

# Application to make the Gas filling Station easy using CRM ( Developer)

## 1. Executive Summary: The Quick Overview

This project is about building a **smart, central computer system** (we call it a CRM, but think of it as a central hub) to manage everything at our gas stations. Our main goal is to stop relying on paper and guesswork.

### Our Objectives:

1. **Stop losing fuel:** Track stock perfectly from delivery to sale.
2. **Make customers happy:** Know who our repeat buyers are and reward them.
3. **Speed up work:** Let the computers handle the paperwork and alerting.

**The Key Benefits:** Everything will be in **one place** (Centralized Data), the computers will do the boring updates (**Automation**), and managers will get **instant reports** to make smart decisions. This will save money, save time, and bring in more repeat customers.

## 2. Introduction

### Purpose of this Document: The Plan

This document is the **official blueprint** for building the new system. It explains *why* we need it, *what* the new system will look like, *how* we will build and test it, and *who* will be using it. It is the guide for everyone involved.

### Scope of the Project: What the System Will Cover

The new system will handle four key parts of the gas station:

1. **Customer Files:** Managing loyalty programs and keeping track of who buys our fuel.
2. **Fuel Stock:** Tracking how much petrol and diesel is in every tank, every minute.
3. **Sales Records:** Logging every single transaction (every time someone fills up).

4. **Buying Fuel:** Managing which suppliers we buy from and tracking the delivery of new stock.

**What it will NOT cover:** We will not use it to run the card machines or the actual fuel dispensers—it will just *record the data* that those existing systems produce.

**Target Audience: Who Will Use It?**

- **Station Staff (Attendants/Cashiers):** They will use the system on a mobile device to quickly process sales and look up customer loyalty IDs.
- **Station Managers:** They will use it daily to check fuel stock, see sales reports, and manage their staff.
- **Head Office Team:** They will use it to choose suppliers, check overall sales performance, and manage the loyalty program.

**3. Business Requirements: Why We Need This**

**Current Challenges: Our Problems Today**

1. **Messy Customer Records:** We often don't collect customer data, or it's stored on paper or in messy spreadsheets. **We can't offer personalized service.**
2. **Guessing Fuel Levels:** Staff manually check tanks with a measuring stick. This is slow and prone to human **errors that lead to fuel loss or stockouts.**
3. **Slow Paperwork:** It takes hours to tally up sales and reconcile cash at the end of the day. **We have no idea what's happening until the next morning.**
4. **Waiting to Order:** We only order new fuel when a tank is almost empty. **This is stressful and costs more money.**

**Desired Outcomes: What We Want to Achieve**

- **Be More Efficient:** Cut the time spent on manual reporting by half.

- **Keep Customers:** Start a loyalty program that brings repeat customers and increases their visits by 15%.
- **Stop Fuel Loss:** Get real-time tracking that helps us find and fix unaccounted fuel loss.

### Functional Requirements: The Must-Haves (Features)

What We Need It  
To Do                      Simple Meaning

**Customer Tracking**                      Store all customer details and automatically calculate their loyalty points after a sale.

**Inventory Alerts**                      Automatically send an email or message to the manager when the Diesel tank hits a dangerously low level.

**Accurate Transactions**                      Record the exact volume, price, time, and payment method for every single sale, linking it to the staff member who sold it.

**Supplier Tracking**                      Keep a database of all our suppliers, their current prices, and track every fuel delivery order from start to finish.

**Variance Reporting**                      Create a simple chart that shows us the difference between the fuel we *think* we sold and the fuel that is *actually* left in the tank.

### Non-Functional Requirements: How Well It Must Work

- **Speed (Performance):** It must be fast. Even during the busiest times, a sales transaction must process in less than three seconds.
- **Safety (Security):** Only managers can see sensitive reports, and staff can only see their own sales. All customer data must be protected and locked down.
- **Usability:** The mobile app for the attendants must be so simple that new staff can use it with minimal training.

## 4. Solution Design: How We Will Build It

### Salesforce Architecture: The Tools We Use

We will build the system on the **Salesforce** platform. We will use the standard parts of Salesforce for things like contacts (our customers and suppliers), but we will mainly build **Custom Objects** (special "folders") tailored just for our gas station needs.

### Data Model: The Main Folders

We will create new digital 'folders' to hold our specific information:

1. **Gas Station Folder:** Holds the unique details for each physical location.
2. **Buyer Folder:** Holds the records for all our loyalty members and their point balances.
3. **Fuel Details Folder:** Holds the daily log of stock levels for Petrol, Diesel, etc.
4. **Transaction Folder:** Holds every sales receipt.
5. **Supplier Folder:** Holds our supplier details and delivery receipts.

### Process Flows: The Automated Steps

We will set up **automated rules** (called **Flows**) to handle routine work. For example:

- **Inventory Alert Flow:** When a staff member records a sale, the system automatically checks the new fuel level. If the level is too low, the system **automatically sends a "Time to order!" alert** to the manager.
- **Loyalty Flow:** When a sale is completed, the system **automatically calculates and adds the points** to the customer's **Buyer Folder**.

### User Interface Design: What the Screens Look Like

- **Attendant Screen (Mobile):** The screen will be simple, with big buttons and few steps, focused entirely on processing the sale and looking up a loyalty ID quickly.

- **Manager Screen (Dashboard):** A single "dashboard" screen that shows 5-7 key charts and meters: how much money we made today, how much fuel is left, and a warning meter for fuel loss.

### **Integrations: Connecting to Other Systems**

We must ensure the CRM can talk to our other machines:

- **POS/Sales System:** We need a connection so that when a sale finishes at the pump, the transaction data **automatically jumps into** our new CRM system.
- **Accounting System:** We will send the final daily sales numbers from the CRM over to the main accounting software.

## **5. Implementation Plan: The Action Steps**

### **Phases and Milestones: The Timeline**

The project will happen in clear stages:

1. **Phase 1: Planning (4 Weeks):** Get all the requirements confirmed, and design the final data folders and process rules.
2. **Phase 2: Building (8 Weeks):** Create the custom folders, build the automatic rules (Flows), and build the reports and dashboards.
3. **Phase 3: Testing and Training (6 Weeks):** Let the staff try it out (**User Acceptance Testing, or UAT**) to make sure it meets their needs. Write the user manuals.
4. **Phase 4: Launch (2 Weeks):** Move all the old data into the new system, launch it at one station first (**the Pilot**), and provide immediate on-site support.
5. **Phase 5: Full Rollout:** Launch the system at all remaining gas stations in small groups.

### **Roles and Responsibilities: Who Does What**

- **Project Sponsor:** The boss who approves the money and makes big decisions.
- **Project Manager:** The person who runs the schedule and manages the team.
- **Salesforce Builders (Admins/Developers):** The technical people who build the system inside Salesforce.
- **Key Users (Managers):** The people who test the system and help train their own staff.

### **Data Migration Strategy: Moving the Old Stuff**

We will carefully move all existing **customer records** and **supplier details** into the new system. We won't move every single old receipt, but we will move some summarized sales data so we have a history to compare against.

### **Testing Plan: Making Sure It Works**

We will test the system in three main ways:

1. **Individual Component Tests:** Check that every little rule and field works correctly on its own.
2. **Connection Tests (Integration):** Check that the POS/sales machine successfully talks to the CRM.
3. **Staff Tests (UAT):** Staff pretend they are having a busy day and use the system to ensure it works smoothly in a real-world setting.

## **6. Training and Support**

### **Training Plan: Teaching the Team**

Training will be specific to the person's job:

- **Attendants:** Hands-on training focused only on how to use the mobile app for sales and customer lookups.

- **Managers:** Training on how to check the reports, manage inventory, and fix small problems themselves.
- **Head Office:** Training on procurement, supplier management, and strategic reporting.

### **Support Model: Fixing Problems**


- **Level 1 (Managers):** They handle simple issues like forgotten passwords or simple "how-to" questions.
- **Level 2 (IT Team):** They handle technical errors, security problems, and fixing any system bugs that pop up.

## **7. Appendices: Extra Details**

This section is for all the technical details that are too complicated for the main document, like:

- **Glossary:** A list of all the technical words and their definitions.
- **User Manuals:** Step-by-step instructions for using the system.
- **Test Cases:** Detailed documents showing every scenario we tested (e.g., "What happens if a customer has 0 points and tries to redeem them?").

FINAL RESULT:



Q Search...

★

+


🔄

?

⚙️

🔔

2



⋮

GAS STATION

Suppliers

▼

Gas Stations

▼

Buyers

▼

Fuel details

▼

Supplier

HP

New Contact

Edit

New Opportunity

▼

Related


Details

Supplier Name

HP

✎

Owner




[sravya sameera](#)

🔒

sum of Fuel supplied


0

Created By



[sravya sameera](#) · 9/6/2025, 3:03 AM

Last Modified By



[sravya sameera](#) · 9/6/2025, 3:09 AM