

Garage Management System

by

Ayush Kumar

ayushkumar68824@gmail.com

College: VIT Bhopal

Index Page

S.No.	Topics	Pages No.
1.	Project Overview	3
2.	Objectives	3
3.	Salesforce Key Features and Concepts Utilized	3
4.	Detailed Steps to Solution Design	4
4.1	Step 1: Creating the Developer Account	4
4.2	Step 2: Creating Custom Objects	5
4.3	Step 3: Custom Tabs	6
4.4	Step 4: Building the Lightning App	7
4.5	Step 5: Creating Custom Fields	8-10
4.6	Step 6: Validation Rules	11-12
4.7	Step 7: Duplicate Rules	13-14
4.8	Step 8: Profiles	15
4.9	Step 9: Roles and Role Hierarchy	16
4.10	Step 10: Users	17
4.11	Step 11: Public Groups	18
4.12	Step 12: Sharing Settings	19
4.13	Step 13: Flows	20-21
4.14	Step 14: Apex Triggers	22-23
4.15	Step 15: Reports	24-25
4.16	Step 16: Dashboards	26
4.17	User Adoption	27
5.	Testing and Validation	28
6.	Key Scenarios Addressed by Salesforce in the Implementation Project	28
7.	Conclusion	29

1. Project Overview

This project is focused on developing a Garage Management System using Salesforce. The primary objective is to streamline garage operations, enhance service efficiency, and improve customer experience. The solution aims to digitize and automate tasks related to appointment scheduling, service tracking, billing, and customer feedback. By leveraging Salesforce's features, the project intends to boost operational efficiency, ensure accurate data management, and support the long-term goals of garage businesses.

2. Objectives

Business Goals:

- Enhance customer satisfaction by providing a seamless experience.
- Improve operational efficiency by automating scheduling, service tracking, and payment processes.
- Centralize data for better decision-making and reporting.

Specific Outcomes:

- Implement a digital solution for managing customer details, appointments, service records, and billing.
- Provide real-time insights through reports and dashboards.
- Automate workflows to reduce manual effort and errors.

3. Salesforce Key Features and Concepts Utilized

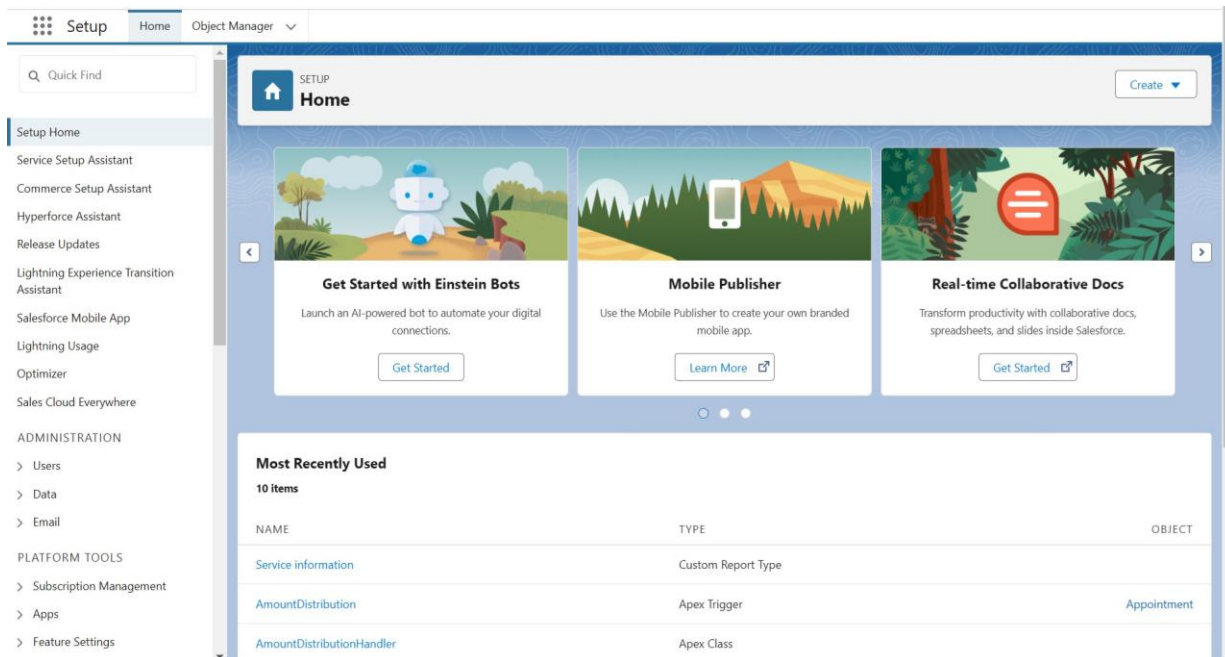
- **Custom Objects:** Customer Details, Appointments, Service Records, Billing Details & Feedback.
- **Custom Tabs:** User-friendly navigation for accessing key objects.
- **Lightning App:** Centralized access to essential tools for the garage management process.
- **Custom Fields:** Multiple field types like lookup, picklist, checkbox, and formula fields to ensure robust data capture.
- **Validation Rules:** Enforced data integrity and error prevention.
- **Profiles & Roles:** Role-based access control for managers and salespersons.
- **Public Groups:** Simplified team-based access to records.
- **Flows:** Automation of record updates and email alerts.
- **Apex Triggers:** Automated calculation of service amounts.
- **Reports & Dashboards:** Data visualization and performance tracking.

4. Detailed Steps to Solution Design

Step 1: Creating the Developer Account

1. **Sign Up** at Salesforce Developer Signup.
2. **Fill in details:** Name, email, role as 'Developer', and company as 'College Name'.
3. **Username** format: username@organization.com.
4. **Activate the account** through the email link.

Your Setup page would look like this:

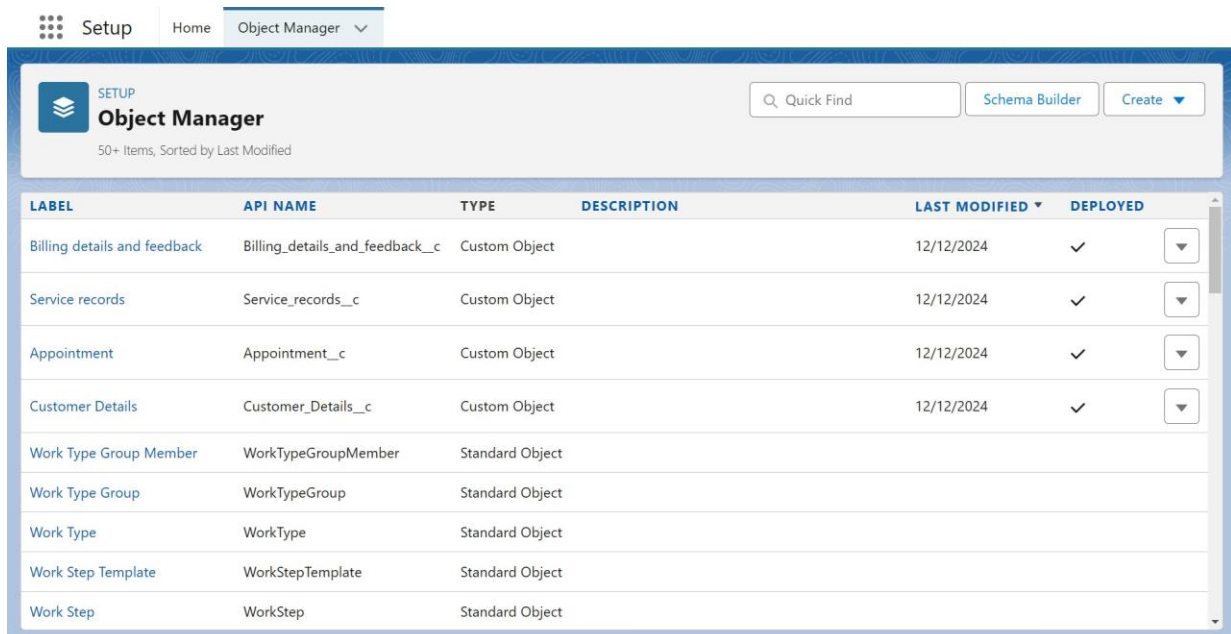


Step 2: Creating Custom Objects

- To Navigate to Setup page: Click on gear icon ? click setup.
- To create an object: From the setup page > Click on Object Manager > Click on Create > Click on Custom Object.
- On Custom object defining page: Enter the label name, plural label name, click on Allow reports, Allow search.
- Click on Save.

For our project we need to create four objects

- **Customer Details:** Text-based customer names with search and report options.
- **Appointment:** Auto-numbering system with fields like appointment name and display format.
- **Service Records:** Auto-numbered service records with fields for tracking service status.
- **Billing Details & Feedback:** Auto-numbered billing records with fields for payment tracking and feedback.



The screenshot shows the Salesforce Object Manager interface. At the top, there are tabs for 'Setup', 'Home', and 'Object Manager'. Below the tabs, the 'Object Manager' header includes a 'Quick Find' search bar, a 'Schema Builder' button, and a 'Create' dropdown menu. The main content area displays a table of objects, sorted by 'Last Modified'.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Billing details and feedback	Billing_details_and_feedback__c	Custom Object		12/12/2024	✓
Service records	Service_records__c	Custom Object		12/12/2024	✓
Appointment	Appointment__c	Custom Object		12/12/2024	✓
Customer Details	Customer_Details__c	Custom Object		12/12/2024	✓
Work Type Group Member	WorkTypeGroupMember	Standard Object			
Work Type Group	WorkTypeGroup	Standard Object			
Work Type	WorkType	Standard Object			
Work Step Template	WorkStepTemplate	Standard Object			
Work Step	WorkStep	Standard Object			

Step 3: Custom Tabs

- Custom tabs for **Customer Details, Appointments, Service Records, and Billing Details & Feedback.**
- Tabs are displayed for easy navigation and user access.

I followed these 4 steps to create all required custom tab

- (i) Go to setup page >> type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab).
- (ii) Select Object(Customer Details) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .
- (iii) Make sure that the Append tab to users' existing personal customizations is checked.
- (iv) Click save.

The screenshot shows the Salesforce Setup interface. The left sidebar has a search bar with 'tabs' entered and a 'User Interface' section with 'Rename Tabs and Labels' and 'Tabs' links. The main content area is titled 'Custom Tabs' and includes a 'Help for this Page' link. Below the title, there is a paragraph explaining that custom tabs can be created to extend Salesforce functionality. The page is divided into three sections: 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Custom Object Tabs' section contains a table with four rows of tabs: 'Appointments' (Computer style), 'Billing details and feedback' (Cell phone style), 'Customer Details' (People style), and 'Service records' (Desk style). The 'Web Tabs' and 'Visualforce Tabs' sections both show 'No Web Tabs have been defined' and 'No Visualforce Tabs have been defined' respectively.

Action	Label	Tab Style	Description
Edit Del	Appointments	Computer	
Edit Del	Billing details and feedback	Cell phone	
Edit Del	Customer Details	People	
Edit Del	Service records	Desk	

Step 4: Building the Lightning App

1. **Name:** Garage Management Application.
2. **Navigation Items:** Customer Details, Appointments, Service Records, Billing, Reports, and Dashboards.
3. **User Profiles:** Assign access to the System Administrator role.

To create a lightning app page:

1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.
2. Fill the app name in app details as Garage Management Application >> Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.
3. To Add Navigation Items
4. Select the items (Customer Details, Appointments, Service records, Billing details and feedback, Reports and Dashboards) from the search bar and move it using the arrow button >> Next.
5. To Add User Profiles: Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.

Setup Home Object Manager

app

Salesforce Mobile App

Data

Mass Transfer Approval Requests

Apps

App Manager

AppExchange Marketplace

Connected Apps

Connected Apps OAuth Usage

Manage Connected Apps

External Client Apps

External Client App Manager

OAuth Usage

Settings

Lightning Bolt

Flow Category

Lightning Bolt Solutions

SETUP Lightning Experience App Manager

New Lightning App New Connected App

24 items • Sorted by Last Modified Date • Filtered by All appmenuitems - TabSet Type, App Type

	App Name	Developer Name	Description	Last Modified	App Type	Visibility
1	Garage Management System	Garage_Management_Syste...	The Garage Management System is ...	12/12/2024, 1:03 pm	Lightning	✓
2	Automation	FlowApp	Automate business processes and re...	25/11/2024, 6:11 pm	Lightning	✓
3	Bolt Solutions	LightningBolt	Discover and manage business soluti...	25/11/2024, 6:10 pm	Lightning	✓
4	Salesforce Scheduler Setup	LightningScheduler	Set up personalized appointment sc...	25/11/2024, 6:10 pm	Lightning	✓
5	Queue Management	QueueManagement	Create and manage queues for your ...	25/11/2024, 6:06 pm	Lightning	✓
6	All Tabs	AllTabSet		25/11/2024, 6:06 pm	Classic	
7	Subscription Management	RevenueCloudConsole	Get started automating your revenu...	25/11/2024, 6:06 pm	Lightning	✓
8	Data Manager	DataManager	Use Data Manager to view limits, mo...	25/11/2024, 6:06 pm	Lightning	✓
9	Digital Experiences	SalesforceCMS	Manage content and media for all of...	25/11/2024, 6:06 pm	Lightning	✓
10	Platform	Platform	The fundamental Lightning Platform	25/11/2024, 6:06 pm	Classic	
11	Sales	Sales	The world's most popular sales force...	25/11/2024, 6:06 pm	Classic	
12	Service	Service	Manage customer service with accou...	25/11/2024, 6:06 pm	Classic	✓
13	Marketing CRM Classic	Marketing	Track sales and marketing efforts wit...	25/11/2024, 6:06 pm	Classic	✓
14	App Launcher	AppLauncher	App Launcher tabs	25/11/2024, 6:06 pm	Classic	✓

Step 5: Creating Custom Fields

- **Customer Details:** Phone and email fields.

1. To create fields in an object:

- Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
- Now click on "Fields & Relationships" >> New
- Select Data Type as a "Phone"
- Click on next.
- Fill the Above as following:
Field Label: Phone number
Field Name : gets auto generated
- Click on Next >> Next >> Save and new.

2. To create another fields in an object:

- Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
- Now click on "Fields & Relationships" >> New
- Select Data type as a "Email" and Click on Next
- Fill the Above as following:
Field Label : Gmail
Field Name : gets auto generated
- Click on Next >> Next >> Save and new.

Setup > OBJECT MANAGER

Customer Details

Details

Fields & Relationships 6 Items, Sorted by Field Label

Q 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		

- **Lookup Fields:** Link Appointments to Customers, Service Records to Appointments, and Billing Details to Service Records.
- **Picklist Fields:** Service status (Started, Completed) and payment status (Pending, Completed).
- **Formula Fields:** Calculate service date from created date.
- **Text Fields:** Vehicle number plate (10 characters, unique) and customer feedback rating (1 character).

Appointments - all fields & relationship:

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

Home

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

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
Appointment	Appointment__c	Lookup(Appointment)		✓
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Quality Check Status	Quality_Check_Status__c	Checkbox		
service date	service_date__c	Formula (Date)		
Service records Name	Name	Auto Number		✓
Service Status	Service_Status__c	Picklist		

Setup

Home

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

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	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	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)		✓

Step 6: Validation 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.

- **Vehicle Number Plate:** Must follow a format (e.g., MH12AB1234).

To create a validation rule to an Appointment Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Appointment object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as " Vehicle ".
4. Insert the Error Condition Formula as : -
`NOT(REGEX(Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))`
5. Enter the Error Message as "Please enter valid number ", select the Error location as Field and select the field as "Vehicle number plate", and click Save.

Setup > OBJECT MANAGER

Billing details and feedback

Validation Rules

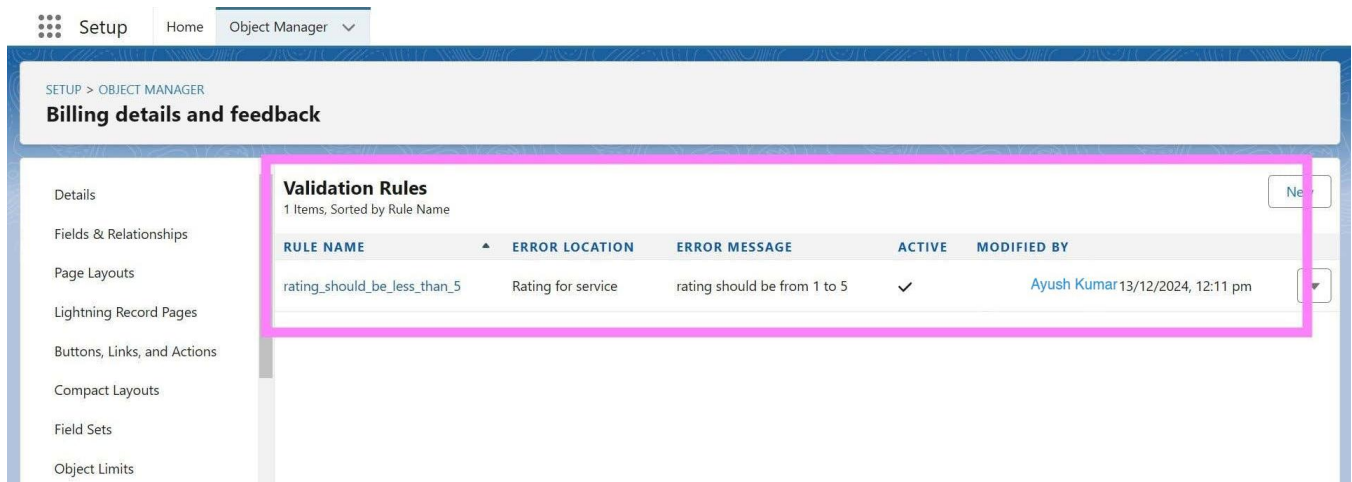
1 Items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
rating_should_be_less_than_5	Rating for service	rating should be from 1 to 5	✓	Ayush Kumar 13/12/2024, 12:11 pm

- **Service Status:** Must be set to "Completed" before record can be saved.
- **Rating:** Customer service rating must be between 1 and 5.

To create a validation rule to an Billing details and feedback Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Billing details and feedback object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as " rating_should_be_less_than_5".
4. Insert the Error Condition Formula as : -
NOT(REGEX(Rating_for_service__c , "[1-5]{1}"))
5. Enter the Error Message as "rating should be from 1 to 5", select the Error location as Field and select the field as "Rating for Service", and click Save.



The screenshot shows the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The left sidebar lists various setup options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, and Object Limits. The main content area is titled 'Billing details and feedback' and displays a table of 'Validation Rules'. The table has columns for Rule Name, Error Location, Error Message, Active status, and Modified By. One rule is listed: 'rating_should_be_less_than_5' with an error location of 'Rating for service', an error message of 'rating should be from 1 to 5', and it is marked as 'Active' with a checkmark. The rule was modified by 'Ayush Kumar' on '13/12/2024, 12:11 pm'.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
rating_should_be_less_than_5	Rating for service	rating should be from 1 to 5	✓	Ayush Kumar 13/12/2024, 12:11 pm

Step 7: Duplicate Rules

- **Matching Rule:** Checks Gmail and phone number to identify duplicate customer details.

To create a matching rule to an Customer details Object

1. Go to quick find box in setup and search for matching Rule.
2. Click on matching rule >> click on New Rule.
3. Select the object as Customer details and click Next.
4. Give the Rule name : Matching customer details
5. Unique name : is auto populated
6. Define the matching criteria as
7.

Field	Matching Method
1. Gmail	Exact
2. Phone Number	Exact
8. Click save.
9. After Saving Click on Activate.

The screenshot displays the Salesforce Setup interface. In the left sidebar, the navigation menu includes 'Setup', 'Home', and 'Object Manager'. Under 'Object Manager', 'Data' is expanded, showing 'Duplicate Management' and 'Matching Rules'. The 'Matching Rules' link is highlighted. The main content area is titled 'Matching Rules' and shows a search bar with 'matchin' entered. Below the search bar, a yellow box highlights the 'Matching Rule' section, which contains the rule name 'Matching customer details'. To the right of this section is a 'Help for this Page' link. Below the highlighted section, the 'Matching Rule Detail' is shown, including fields for Object (Customer Details), Rule Name (Matching customer details), Unique Name (Matching_customer_details), Description, Matching Criteria ((Customer Details: Gmail EXACT MatchBlank = FALSE) AND (Customer Details: Phone_number EXACT MatchBlank = FALSE)), and Status (Active). Action buttons for 'Delete', 'Clone', and 'Deactivate' are also visible.

- **Duplicate Rule:** Prevents duplicate customer records.

To create a Duplicate rule to an Customer details Object

1. Go to quick find box in setup and search for Duplicate rules.
2. Click on Duplicate rule >> click on New Rule >> select customer details object.
3. Give the Rule name as : Customer Detail duplicate
4. Scroll a little in Matching rule section
5. Select the matching rule : Matching customer details
6. And Click on save.
7. After saving the Duplicate Rule, Click on Activate.

The screenshot displays the Salesforce Setup interface for configuring a Duplicate Rule. The left sidebar shows the navigation menu with 'Setup', 'Home', and 'Object Manager' tabs. The 'Duplicate Rules' section is highlighted under 'Data' > 'Duplicate Management'. The main content area shows the 'Customer Details Duplicate Rule' configuration page. The rule is named 'Customer Detail duplicate' and is currently active. The matching criteria are set to '(Customer Details: Email EXACT MatchBlank = FALSE) AND (Customer Details: Phone_number EXACT MatchBlank = FALSE)'. The rule is configured to allow creation and editing, with alerts and reports enabled for creation.

Duplicate Rule Detail		Order	1 of 1 [Reorder]
Rule Name	Customer Detail duplicate		
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	<input checked="" type="checkbox"/> Matching customer details <input checked="" type="checkbox"/> Mapped	Matching Criteria	(Customer Details: Email EXACT MatchBlank = FALSE) AND (Customer Details: Phone_number EXACT MatchBlank = FALSE)
Conditions			

Step 8: Profiles

- **Manager Profile:** Access to all custom objects with extended session time and password settings.

The screenshot shows the Salesforce Setup interface. In the left sidebar, the 'Profiles' link is highlighted. The main content area displays the 'Manager' profile details. The profile name 'Manager' is highlighted with an orange box. The profile is a Custom Profile. The 'Page Layouts' section shows the following assignments:

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

- **Sales Person Profile:** Limited access to objects with tailored permissions.

The screenshot shows the Salesforce Setup interface. In the left sidebar, the 'Profiles' link is highlighted. The main content area displays the 'sales person' profile details. The profile name 'sales person' is highlighted with an orange box. The profile is a Custom Profile. The 'Page Layouts' section shows the following assignments:

Standard Object Layouts	Assignment
Global	Global Layout
Fulfillment Order Item Tax	Fulfillment Order Item Tax Layout

Step 9: Roles and Role Hierarchy

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.

- **Manager Role:** Directly under the CEO.

Creating Manager Role:

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click on Expand All and click on add role under whom this role works.
3. Give Label as "Manager" and Role name gets auto populated. Then click on Save.

- **Sales Person Role:** Reports to the Manager role.

Creating another two roles under manager

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click plus on CEO role, and click add role under manager.
3. Give Label as "sales person" and Role name gets auto populated. Then click on Save.

The screenshot displays the Salesforce Setup interface for the 'Roles' section. The left sidebar shows the navigation menu with 'Roles' selected. The main content area is titled 'Creating the Role Hierarchy' and includes instructions on how to build the hierarchy. A tree view shows the existing role hierarchy, starting with 'VIT Bhopal' at the top. Below it are roles like 'CEO', 'CFO', 'COO', and 'Manager'. The 'Manager' role is highlighted with a yellow arrow, and the 'sales.person' role is also highlighted with a yellow arrow. The 'sales.person' role is shown as a child of the 'Manager' role. Other roles in the hierarchy include 'SVP, Customer Service & Support', 'Customer Support, International', 'Customer Support, North America', and 'Installation & Repair Services'. Each role in the tree has an 'Add Role' link next to it.

Step 10: Users

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

- **Manager User:** Role as Manager, Profile as Manager, Salesforce license.

The screenshot shows the Salesforce Setup interface for a user named Niklaus Mikaelson. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area displays the user's details, including Name, Alias, Email, Username, Nickname, Title, Company, Department, Division, Address, Time Zone, Locale, Language, Delegated Approver, Manager, Role, User License, Profile, Active status, and various user types (Marketing User, Offline User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, Site.com Publisher User, WDC User, Mobile Push Registrations, Data.com User Type, Accessibility Mode, and Debug Mode). The user's role is Manager, user license is Salesforce, and profile is Manager. The user is active.

- **Sales Person Users:** Role as Sales Person, Profile as Sales Person, Salesforce Platform license.

The screenshot shows the Salesforce Setup interface for a user named Ayush Kumar. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area displays the user's details, including Name, Alias, Email, Username, Nickname, Title, Company, Department, Role, User License, Profile, Active status, and various user types (Marketing User, Offline User, Knowledge User, Flow User). The user's role is sales person, user license is Salesforce Platform, and profile is sales person. The user is active.

Step 11: Public Groups

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.

- **Sales Team Group:** Contains all Sales Person roles for sharing access.

Creating New Public Group

1. Go to setup >> type users in quick find box >> select public groups >> click New.
2. Give the Label as "sales team".
3. Group name is autopopulated.
4. Search for Roles.
5. In Available Members select Sales person and click on add it will be moved to selected member.
6. Click on save.

The screenshot displays the Salesforce Setup interface. On the left, the 'Setup' menu is open, showing 'Users' > 'Public Groups'. The main content area shows the 'Public Groups' page for the 'sales team' group. The group details include: Label 'sales team', Group Name 'sales_team', Grant Access Using Hierarchies checked, and Description empty. The group was created by Ayush Kumar on 13/12/2024 at 12:50 pm and modified by the same user at the same time. Below the details, there is a section titled 'All Users in Group' with a 'View Group Members' button. The table below lists the members of the group.

Full Name	Reason For Membership
Niklaus Mikaelson	Manager of Group Member
Tanjul sarathe	Group Member
ashish patel	Group Member
Ayush Kumar	Group Member

Step 12: Sharing Settings

- **Service Records OWD:** Set to Private.
- **Sharing Rule:** Grants read/write access to Manager for Sales Person's service records.

Creating Sharing settings

1. Go to setup >> type users in quick find box >> select Sharing Settings >> click Edit.
2. Change the OWD setting of the Service records Object to private as shown in fig.
3. Click on save and refresh.
4. Scroll down a bit, Click new on Service records sharing Rules.
6. Give the Label name as " Sharing setting"
7. Rule name is auto populated.
8. In step 3 : Select which records to be shared, members of " Roles " >> " Sales person"
9. In step 4: share with, select " Roles " >> " Manager "
10. In step 5 : Change the access level to " Read / write ".
11. Click on save.

The screenshot displays the Salesforce Setup interface for Sharing Settings. The left sidebar shows the navigation menu with 'Setup', 'Home', and 'Object Manager'. The main content area is titled 'Sharing Settings' and lists several sharing rule categories: Work Type Group Sharing Rules, Appointment Sharing Rules, Billing details and feedback Sharing Rules, Customer Details Sharing Rules, and Service records Sharing Rules. The 'Service records Sharing Rules' section is highlighted with a yellow box, showing a rule where 'Owner in Role: sales person' is shared with 'Role: Manager' at the 'Read/Write' access level.

Action	Criteria	Shared With	Access Level
Edit Del	Owner in Role: sales person	Role: Manager	Read/Write

Step 13: Flows

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.

- **Record-Triggered Flow:** Automates updates and email alerts for billing completion.
- Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Update records Element”.

The screenshot displays the Salesforce Flow Builder interface for a flow named "Billing Amount Flow - V1". The flow is configured as a Record-Triggered Flow. The flow diagram on the left shows the following steps: a "Record-Triggered Flow Start" node, followed by a "Run Immediately" node, then an "Amount Update" node (Update Records), an "Email Alert" node (Action), and finally an "End" node. The right panel shows the configuration for the "Update Records" element. It includes fields for "Label" (Amount Update) and "API Name" (Amount_Update). The "Description" field is empty. Below this, there are options for "How to Find Records to Update and Set Their Values", with the first option selected: "Use the billing details and feedback record that triggered the flow". The "Set Filter Conditions" section shows a condition: "Payment_Status_c" equals "Completed". The "Set Field Values for the Billing details and feedback Record" section shows a field "Payment_Paid_c" being set to a value from a record collection: "SRRecord > Service records > Appointment > Ser...".

- **Email Alert:** Sends a "Thank you" message when payments are completed.

Flow Builder: Billing Amount Flow - V1

Last saved on 13/12/2024, 01:25 pm **Active** Run Debug View Tests Save As New Version Save Deactivate

Toolbox

Manager

Search this flow...

New Resource

RESOURCES

Text Templates (1)

alert

ELEMENTS

Actions (1)

Email Alert

Update Records (1)

Amount Update

Flow Diagram:

```

graph TD
    Start([Record-Triggered Flow Start]) --> RunImmediately[Run Immediately]
    RunImmediately --> AmountUpdate[Amount Update Update Records]
    AmountUpdate --> EmailAlert[Email Alert Action]
    EmailAlert --> End([End])
  
```

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

BCC Recipient Address List Not Included

Body

alert

CC Recipient Address List Not Included

- **Another Flow - Update Service Status**

Flow Builder: Update Service Status - V1

Last saved on 13/12/2024, 01:34 pm **Active** Run Debug View Tests Save As New Version Save Deactivate

Flow Diagram:

```

graph TD
    Start([Record-Triggered Flow Start]) --> RunImmediately[Run Immediately]
    RunImmediately --> UpdateRecords[Update records Ele Update Records]
    UpdateRecords --> End([End])
  
```

Update Records

* 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

Field

Service_Status_c

Value

Completed

+ Add Field

Step 14: 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.

- **Handler Class (AmountDistributionHandler):** Calculates service amounts based on selected services.

Code:

"AccountDistributionHandler.apxc"

```
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;
    }
}
}
}

```

- **Trigger (AmountDistribution):** Runs on appointment insert or update to update service amounts.

Code:

"AmountDistribution.apxt"

```

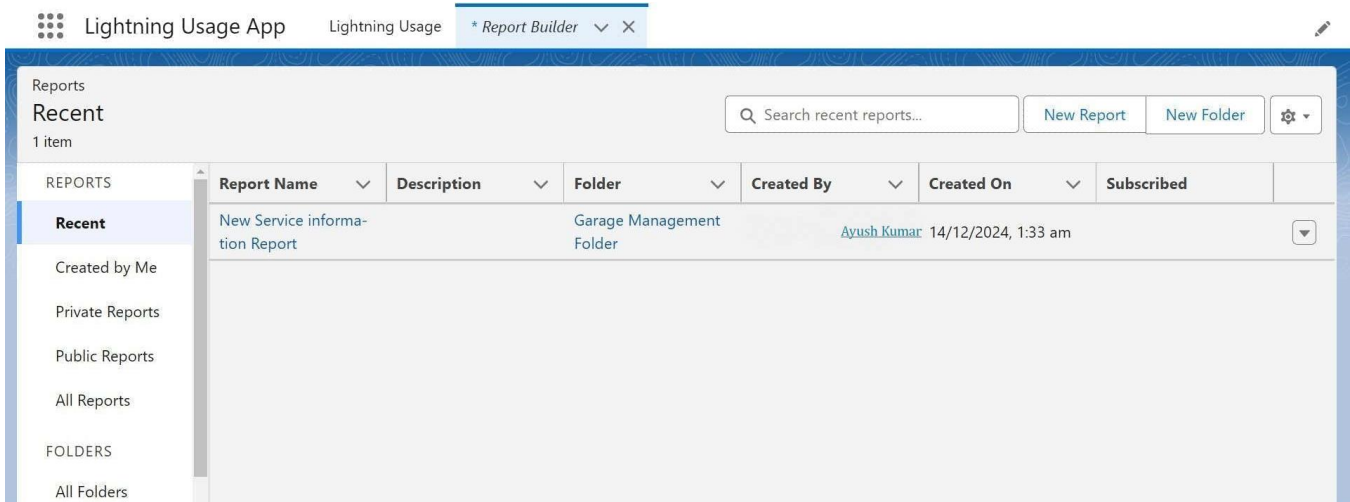
trigger AmountDistribution on Appointment__c (before insert, before
update) {
    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
        AmountDistributionHandler.amountDist(trigger.new);
    }
}

```

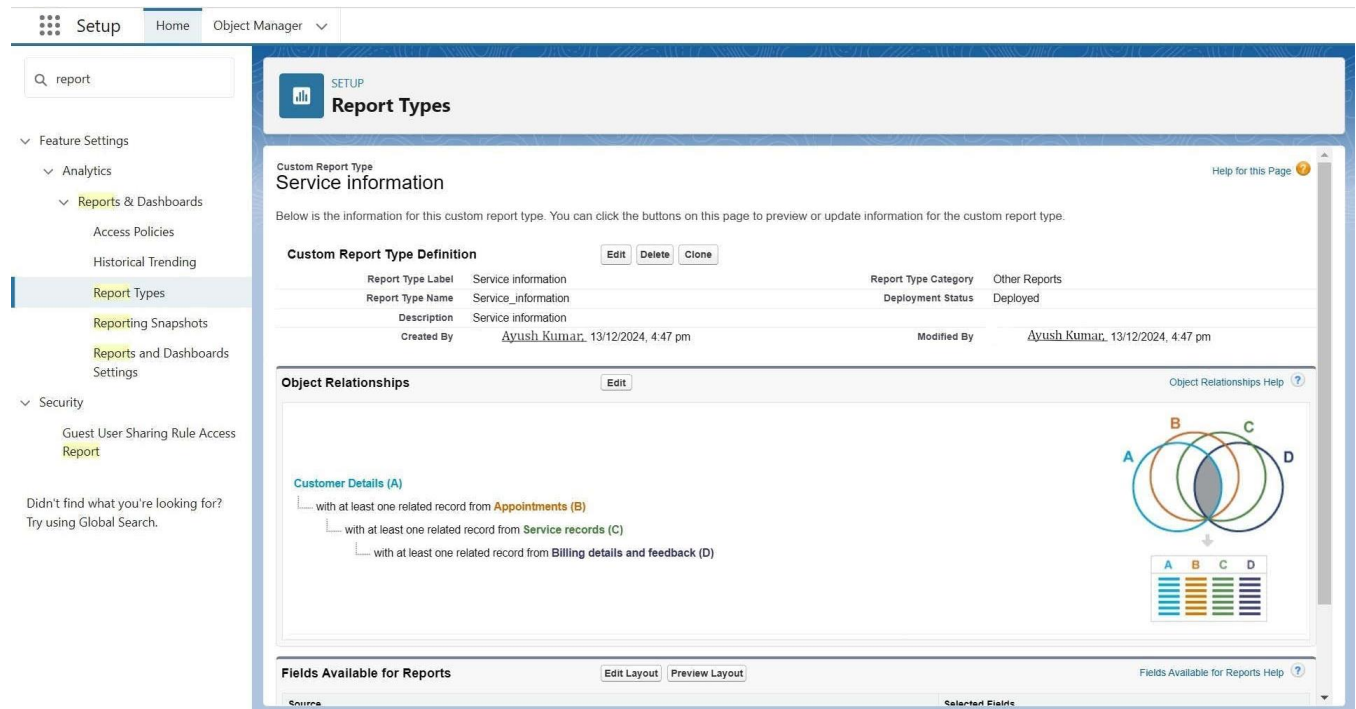
Step 15: Reports

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.

- **Report Folder:** Organize all reports under "Garage Management Folder".
- **Custom Report Type:** Combines Customer Details, Appointments, Service Records,



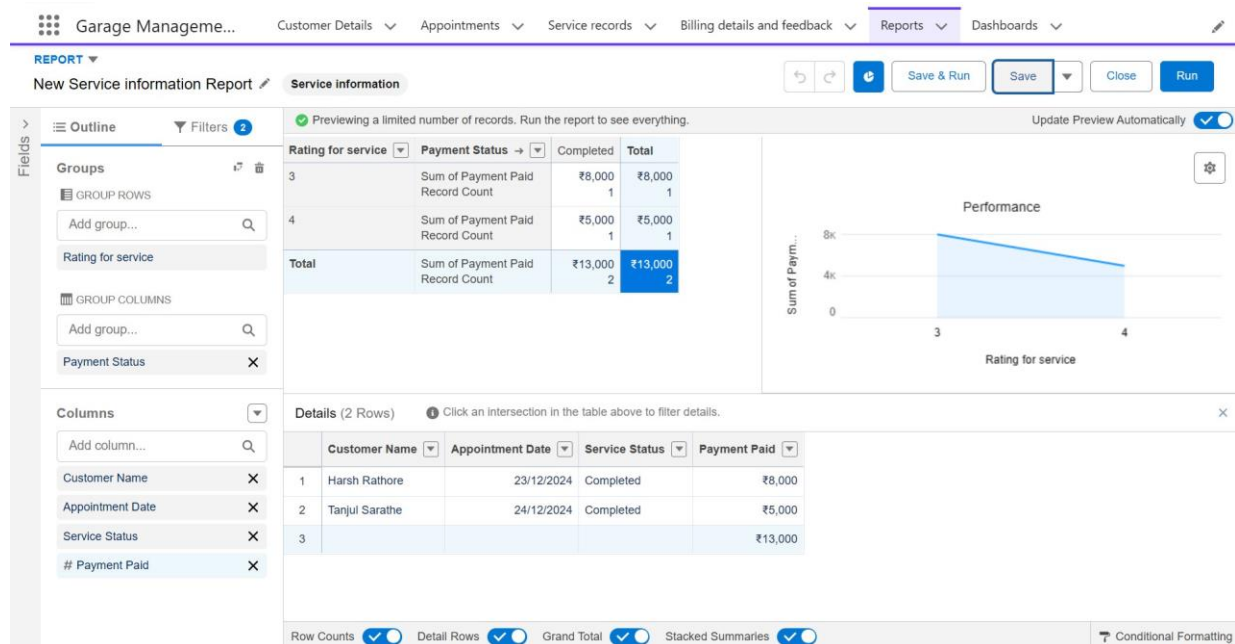
and Billing.



- **Reports:** Custom report "Service Information Report" with fields for customer, appointment date, service status, and payments.

Create Report

1. Go to the app >> click on the reports tab
2. Click New Report.
3. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.
4. Their outline pane is opened already, select the fields that mentioned below in column section.
 1. Customer name
 2. Appointment Date
 3. Service Status
 4. Payment paid
5. Remove the unnecessary fields.
6. Select the fields that mentioned below in GROUP ROWS section.
 1. Rating for Service
7. Select the fields that mentioned below in GROUP ROWS section.
 1. Payment Status
8. Click on Add Chart , Select the Line Chart.
9. Click on save, Give the report Name : New Service information Report
10. Report unique Name is auto populated.
11. Select the folder the created and Click on save.



Step 16: Dashboards

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.

- **Dashboard Folder:** "Service Rating Dashboard" for organizing dashboards.
- **Dashboard Components:** Visualizes service ratings, payment statuses, and operational KPIs.

The screenshot displays the 'Service Rating dashboard' interface. At the top, there's a navigation bar with tabs: 'Garage Manageme...', 'Customer Details', 'Appointments', 'Service records', 'Dashboards', and 'More'. Below this, the 'Dashboards' section shows a list of dashboards under the 'Service Rating dashboard' folder. The table has columns: Name, Description, Folder, Created By, Created On, and Subscribed. One dashboard is listed: 'Customer Review' by 'Ayush Kumar' created on '14/12/2024, 1:49 am'. Below the table, the 'Customer Review' dashboard is expanded, showing a line chart titled 'New Service information Report'. The chart plots 'Sum of Payment Paid' (Y-axis, ranging from ₹0 to ₹8k) against 'Rating for service' (X-axis, with values 3 and 4). The data points are approximately (3, ₹7.5k) and (4, ₹5.5k), connected by a blue line. A legend indicates 'Payment Status' with a blue dot for 'Completed'. Below the chart, there's a link to 'View Report (New Service information Report)'.

DASHBOARDS	Name	Description	Folder	Created By	Created On	Subscribed
Recent	Customer Review		Service Rating dashboard	Ayush Kumar	14/12/2024, 1:49 am	

Customer Review
As of 22-Dec-2024, 12:32 am-Viewing as Ayush Kumar

New Service information Report

Payment Status: Completed

Sum of Payment Paid

Rating for service

View Report (New Service information Report)

Also Subscribe added:- Set the Frequency as " weekly ". >> Set a day as monday.

- and then we saved it.

User Adoption

In our Garage Management System we created records for all objects.

Garage Manage... Customer Details Appointments Service records Billing details and feedback Reports Dashboards

Customer Details

Recently Viewed

5 items • Updated a few seconds ago

Search this list...

	<input type="checkbox"/> Customer Name	
1	<input type="checkbox"/> Meee	
2	<input type="checkbox"/> Pushpraj Dubey	
3	<input type="checkbox"/> Ashish Patel	
4	<input type="checkbox"/> Tanjul Sarathe	
5	<input type="checkbox"/> Harsh Rathore	

Garage Manage... Customer Details Appointments Service records Billing details and feedback Reports Dashboards

Appointments

Recently Viewed

3 items • Updated a few seconds ago

Search this list...

	<input type="checkbox"/> Appointment Name	
1	<input type="checkbox"/> app-007	
2	<input type="checkbox"/> app-006	
3	<input type="checkbox"/> app-005	

Garage Manage... Customer Details Appointments Service records Billing details and feedback Reports Dashboards

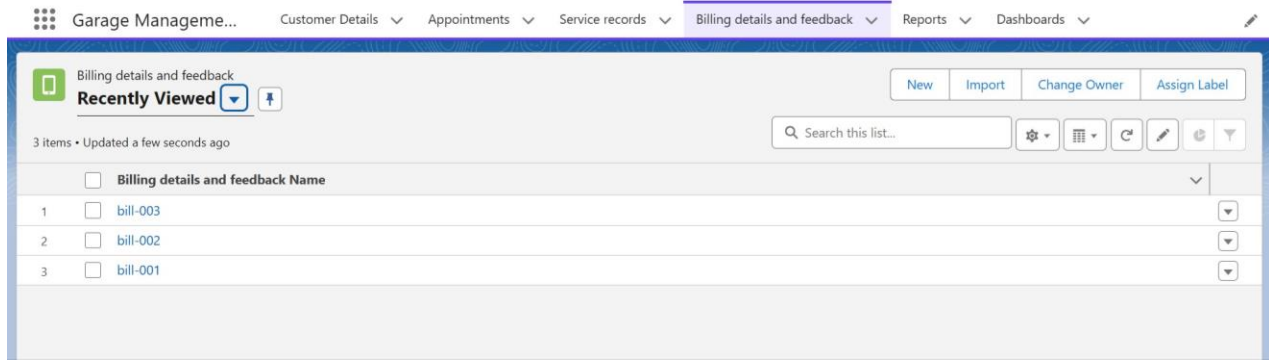
Service records

Recently Viewed

3 items • Updated a few seconds ago

Search this list...

	<input type="checkbox"/> Service records Name	
1	<input type="checkbox"/> ser-014	
2	<input type="checkbox"/> ser-013	
3	<input type="checkbox"/> ser-012	



5. Testing and Validation

5.1. Unit Testing

- Comprehensive testing of individual components, including Apex classes and triggers, to ensure code accuracy and functionality.
- Each module will be tested in isolation to detect and fix issues at an early stage.

5.2. User Interface Testing

- Ensure the user interface is user-friendly, responsive, and works on different devices (desktop, tablet, mobile).
- Validate the functionality of buttons, links, forms, and navigation.

6. Key Scenarios Addressed by Salesforce in the Implementation Project

This section outlines the key scenarios that Salesforce addresses in the implementation of the Garage Management System:

- **Service Appointment Booking:** Customers can book appointments via a self-service portal.
- **Customer Communication:** Automated notifications and reminders are sent to customers.
- **Inventory Management:** Salesforce tracks spare parts and sends reordering alerts.
- **Job Tracking:** Service advisors and technicians can track the status of jobs and update customers in real time.
- **Billing and Payments:** Generate invoices and process payments through an integrated payment gateway.

- **Customer Feedback Collection:** Customers provide feedback after service completion, which is used to improve service quality.

Conclusion

The Garage Management System built on Salesforce provides a comprehensive platform for managing appointments, services, and billing. Through the use of custom objects, tabs, profiles, flows, validation rules, and Apex triggers, the system ensures smooth operations, efficient record management, and enhanced customer satisfaction. This project enables better tracking of garage operations, fosters data-driven decision-making, and supports the long-term growth of garage businesses.

Thank You