

Phase 2: Org Setup & Configuration – Smart Vehicle CRM

This phase documents the Salesforce org configuration required to implement the Smart Vehicle Service & Maintenance CRM project. All settings are aligned with best practices to support users, roles, profiles, and automation.

1. Salesforce Edition

- **Edition Used:** Salesforce Developer Edition
- **Reason:** Free edition with full metadata and automation support suitable for implementation and testing.

2. Company Profile Setup

- **Company Name:** Smart Vehicle CRM .
- **Default Currency:** INR (₹)
- **Timezone:** IST (+5:30)
- **Locale:** English (India)
- **Fiscal Year:** Standard (January-December)

The screenshot shows the 'Company Information' setup page for the organization 'Smart Vehicle CRM'. The page includes sections for Organization Detail, Address, and various system settings like currency and locale. Key details shown include:

- Organization Name:** Smart Vehicle CRM
- Primary Contact:** Gundala Bipin
- Address:** Adapabazar, Hyderabad, India
- Fiscal Year Starts In:** January
- Default Locale:** English (India)
- Currency Locale:** Telugu (India) - INR
- Used Data Space:** 342 KB (7%)
- Used File Space:** 17 KB (0%)
- API Requests, Last 24 Hours:** 0 (15,000 max)
- Streaming API Events, Last 24 Hours:** 0 (10,000 max)
- Restricted Logins, Current Month:** 0 (0 max)
- Salesforce.com Organization ID:** 00DgL00000AWysw
- Organization Edition:** Developer Edition
- Instance:** CAN98

The page also shows the creation and modification details:

- Created By:** OrgFarm EPIC, 8/30/2025, 11:14 PM
- Modified By:** Bipin Gundala, 9/21/2025, 5:39 AM

The URL at the bottom of the page is <https://orgfarm-a7a5e8a683-dev-ed.lightning.force.com/lightning/setup/CompanyProfileInfo/home>.

3. Business Hours & Holidays

- **Business Hours:** 09:00 AM – 06:00 PM (Monday–Sunday)
- **Holidays:** Configured public holidays to prevent automated processes from running on non-working days.
 - ->Gandhi Jayanthi
 - ->Independence Day etc

The screenshot shows the Salesforce Setup interface for 'Business Hours'. The page title is 'Business Hours' under the 'SETUP' tab. It displays the 'Organization Business Hours' section, which defines standard business hours from 9:00 AM to 6:00 PM from Monday to Saturday. A 'Default Business Hours' checkbox is checked. Below this, a 'Holidays' section lists two entries: 'GandhiJayanthi' and 'IndependenceDay', both categorized as 'Public Holiday'.

Action	Holiday Name	Description	Date and Time
Edit Del	GandhiJayanthi		10/2/2025 All Day
Edit Del	IndependenceDay	Public Holiday	8/15/2026 All Day

At the bottom of the page, the URL is https://orgfarm-a7a5e8a683-dev-ed.develop.lightning.force.com/lightning/setup/Holiday/home

4. User Setup & Licenses

- **User Licenses Used:** Salesforce, Salesforce Platform
- **Profiles Created:**
 1. Service Manager Profile
 2. Technician Profile
 3. Customer Service Agent (CSA) Profile
- **Roles Created:**
 - Service Manager
 - Technician (reports to Service Manager)
 - Customer Service Agent (reports to Service Manager)
- **Users Created:**

User	Role	Profile	License
Service Manager	Service Manager	Service Manager Profile	Salesforce
Technician	Technician	Technician Profile	Salesforce
Customer Service Agent (CSA)	Customer Service Agent	CSA Profile (Platform User)	Salesforce Platform

Purpose: Proper user, role, and profile setup ensures access control and segregation of duties.

The screenshot shows a Mac OS X desktop environment. In the foreground, a terminal window is open with the command `git log` running, displaying a list of commits for a project named "SmartVehicleCRM_Clean". The commits are dated from September 21, 2018, at 6:37 PM. In the background, a code editor (VS Code) is visible, showing the file `ProjectDocumentation.txt` which contains detailed information about the project's implementation and profiles created. The code editor also shows other files like `package.xml` and `README.md`. The desktop dock at the bottom includes icons for various applications like Finder, Safari, Mail, and others.

```
bipungundala@Bipins-MacBook-Air ~ % git log
commit 45f4d40... [redacted]
Author: Bipin Gundala <bipungundala@Bipins-MacBook-Air>
Date:   Sat Sep 21 18:37:00 2018 +0530

    This project was implemented from scratch in Salesforce with the following
    1. Custom Objects Created:
        Vehicle__c - Stores customer vehicle details.
        Service_Request__c - Tracks service bookings and status.
        Technician__c - Holds technician details.
        Parts_Inventory__c - Manages spare parts & stock.
        Service_Parts_Used__c - Logs parts consumed during service.
        Invoice__c - Handles billing and payments.

    2. Page Layouts & Fields Configured:
        Custom fields (Date, Lookup, Picklist, Currency, etc.) created for each object.
        Page layouts customized for Technician, Service Request, Vehicle, Parts Inventory, Invoice.

    3. Profiles Created:
        Service Manager Profile - Read/Write on all service-related objects.
        Technician Profile - Read/Write only on assigned requests, update status & parts used.
        Customer Service Agent (CSA) Profile - Can create service requests, view status, and manage customer communication.

hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: repository.
hint: Note: Never specify how to reconcile divergent branches.
bipungundala@Bipins-MacBook-Air SmartVehicleCRM_Clean % git push origin main --force
Enumerating objects: 86, done.
Counting objects: 100% (86/86), done.
Delta compression using up to 8 threads.
Compressing objects: 100% (79/79), done.
Writing objects: 100% (86/86), 1.00 MiB | 4.69 MiB/s, done.
Total 86 (delta 34), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (34/34), done.
To https://github.com/bipungundala/SmartVehicle-Service-Maintenance-CRM.git
 * [new branch] main      -> main (forced update)
bipungundala@Bipins-MacBook-Air SmartVehicleCRM_Clean %
* [history restored]
```

All Users

Action	Full Name	Alias	Username	Role	Active	Profile
Edit	Chatter Expert	Chatter	chatty.00dg100000awyswuaad.gn6uaxp0fh@chatter.salesforce.com	Technician	✓	Chatter Free User
Edit	doe_alex	adoe	y22cd044@gmail.com	Customer Service Agent	✓	Technician Profile
Edit	doe_bob	bdoe	bipin.gundala@gmail.com	Service Manager	✓	CSA Platform Profile
Edit	doe_John	ido	bipin.gundala044@gmail.com	System Administrator	✓	Service Manager Profile
Edit	EPIC_OrgFarm	OEPIIC	epic.b2f701a9f415@orgfarm.salesforce.com	System Administrator	✓	System Administrator
Edit	Gundala_Bipin	bip	bipin.gundala528@openforce.com	User Integration	✓	Analytics Cloud Integration User
Edit	User_Integration	integ	integration@00dg100000awyswuaad.com	User_Security	✓	Analytics Cloud Security User
Edit	User_Security	sec	insightssecurity@00dg100000awyswuaad.com			

Creating the Role Hierarchy

You can build on the existing role hierarchy shown on this page. To insert a new role, click **Add Role**.

Your Organization's Role Hierarchy

- [Collapse All](#) [Expand All](#) [Show in tree view](#)
- Smart Vehicle CRM
 - [Add Role](#)
 - CEO [Edit](#) [Del](#) [Assign](#)
 - [Add Role](#)
 - CFO [Edit](#) [Del](#) [Assign](#)
 - [Add Role](#)
 - COO [Edit](#) [Del](#) [Assign](#)
 - [Add Role](#)
 - Customer Service Agent [Edit](#) [Del](#) [Assign](#)
 - [Add Role](#)
 - Service Manager [Edit](#) [Del](#) [Assign](#)
 - [Add Role](#)
 - Technician [Edit](#) [Del](#) [Assign](#)
 - [Add Role](#)
 - SVP_Customer Service & Support [Edit](#) [Del](#) [Assign](#)
 - [Add Role](#)
 - SVP_Human Resources [Edit](#) [Del](#) [Assign](#)
 - [Add Role](#)
 - SVP_Sales & Marketing [Edit](#) [Del](#) [Assign](#)
 - [Add Role](#)

5. Permission Sets

- Configured to grant **additional access** such as:
 - Sending email alerts
 - Viewing dashboards & reports
 - Managing inventory
-

6. Org-Wide Defaults (OWD)

Object	Default Access
Vehicle__c	Private
Service_Request__c	Private
Invoice__c	Private
Parts_Inventory__c	Public Read-Only

Purpose: Lockdown records by default to maintain data security; sharing rules define exceptions.

7. Sharing Rules

- **Rule Configured:**
 - Service Manager-owned Service Requests → Shared with Technicians (Read/Write)

Purpose: Allows technicians to access only the requests they are assigned while keeping other records private.

The screenshot shows the Salesforce Sharing Settings page. At the top, there's a search bar with 'sharing' typed in. Below the search bar, the 'Sharing Settings' section is visible, which includes sections for 'Work Type Sharing Rules', 'Work Type Group Sharing Rules', 'Invoice Sharing Rules', 'Parts Inventory Sharing Rules', 'Service Request Sharing Rules', 'Technician Sharing Rules', and 'Vehicle Sharing Rules'. Each section has 'New' and 'Recalculate' buttons. The 'Service Request Sharing Rules' section is expanded, showing a table with one row:

Action	Criteria	Shared With	Access Level
Edit Del	Owner in Role: Service Manager	Role: Technician	ReadWrite

At the bottom of the page, there's a URL bar with the address <https://orgfarm-a7a5e8a683-dev-ed.develop.lightning.force.com/lightning/setup/SecuritySharing/home>.

8. Login Access Policies

- **Trusted IP Ranges:** Configured to restrict login to company network.

9. Sandbox & Deployment

- **Development Environment:** Salesforce Developer Org
- **Deployment Strategy:**
 - Metadata retrieved using **Salesforce CLI**
 - Stored and version-controlled in **VS Code**
 - Pushed to **GitHub repository** for submission and version tracking

10. Deployment Basics

- **Retrieve Metadata:** Objects, profiles, layouts, sharing rules.
- **Clean Metadata:** Removed unnecessary components to simplify repo.
- **Version Control:** GitHub used to maintain clean project history.

The screenshot shows a Mac desktop environment. At the top is a dark-themed menu bar with options like Code, File, Edit, Selection, View, Go, Run, Terminal, Window, and Help. The date and time 'Sun Sep 21 6:43PM' are in the top right. Below the menu is a toolbar with various icons. The main area contains a terminal window with a command-line interface and a large Code editor window titled 'SmartVehicleCRM_Clean'. The Code editor displays a file named 'package.xml' containing XML metadata for a Salesforce project. The XML includes sections for objects, profiles, sharing rules, and custom profiles. The sidebar on the left lists files and folders related to the project, such as 'force-app/main/default', 'manifest', and various configuration files. The bottom of the screen shows the Mac OS X dock with various application icons.

The screenshot shows a GitHub repository page for 'Smart-Vehicle-Service-Maintenance-CRM'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository name 'Smart-Vehicle-Service-Maintenance-CRM' is displayed with a public status. The main content area shows a list of files in the 'main' branch. The files listed are: .husky, .vscode, config, force-app/main/default, manifest, scripts, .forceignore, .gitignore, .prettierignore, ProjectDocumentation.txt, README.md, and eslint.config.js. Each file entry includes a preview icon, the file name, a description, and a timestamp indicating it was added 4 days ago. To the right of the file list is an 'About' section describing the project as a Salesforce CRM implementation for the automobile service industry. It highlights features like service request automation, technician allocation, inventory tracking, preventive maintenance reminders, and customer communication. Below the About section are sections for Releases, Packages, and a link to Create a new release. There are also links for Readme, Activity, Stars, Watching, and Forks.

Outcome:

The Salesforce org is configured to support the Smart Vehicle CRM project with proper roles, profiles, users, sharing rules, business hours, and security. The org is ready for automation (flows, approval processes) and reporting dashboards.