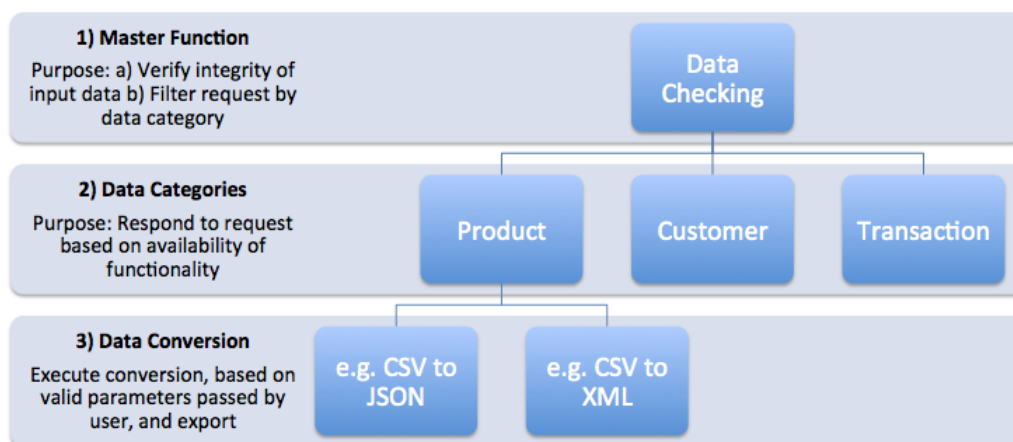


To: ShopKeep
From: Darragh Monaghan
Date: 31/01/2016
RE: Write-up for Coding Challenge – Importer / Exporter Product

Design Concept:

- The product was designed so that users only interact with a single file, importer_exporter.rb, then running the file through the IRB
- It was structured around category of conversion data - Product, Customer, Transaction, which can be easily scaled, as the ShopKeep platform evolves
- With the product running through the Interactive Ruby Shell, clear and concise error messages were developed to ensure maximum ease of use

Exhibit 1: Functionality Matrix – conversions currently possible



Current Functionality:

At present, as instructed, the Importer/Exporter only handles CSV – JSON conversions of Product data.

Exhibit 2: Functionality Matrix – conversions currently possible

Input Format	Export Format	Product	Customer	Transaction
CSV	JSON			
CSV	XML			
XML	JSON			
XML	CSV			
JSON	XML			
JSON	CSV			

Due to the structuring of the product, adding further functionality, either vertically, within the Product data category, or horizontally, covering additional data categories, would only require the bolt on additions and minor changes outlined in exhibit 3 and 4:

Exhibit 3: Adding Conversion Types within the Product Category:

Function / Aspect	Details	Workload (10 = highest)
product_category	Adjust outcome of if / elsif statement to call new function instead of return error message	1 / 10
New conversion function	The function to actually execute the conversion would need developed	5 / 10
Testing	Additional RSpec testing suite required	2 / 10
Support Functions	Potentially require additional support functions to be developed	2 / 10

Exhibit 4: Adding new Data Category Conversions (Customer, Transaction):

Function / Aspect	Details	Workload (10 = highest)
check_input_data	Adjust outcome of if / elsif statement to call new function instead of return error message	1 / 10
New data category function required	New function required to filter the input and output data formats	2 / 10
New conversion function	The function to actually execute the conversion would need developed	5 / 10
Testing	Additional RSpec testing suite required	2 / 10
Support Functions	Potentially require additional support functions to be developed	2 / 10

Assumptions:

As requested, all assumptions made throughout the exercise have been outlined below:

- It is assumed that users of the product possess a base level of technical proficiency to feel comfortable using the command line interface and Ruby Shell
- Assumptions made on the data imported from the sample.csv file are detailed in exhibit 5

Exhibit 5: Assumptions made on data within sample.csv

Area	Details
Item ID:	Will only consist of Integers, never text based characters
Description:	Assumed the numbers found in the item descriptions, e.g. cells B6 – B15, are not required, possibly a data entry error
Price:	Assumed no incorrect data entry would occur in this field, with consistent formatting
Cost:	Entries in the form of Fixnum are invalid, this was decided due to their correlation with the errors believed to exist in their corresponding field in column B, 'Description', seen above
Price Type:	Assumed that no data entry errors would occur in this field
Quantity on Hand:	Assumed that no data entry errors would occur in this field
Modifiers:	Assumed a maximum of 3 modifiers would exist, each with 2 sets of key value pairs, name and price

Further Information:

Further information on the Importer/Exporter product can be requested from the developer, and author of this report, Darragh Monaghan monaghan.darragh@gmail.com