REVOLISM

Apex Code Correction project

COMPANY OVERVIEW

REVOLISM is a recently-launched wellness and weight loss franchise providing clients with a program to revolutionize their metabolism and achieve wellness and weight loss goals.   After unsuccessfully attempting to implement a franchisee management solution with Dynamics, REVOLISM switched to Salesforce.   They originally worked with another Salesforce implementer for 8 months but terminated the relationship after a critical defect was discovered in the solution. REVOLISM then contracted with HigherEchelon to correct the defect.

CHALLENGE	SOLUTION	RESULTS
▪Due to the complexity of the developed code, the customer requested that code refactoring be kept to a minimum to reduce the risk of breaking something elsewhere in the system ▪Sales tax calculations were failing to account for price discounts, resulting in a mistakenly high tax liability for the company ▪Apex unit tests didn't validate any processing logic – they only ran through the code to achieve the required 75% code coverage. This meant that fixes had to be tested manually. ▪Poor error handling resulted in some credit card transaction records not being saved, creating a situation for potential double-charging of the customer ▪Inefficient code caused batch processing to fail due to SOQL limits being exceeded. ▪The customer was told that testing had to be done in the production org because of sandbox limitations.	▪Properly configured sandboxes for development and testing ▪Corrected logic around sales tax calculations and franchisee cut calculations. ▪Modified error handling to allow for saving of as many records as possible during batch processing. ▪In cases where an error occurred during batch processing, added logic to create a Case to handle the issue. ▪Corrected SOQL inefficiencies to ensure governor limits wouldn't be reached during batch processing. ▪Automated testing of processing logic. ▪Updated page layouts and created list views to give the customer easier access to critical information. ▪Added logic to allow for fine-grained monitoring of REST-based interactions between Salesforce and Shipstation	▪Sales tax calculations now function as expected ▪The customer is able to perform testing in a sandbox environment per best practices ▪Almost 200 unique test cases are defined and stored as CSV static resources. These automated test cases allow a full calculation regression test to be done with the click of a button. Customer no longer has to manually test every possible combination of discount, payment option and shipping option. ▪A utility class now exists to allow for the creation of multiple orders at one time for testing, reducing the amount of time needed to set up test scenarios. ▪Batch processing happens without the risk of hitting SOQL governor limits. ▪Customer can quickly and easily verify that interactions with Shipstation are working properly.

More Detailed Info and Quantitative Results

[Color Commentary and Notes About Details of Deal or Implementation]

NOTES

The following steps were taken to automate the testing of calculation logic:
- Created static resources to hold data used during testing
- Created a custom object and related static resource to define inputs and expected results of each individual test case
- Created a test utility class which reads data from the custom object, creates an Opportunity with the given test scenario parameters, simulates a credit card transaction and then validates the results against the expected values

To ensure credit card transaction data is always saved, a series of try-catch blocks was coded.   Each catch block creates a new Case with instructions on what went wrong during processing and on what data needs to be corrected in the transaction, opportunity or shipment record.

Batch processing was modified to use the Apex Database class so that most records would still save successfully even if some records had issues. If any records fail to save, a Case is created with details on what went wrong during processing and what steps should be manually taken to correct the issue.

REST endpoint logging saves the full Request/Response message for each interaction between Salesforce and Shipstation, allowing for the ability to debug any problems that might arise. A daily job regularly purges older records to prevent data storage problems.

QUANTITATIVE RESULTS IF AVAILABLE

- Tests that previously needed to be run manually can all now be run with the click of a button. A typical test case would take 1-2 minutes to set up through the user interface. Since 198 unique cases were identified, this would have meant an effort of 3-6 hours for each round of regression testing.   Instead, all tests can now be run from the Setup area or Developer Console. Total user time required is reduced from 3-6 hours to less than 5 minutes.

Solution Detail

Competitors of Salesforce engaged in sales cycle:	*None identified*
Previous technology replaced by Salesforce:	*MSFT Dynamics*
Salesforce products deployed:	*Sales Cloud, Community Cloud, REST services, Service Cloud*
Customer Business Model (B2B, B2C, or Both)	*B2C*
Salesforce Product features:	*Community Cloud, REST, Cases, Opportunities, custom objects*
If using Service Cloud, list use case (e.g. customer support, call center, field service, telesales, etc.)	*Cases are used for 2 scenarios: 1 - track customer service requests; 2 - notify company users about problems encountered during batch processing*
Integrations:	*Chargent, Shipstation, Metabolic Code*
AppExchange Apps/Partners	*Chargent, Gearset*
Solution 'Go Live' date:	*6/4/2018*

Screenshot – related information section of Opportunity

Screenshot – Case created due to processing failure

Case Detail   Edit | Delete

▼ Case Information

Case Number	00001074	Related Customer	Internal
Case Owner	HQ HQ		
Related Customer's Name			

▼ User Information

User Name			
User Phone			
Subject	Transaction calculation failure	Case Origin	Internal
Description	Failed to calculate Franchisee/HQ cuts for transaction for order #712. Gateway transaction #: [A20A2D566FB6]. Please manually calculate the franchisee/HQ cuts for this transaction. Original error message: Attempt to de-reference a null object Stack trace: Trigger.SetUpRecurringBillingAndShipments: line 290, column 1 Trigger.SetUpRecurringBillingAndShipments: line 111, column 1	Status	New/Escalated
		Priority	High

salesforce

Screenshot – REST endpoint log entry

Shipstation Log Detail

Shipstation Log Name	GET2018-06-04 18:51:32
Request	RestRequest:[headers={Accept=application/json, Accept-Encoding=gzip, CipherSuite=ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 256-bits, Content-Type=application/json; charset=UTF-8, Host=na35.salesforce.com, User-Agent=Workbench/42.0.0, X-PrettyPrint=true, X-Salesforce-SIP=35.168.189.83}, httpMethod=GET, params={}, remoteAddress=35.168.189.83, requestBody=Blob[0], requestURI=/customStore, resourcePath=/services/apexrest/customStore/*]Body:
Response	RestResponse:[headers={}, responseBody=Blob[14], statusCode=404]Body: NOT AUTHORIZED
Part Number	1
Created By	HQ HQ, 6/4/2018 1:51 PM

Sharing Guidelines

These questions are required in order for your story submission to be accepted. Please answer these questions to the best of your ability.

We will not contact the customer without reaching out to you first.

SHARING USE CASE/SCENARIO:	
Can Salesforce AEs share this story and overview slide in sales settings?	Y
Can Salesforce AEs mention the customer name in sales settings?	Y
Is this customer willing to act as a reference customer for prospects?	Y
Would the customer be willing to speak at Dreamforce or other events?	Y

PARTNER INFORMATION	
Name of reference approver:	**Natalya Murphy**
Title of reference approver:	**Sr. Salesforce Developer**
Email of reference approver:	**natalya.murphy@higherechelon.com**

salesforce