GENROCKET AUTOMOTIVE PARTS RETAILER CASE STUDY

AUTOMOTIVE PARTS RETAILER

The $130 billion-dollar automotive aftermarket is dominated by local retailers who supply parts and accessories for cars and trucks in the United States and around the world. Increasingly, these brick and mortar retailers are competing with online suppliers for their share of DIY consumers and commercial mechanics working in automotive repair shops.

Through superior customer service, the expansion of retail locations and the addition of regional distribution hubs, auto parts suppliers are forging a multi-prong strategy to respond to the competitive threat posed by Amazon and other ecommerce powerhouses. The more established auto parts retailers are adding new retail locations, beefing up parts inventories and streamlining their supply chains and delivery methods to better support the customer.

Regional distribution hubs are capable of stocking tens of thousands of items. This enables the rapid resupply of surrounding stores and provides customers with more buying options and faster delivery. With many thousands of store locations to support, the major retailers have come to rely on their information systems to efficiently manage inventory levels and to rapidly provision new store locations.

One major automotive parts retailer looked to GenRocket for help in meeting their software testing challenges with an *Enterprise Test Data Automation* solution. In this case study we will cover two use cases where comprehensive test data was provisioned to support the quality assurance goals of their store inventory management and new store provisioning software applications.

USE CASE #1:

Store Inventory Adjustment - Change the quantity by part or SKU

Current Workflow and Testing Team Needs:

When a customer is placing an order for an auto part from the online portal, the system checks the SKU (Stock Keeping Unit) ID against the inventory available and if the quantity is low the system will provide a list of store locations in which the part is available based on the customer address.

The testing team wanted to be able to easily test a scenario where an order is placed online and inventory values are updated and suggestions to nearby stores are made when the order values are higher than the available inventory.

Part 1: Updating the Quantity:

The Testing team was testing the workflow manually by updating the inventory values using a spreadsheet. SKU ID and Quantities are stored in an Unstructured File Format and the team uses a C program to convert the binary file to a fixed file format.

For this use case, GenRocket implemented the following workflow and integrated it with the existing Test Case.

- Used GroovyScriptReceiver to call C-program to convert binary data to a fixed file format
- Ran a GenRocket Scenario to perform the Synthetic Data Replacement on the fixed file
- Used GroovyScriptReceiver to call C-program to convert the fixed file back to binary format

GenRocket was able to successfully implement the solution into the existing Test Environment.

Part 2: Updating the Store Quantities

The testing team was not able to test the scenario where the Parts availability was provided based on the store location. To be able to successfully test the scenario, GenRocket needed to perform SQL updates on store quantities within a database and call the customer API to provide the Store location.

For testing this workflow, the GenRocket API was used to update the GenRocket Receivers in real-time with the values retrieved from the API.

- Used GroovyScriptReceiver to make API call to retrieve store numbers
- Used GenRocket API to dynamically update the GenRocket Scenario with the list of storeNumbers
- Executed the GenRocket Scenario to update the quantity on hand using the GenRocket SQLUpdateReceiver

The implemented solution shows the power of the GenRocket API and how GenRocket Generators can be updated in real-time thus providing all the test data required for testing the required workflow.

USE CASE #2:

Capability to Provision a New Store Environment

Creating a new Store for testing purposes was always a challenge for the customer as a new Store requires data for ~200 Local Databases, PostgreSQL and MySQL. It also requires Pricing, Item availability, Deals, Rebates, Customer Data, Commercial Customer data, Vendor Portal, etc. to be communicated with the central data center. The customer's automated process was taking 6 hours to create a new Store and an additional 4 hours to create the test data for the stores by copying from one of the test environments.

This is a complex environment where minimal documentation was available. GenRocket broke down the test data requirements into different epics (interrelated stories). GenRocket identified the requirements for generating Satellite Stores which are available throughout the customer's region, the Hub which maintains a lot of inventory and Mega Hub. The diverse customer Store environment required GenRocket to generate Test Data and insert the data into an Oracle DB and Mainframe for the data center and ~200 Local Databases, PostgreSQL and MySQL for the stores.

The test data was provisioned and implemented for **Use Case #2** in three phases moving from basic test data integration in Phase 1 to more advanced test data integration by the completion of Phase 3:

Phase 1:

- Vendor Portal Parts
- Consumer Customers
- Commercial Customers
- GenRocket Integration with the existing automation framework

Phase 2:

- Enable deals based on parts or SKU
- Centralized Inventory Adjustment
- Delete Shadow Profiles

Phase 3:

- Stock and consignment
- Customer with TAX Exempt / Non TAX Exempt
- Price Override - Update in Data Center for Price Override

By storyboarding and creating an Epic with multiple interrelated stories GenRocket broke down the implementation into three Phases. Using this approach GenRocket was able to successfully implement a complete test data solution into the customer's existing test environment.

By embracing the market trend for online product purchasing and rapid fulfillment, this automotive retailer is able to fend off the threat of ecommerce competitors and defend their market position as a preferred source of automotive parts and accessories. With rigorous software testing supported by automated tools and test data, the quality of the critical systems needed to support this strategy is assured and the business impact is maximized.