builder to create custom pages for technicians and service managers.

# Design Mobile-Friendly Pages for technicians to ensure seamless access via the Salesforce Mobile App.

# 3. Business Logic and Automation

# Automation is crucial for improving operational efficiency and reducing manual effort. We will use Salesforce’s Process Builder, Flows, and Workflow Rules to automate common tasks and business processes.

# **Key Reports:**

# Service Performance Report: Tracks service turnaround times, technician productivity, and repair status.

# Customer Satisfaction Report: Collects and visualizes customer feedback after each service.

# Parts Usage Report: Displays the inventory of parts used for repairs.

# Dashboards:

# Visual dashboards to monitor key business metrics, such as service completion rates, technician performance, and customer feedback scores.

# Action Steps:

# Create Reports to track critical metrics like service times, technician work, and parts usage.

# Build Dashboards for a real-time, visual overview of garage performance.

# **Security and Permissions:**

# Security is crucial to protect sensitive customer and business data. Salesforce provides robust security features to control who can access and modify data.

# Action Steps:Define User Profiles and assign specific permissions for each role.Set up Field-Level Security to restrict access to sensitive data.

# Implement Sharing Rules to ensure proper data visibility based on user roles.

# 5 . Testing and Validation

The testing phase included rigorous unit testing, interface testing, and integration testing to ensure the solution meets business requirements.

**5.1 Unit Testing**

**Apex Trigger**

An **Apex Trigger** in Salesforce is a piece of code that executes before or after specific data manipulation events (like insert, update, delete) on Salesforce records. Apex Triggers enable you to perform custom actions or automate processes when data changes occur.

EXAMPLE CODE:

trigger AmountDistribution on Appointment\_\_c (before insert, before update) {

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

        AmountDistributionHandler.amountDist(trigger.new);

    }

}

**Apex Classes:**

An **Apex Class** in Salesforce is a template or blueprint that contains variables and methods (functions) to define the behavior and state of an object. Apex Classes allow developers to implement complex business logic, create reusable code, and integrate with other systems within the Salesforce environment.

EXAMPLE CODE:

public class AmountDistributionHandler {

    public static void amountDist(list<Appointment\_\_c> listApp){

        list<Service\_records\_\_c> serList = new list <Service\_records\_\_c>();

        for(Appointment\_\_c app : listApp){

            if(app.Maintenance\_service\_\_c == true && app.Repairs\_\_c == true && app.Replacement\_Parts\_\_c == true){

                app.Service\_Amount\_\_c = 10000;

            }

            else if(app.Maintenance\_service\_\_c == true && app.Repairs\_\_c == true){

                app.Service\_Amount\_\_c = 5000;

            }

            else if(app.Maintenance\_service\_\_c == true && app.Replacement\_Parts\_\_c == true){

                app.Service\_Amount\_\_c = 8000;

            }

            else if(app.Repairs\_\_c == true && app.Replacement\_Parts\_\_c == true){

                app.Service\_Amount\_\_c = 7000;

            }

            else if(app.Maintenance\_service\_\_c == true){

                app.Service\_Amount\_\_c = 2000;

            }

 else if(app.Repairs\_\_c == true){

                app.Service\_Amount\_\_c = 3000;

            }

            else if(app.Replacement\_Parts\_\_c == true){

                app.Service\_Amount\_\_c = 5000;

            }

    }

    }

}

**5.2 User Interface Testing:**

* For the Garage Management System project, User Interface (UI) Testing would focus on ensuring that the system’s interface is intuitive, visually appealing, and user-friendly for both garage staff and customers.
* The key testing areas would include verifying that appointment booking forms, service tracking, and customer dashboards display correctly on different devices. It would also check if buttons, filters, and search options function properly, providing a smooth user experience. Additionally, UI Testing would ensure that the design aligns with the business’s branding, is accessible to users with disabilities, and maintains consistency across pages and features.
* The goal is to ensure a seamless experience for users when interacting with the garage management system’s front-end components.

**5.3 Integration Testing**

Data Flow Testing:

**Data Flow Testing** for the **Garage Management System** project focuses on verifying how data flows through the system, ensuring that inputs, processes, and outputs are handled correctly and efficiently. It ensures that data is properly stored, accessed, and updated throughout the application's componen

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

* **Customer and Vehicle Management:**

Scenario: Storing and managing customer profiles and their vehicles.

How Salesforce Helps: Salesforce stores detailed customer information (e.g., contact details, service history) and tracks each vehicle’s specific data (make, model, VIN). This helps the garage easily access customer and vehicle information when needed.

* **Service Appointment Scheduling**

Scenario: Booking, managing, and tracking service appointments for customers.

How Salesforce Helps: Salesforce allows customers to book appointments online or via service reps, automatically checking for technician availability and scheduling the service. The system sends reminders and updates to both customers and technicians.

* **Service Record Tracking**

Scenario: Tracking the status and progress of ongoing repairs.

How Salesforce Helps: Service records are created for each repair job, showing detailed information such as service type, parts used, technician assigned, and status (e.g., In Progress, Completed). This allows for easy tracking and updates.

# 7 . Conclusion:

# The Garage Management System (GMS) project has successfully improved the garage’s operations by centralizing customer and vehicle information, automating appointment scheduling, and efficiently tracking service progress. Salesforce has streamlined technician assignments, inventory management, and communication through automated alerts, reducing manual work and increasing efficiency.

# The system provides real-time reports and insights, and mobile access allows technicians to manage tasks on the go. With robust security in place to protect data, the system has enhanced operational efficiency, customer satisfaction, and decision-making, enabling the garage to scale and operate more smoothly.