

Gas Station Management CRM: A Smart Solution

This project involves developing an intelligent, centralized

Customer Relationship Management (CRM) system to manage all operations at our gas stations, acting as a main digital hub. The primary aim is to move away from unreliable paper records and guesswork.

Core Goals and Benefits

The key objectives are focused on efficiency and control:

- **Preventing Fuel Loss:** Achieving perfect stock tracking from the moment fuel is delivered to the point of sale.
- **Enhancing Customer Satisfaction:** Identifying and rewarding repeat customers to build loyalty.
- **Accelerating Operations:** Utilizing computers to manage paperwork and automated alerts.

By centralizing all data, automating mundane updates, and providing managers with instant reports for data-driven decisions, the system will deliver significant benefits. Ultimately, this effort will result in reduced costs, saved time, and an increase in loyal, repeat customers.

Project Scope: What the System Covers

The new CRM will manage four critical areas of the gas station business:

1. **Customer Management:** Maintaining loyalty programs and tracking repeat fuel buyers.
2. **Fuel Inventory:** Monitoring the exact amount of petrol and diesel in every tank in real-time.
3. **Sales Tracking:** Recording every transaction, including all fill-ups, as a sales record.
4. **Procurement:** Managing fuel suppliers and overseeing the delivery process for new stock.

What is *not* covered is the operation of existing physical hardware, like card machines or the fuel dispensers; the new system will instead **record the data** generated b

Addressing Current Challenges

We are currently facing several problems that this new system is designed to solve:

- **Disorganized Customer Data:** Customer information is often not collected or is stored in

messy spreadsheets or on paper, making personalized service impossible.

- **Manual and Inaccurate Fuel Checks:** Staff must manually check tank levels with a measuring stick, a slow method prone to errors that cause fuel loss or stockouts.
- **Delayed Reporting:** Reconciliation of cash and sales takes hours at the end of the day, leaving managers without crucial information until the following morning.
- **Reactive Ordering:** New fuel is only ordered when tanks are almost empty, leading to stress and higher costs.

The desired outcomes are to cut manual reporting time by half, implement a loyalty program to boost repeat customer visits by 15%, and achieve real-time tracking to eliminate unaccounted fuel loss.

Key System Requirements (Features)

The system is required to have the following capabilities (Functional Requirements):

- **Customer Tracking:** Store all customer information and automatically calculate loyalty points after each sale.
- **Inventory Alerts:** Automatically message the manager (e.g., via email) when the diesel stock drops to a critically low level.
- **Accurate Transactions:** Record the precise volume, price, time, payment method, and the staff member responsible for every sale.
- **Supplier Tracking:** Maintain a database of suppliers, their current prices, and track new fuel orders through to delivery.
- **Variance Reporting:** Produce a simple chart comparing the *theoretical* fuel sold with the *actual* fuel remaining in the tank.

Solution and Implementation

Building on the Salesforce Platform

The system will be built on the

Salesforce platform. While leveraging standard Salesforce components for basic records like contacts, the core of the system will be built using

Custom Objects—special digital "folders" tailored to the gas station's unique needs.

The new data structure will include dedicated digital "folders" for:

- **Gas Station:** Details unique to each physical location.
- **Buyer:** Records for loyalty members and their point balances.
- **Fuel Details:** Daily logs of inventory levels for Petrol, Diesel, etc..
- **Transaction:** Every sales receipt.
- **Supplier:** Supplier information and delivery receipts.

Automated Processes (Flows): Routine tasks will be handled by automated rules called **Flows**. For example, the

Inventory Alert Flow will check the new fuel level after a sale and automatically send a "Time to order!" alert to the manager if the stock is too low. The

Loyalty Flow will automatically calculate and credit points to the customer's Buyer Folder upon sale completion.

Implementation Timeline (Phases)

The project will be completed in clear stages:

- **Phase 1 (4 Weeks) – Planning:** Finalizing all requirements, data folders, and process rules.
- **Phase 2 (8 Weeks) – Building:** Creating the custom components, automated rules (Flows), reports, and dashboards.
- **Phase 3 (6 Weeks) – Testing and Training:** Conducting **User Acceptance Testing (UAT)** with staff and developing user manuals.
- **Phase 4 (2 Weeks) – Launch:** Migrating existing data, launching the system at a single **Pilot** station, and providing immediate on-site support.
- **Phase 5 – Full Rollout:** Deploying the system to all remaining gas stations in small

batches.

Training and Support

Training will be specific to the user's role:

- **Attendants:** Will receive hands-on training focused on using the mobile app for sales and customer lookups.
- **Managers:** Will be trained on checking reports, managing inventory, and fixing minor issues.
- **Head Office:** Will be trained on procurement, supplier management, and strategic reporting.

For support,

Level 1 will be handled by Managers for simple issues like passwords and "how-to" questions.

Level 2 will be handled by the IT Team for technical errors, security problems, and system bugs.

Non-Functional Requirements: Defining System Quality

While the functional requirements specify *what* the system must do, these **Non-Functional Requirements (NFRs)** dictate *how well* it must perform. These attributes are often crucial for the success of any digital solution, particularly for a high-volume retail application.

Performance and Reliability

The system must be ready to work around the clock; therefore, a high degree of **availability** is non-negotiable, targeting **99.9%** uptime. To prevent customer dissatisfaction, every sales transaction must be authorized and recorded instantly, typically **within three seconds**, even during the busiest periods. The architecture must be robust enough to handle the strain of up to 50 simultaneous attendants using the mobile application across all locations without any dip in responsiveness or speed.

Security and Usability

Data protection is paramount. Customer loyalty data, alongside any sensitive financial information, must be secured using **end-to-end encryption** to ensure compliance with privacy standards. Furthermore, the system relies heavily on **access control**, meaning staff should only be able to view and manipulate data that is explicitly necessary for their specific job role. Despite the technical complexity behind the scenes, the end-user interface must be highly

usable and intuitive; the mobile POS application needs minimal training time and clear, oversized buttons to prevent errors during hurried transactions.

Technical Structure: Deeper Data and Automation

The intelligence of the platform resides in its underlying **Salesforce Custom Objects** and automated business logic, known as **Flows**. These components manage the business processes previously done manually.

Core Data Structure

The **Fuel Details** custom object is the cornerstone of inventory control, containing critical fields such as the **real-time volume** (in liters), the timestamp of the **last successful fill-up**, and the **low-stock threshold** that triggers immediate re-ordering. Conversely, the **Transaction** object acts as the system's central ledger, detailing not only the fuel type and volume but also linking to the specific **Gas Station**, the attendant who processed the sale, and the **Buyer** record for loyalty tracking.

The Power of Automation (Flows)

Routine operations are handled automatically, eliminating manual errors. The **Inventory Alert Flow** activates immediately upon the completion of any fuel sale. This automation checks the remaining stock against a pre-set low-stock limit; if the level falls beneath this critical point, a warning text or email is automatically dispatched to the relevant station manager, ensuring a proactive ordering process. Similarly, the **Loyalty Flow** instantaneously calculates and credits points to the customer's account based on the purchase amount, removing any opportunity for staff error in loyalty calculations.

Project Validation and Long-Term Support

The final phases of the project emphasize rigorous testing and the creation of detailed documentation to ensure the solution's longevity.

Multi-Layered Testing Protocol

The deployment follows a multi-layered testing protocol. This procedure begins with **Individual Component Tests**, verifying that every single formula, field, and piece of logic operates exactly as intended. Subsequently, **Connection Tests** guarantee seamless, real-time data exchange between the physical Point-of-Sale hardware and the Salesforce platform. Crucially, the final phase is **User Acceptance Testing (UAT)**, during which employees simulate high-volume workdays to provide the final, essential sign-off that the system is reliable under the stress of a real-world environment.

Critical Documentation

The **Appendices** are not optional, but vital long-term blueprints. They contain all the necessary technical details required for ongoing maintenance and future development. The **Test Cases** document is especially important, as it includes detailed evidence for every scenario validated (e.g., verifying the correct process when a customer attempts to redeem points when their balance is zero). This comprehensive documentation is essential for safeguarding the system's maintainability after the initial project team hands over support responsibilities.

FINAL RESULT:

