|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BILL TO** |  |  | **Work Package Number:** | BVOX01 |
| **BRAINTREE BY VOX TELECOM** |  |  | **Package Date:** | 30 Oct. 18 |
| Avril Howes  BLOCK D1, RUTHERFORD BUSINESS PARK 1 SCOTT STREET, WAVERLEY JOHANNESBURG, Gauteng  South Africa |  |  | **Estimate Number:** | ESTIMATE 25 |

This document contains the work package information required to develop for following for Dental Warehouse on behalf of Braintree by VOX:

1. BIDVEST INTEGRATION
2. ORDERWISE INTEGRATION
3. WEBSITE INTEGRATION

BIDVEST INTEGRATION

This section describes the requirements for the Bidvest integration to be implemented from Microsoft Dynamics NAV.

Bidvest require that their platform receive following XML files from Microsoft Dynamics NAV:

* INVOICES
* CREDIT NOTES
* STATEMENTS

The following points describe what will be developed into Microsoft Dynamics NAV:

* Create a new codeunit called “DW.BIDVEST.IntegrationMgt” that performs the following:
  + Listens for Invoice creation
  + Listens for Credit Memo creation
  + Saves export XMLPort information to a desired SFTP Location based on setup.
* Create a new setup table called “DW.IntegrationSetup” to store any parameter needed such as SFTP Location and credentials.
* Update the statement report to add an options flag to export to Bidvest. This flag is to be used to submit statement xml information.
* Create a new xmlport called “DW.BIDVEST.InvoiceXML” with a schema to export out information presented on the Sales Invoice report.
* Create a new xmlport called “DW.BIDVEST.CreditMemoXML” with a schema to export out information presented on the Sales Credit Memo report.
* Create a new xmlport called “DW.BIDVEST.StatementXML” with a schema to export out information presented on the Statement report.

*Additional Notes:*

*IP: 41.170.90.242*

*PORT:22*

*U:DentalWH*

*P: VkZ38vpr9CUG*

ORDERWISE INTEGRATION

This section describes the requirements for the orderwise integration to be implemented from Microsoft Dynamics NAV.

Integration is to be performed based on the documentation outlined ‘Order Logistics – Supplier Integration – Rest API’



The following points describe what will be developed into Microsoft Dynamics NAV:

* Create a new codeunit called “DW.ORDERWISE.IntegrationMgt” that perform the following functions:
  + Receive and Process orders. This process will query the Rest API for any new orders and create the documents in NAV. Order line need only to be created where there is stock and generate a response file back to the Rest API once completed.
  + Export Price Lists. This function will export and submit price lists to Orderwise SFTP location where there have been modifications to flagged customers.
* Update the customer table to include a new field called ‘Submit Prices to Orderwise’. This is used to flag medicross, netcare and intercar customer accounts for orderwise pricing.
* Create a new xmlport called “DW.ORDERWISE.PriceExport” an example of this layout can be found below:  
   
* Add fields to the table “DW.IntegrationSetup” to contain FTP path information as well as RestAPI paths for modes such as development and live.
* Create a new codeunit called “DW.ORDERWISE.JobQueue”. This codeunit will link into Microsoft Dynamics NAV’s task scheduler system to check for new incoming orders.

WEBSITE INTEGRATION

This section describes the requirements for the Streetbyte website integration to be implemented from Microsoft Dynamics NAV.

Integration is to be performed based on the outlined [http://woocommerce.github.io/woocommerce-rest-api-docs/#introduction](https://protect-za.mimecast.com/s/wr_QCJZ13YHpDQqXfVV2YW?domain=woocommerce.github.io) documentation.

The following points describe what will be developed into Microsoft Dynamics NAV:

* Create a new codeunit called “DW.WEBSITE.IntegrationMgt” that performs the following functions:
  + Add a trigger the insertion of customer ledger entries. When a customer entry of document type invoice is created this information is to be sent to the RestAPI provided.
  + UpdateWebsite. This function will collect new information gathered by the Microsoft Dynamics change log to send to the Woo integration.
  + PullNewOrders. This function retrieves all new orders to be created in Microsoft Dynamics NAV.
* Add a new field to the sales header table called “WEBSITE ORDER GUID”. This field is used to track orders from the website and ensure that duplicates are not created.
* Add a new codeunit called “DW.WEBSITE.JobQueue” that will periodically monitor new changes from the change log to update the website with any new information.
* The woo integration has many entities that require Microsoft Dynamics NAV to sent to. Each entities table must be setup on the Microsoft Dynamics NAV change log to track changes to information in these tables.
* Add fields to the table “DW.IntegrationSetup” to contain RestAPI paths and authentication.
* The following entity to table mapping is provided below:

<HI MALCOM, PLEASE CAN YOU UPDATE THE TABLE BELOW. IF YOU NEED WE CAN MAP THIS TOGETHER OVER A SKYPE CALL.>

|  |  |
| --- | --- |
| **WOO ENTITY NAME** | **MICROSOFT DYNAMICS NAV TABLE** |
|  |  |
|  |  |
|  |  |
|  |  |