TXWC Voucher Code changes :-

Service folder

1. Blank Operation

In Blank Operation for TXWC voucher the one method added to open the voucher redemption page which name is frmTXWCCashVoucher

Method description: -

The operation id 10031 is hard-core for TXWC voucher , the customer validation is required because in Voucher redemption and balance the customer record mandatory

if (operationInfo.OperationId == "10031") // For TXWC cash vocuher added by amandeep sharma 10/08/2021

{

RetailTransaction retailtransaction = posTransaction as RetailTransaction;

string customeremail = "";

if (retailtransaction == null)

{

using (LSRetailPosis.POSProcesses.frmMessage dialog = new LSRetailPosis.POSProcesses.frmMessage("Please Add the item lines", MessageBoxButtons.OK, MessageBoxIcon.Information))

{

LSRetailPosis.POSProcesses.POSFormsManager.ShowPOSForm(dialog);

}

return;

}

if (retailtransaction.SaleItems.Count <=0 )

{

using (LSRetailPosis.POSProcesses.frmMessage dialog = new LSRetailPosis.POSProcesses.frmMessage("Please Add the item lines", MessageBoxButtons.OK, MessageBoxIcon.Information))

{

LSRetailPosis.POSProcesses.POSFormsManager.ShowPOSForm(dialog);

}

return;

}

if (retailtransaction.NetAmountWithTax <= 0)

{

using (LSRetailPosis.POSProcesses.frmMessage dialog = new LSRetailPosis.POSProcesses.frmMessage("Please Add the item lines", MessageBoxButtons.OK, MessageBoxIcon.Information))

{

LSRetailPosis.POSProcesses.POSFormsManager.ShowPOSForm(dialog);

}

return;

}

if (string.IsNullOrEmpty(retailtransaction.Customer.MobilePhone))

{

using (LSRetailPosis.POSProcesses.frmMessage dialog = new LSRetailPosis.POSProcesses.frmMessage("Please Add the Customer Mobile No", MessageBoxButtons.OK, MessageBoxIcon.Information))

{

LSRetailPosis.POSProcesses.POSFormsManager.ShowPOSForm(dialog);

}

return;

}

#region "getting email address"

System.Data.SqlClient.SqlDataAdapter sqlda;

System.Data.SqlClient.SqlCommand cmd;

System.Data.DataTable dtCustList;

string strQryCusAccNo = "Select Email1 from [ax].[ACXCUSTTABLE] Where CUSTACCOUNT = '" + retailtransaction.Customer.MobilePhone + "' ";

cmd = new System.Data.SqlClient.SqlCommand(strQryCusAccNo, LSRetailPosis.Settings.ApplicationSettings.Database.LocalConnection);

sqlda = new System.Data.SqlClient.SqlDataAdapter(cmd);

dtCustList = new System.Data.DataTable();

sqlda.Fill(dtCustList);

if (dtCustList.Rows.Count > 0)

{

customeremail = dtCustList.Rows[0]["EMail1"].ToString();

if (string.IsNullOrEmpty(customeremail))

{

using (LSRetailPosis.POSProcesses.frmMessage dialog = new LSRetailPosis.POSProcesses.frmMessage("Please Add the Customer Email Address", MessageBoxButtons.OK, MessageBoxIcon.Information))

{

LSRetailPosis.POSProcesses.POSFormsManager.ShowPOSForm(dialog);

}

return;

}

}

else

{

using (LSRetailPosis.POSProcesses.frmMessage dialog = new LSRetailPosis.POSProcesses.frmMessage("Please Add the Customer Email Address", MessageBoxButtons.OK, MessageBoxIcon.Information))

{

LSRetailPosis.POSProcesses.POSFormsManager.ShowPOSForm(dialog);

}

return;

}

#endregion

frmTXWCCashVoucher \_cashvocuher = new frmTXWCCashVoucher(retailtransaction, Application, customeremail);

\_cashvocuher.ShowDialog();

}

In the Property class folder the four Classes are added

1. ResponseClass:--this class is used to get the API calling Status and response
2. BalanceEnquiryResponse:-- this class is used to convert the json response which is getting from the Voucher balance inquiry into model
3. VocuherRedeemResponse :--this class is used to convert the json response which is getting from the After Voucher Redemption into model
4. CaptureVoucherDetails:-- this class used to capture the voucher details which is send in post api of voucher redemption
5. frmTXWCCashVoucher Page:--

This is new page and in which the balance enquiry and apply voucher url and code written and also validation added.

C.Printing:-

In printing the method name:-

public bool PrintReceipt(FormType formType, IPosTransaction posTransaction, bool copyReceipt)

The query is changed in the method to get the tender details

Old query:--

string Footerqry = @"Select CASE WHEN RTPT.TENDERTYPE=2 AND RTPT.CARDTYPEID<>'' THEN ISNULL('CARD-'+ RSTC.CardName,'')

WHEN RTPT.TENDERTYPE=2 AND RTPT.CARDTYPEID='' THEN 'CARD'

ELSE RST.NAME

END AS [TENDERNAME], RTPT.AMOUNTTENDERED AS [AMOUNT], RTPT.CARDTYPEID from AX.RETAILTRANSACTIONPAYMENTTRANS RTPT "

+ " Inner Join (Select distinct TENDERTYPEID,NAME,DATAAREAID from RETAILSTORETENDERTYPETABLE) RST on RTPT.TENDERTYPE=RST.TENDERTYPEID and RTPT.DATAAREAID=RST.DATAAREAID "

+ " LEFT OUTER join (Select Distinct TENDERTYPEID,CARDTYPEID,NAME AS CardName, DATAAREAID from ax.RETAILSTORETENDERTYPECARDTABLE) RSTC ON RSTC.TENDERTYPEID = RST.TENDERTYPEID and RTPT.CARDTYPEID=RSTC.CARDTYPEID AND RSTC.DATAAREAID = RTPT.DATAAREAID "

+ " WHERE RTPT.AMOUNTTENDERED > 0 AND RTPT.TRANSACTIONID = '" + posTransaction.TransactionId.ToString() + "' ";

New Query:-

string Footerqry = @"Select RTPT.TENDERTYPE, CASE WHEN RTPT.TENDERTYPE = 1 THEN 'CASH'

WHEN RTPT.TENDERTYPE=2 AND RTPT.CARDTYPEID = '' THEN ISNULL('CARD (REF ID: '+ RTPT.CARDORACCOUNT+')','')

WHEN RTPT.TENDERTYPE=2 AND RTPT.CARDTYPEID = '98' THEN ISNULL(RSTC.CardName + ' (REF ID: ' +RTT.ACX\_WalletSystemTxnID+')','')

WHEN RTPT.TENDERTYPE=2 AND RTPT.CARDTYPEID = '101' THEN ISNULL(RSTC.CardName + ' (REF ID: ' +RTT.ACX\_AIRTELTRANSID+')','')

WHEN RTPT.TENDERTYPE=2 AND RTPT.CARDTYPEID = '102' THEN ISNULL(RSTC.CardName + ' (REF ID: ' +RTT.ACX\_UPITXNID+')','')

WHEN RTPT.TENDERTYPE=2 AND RTPT.CARDTYPEID = '98' THEN ISNULL(RSTC.CardName + ' (REF ID: ' +RTT.Acx\_PhonePeReferenceID+')','')

WHEN RTPT.TENDERTYPE=4 THEN ISNULL(RST.NAME + ' (CR NO: ' +RTPT.CREDITVOUCHERID+')','')

WHEN RTPT.TENDERTYPE=11 THEN ISNULL(RSTC.CardName + ' (VOUCHER ID: ' +RTT.ACX\_CASHVOUCHERID+')','')

WHEN RTPT.TENDERTYPE=15 THEN ISNULL('TXWC VOUCHER:'+RTPT.DESCRIPTION,'')

ELSE RST.NAME END AS [TENDERNAME], RTPT.AMOUNTTENDERED AS [AMOUNT] from AX.RETAILTRANSACTIONPAYMENTTRANS RTPT

LEFT OUTER JOIN AX.RetailTransactionTable RTT ON RTT.RECEIPTID = RTPT.RECEIPTID

Inner Join (Select distinct TENDERTYPEID,NAME,DATAAREAID

from RETAILSTORETENDERTYPETABLE) RST on RTPT.TENDERTYPE=RST.TENDERTYPEID and RTPT.DATAAREAID=RST.DATAAREAID

LEFT OUTER join (Select Distinct TENDERTYPEID,CARDTYPEID,NAME AS CardName, DATAAREAID from ax.RETAILSTORETENDERTYPECARDTABLE) RSTC

ON RSTC.TENDERTYPEID = RST.TENDERTYPEID and RTPT.CARDTYPEID=RSTC.CARDTYPEID AND RSTC.DATAAREAID = RTPT.DATAAREAID

WHERE RTPT.AMOUNTTENDERED > 0 AND RTPT.TRANSACTIONID = '" + posTransaction.TransactionId.ToString() + "' ";

TRIGGERS FOLDER

A.Transaction Triggers:-

This list is declare in triggers code of transaction and the data in list is added when we apply the voucher on transaction,and the applied voucher details are added in TXWCVoucherdata model class on service level code and on the postend of transaction the data are saved in the tables

public static List<TXWCVoucherData> listtxwcvocuherdata = new List<TXWCVoucherData>();

New code added in PostendTransaction:--

public void PostEndTransaction(IPosTransaction posTransaction)

{

#region "Voucher details added"

if (listtxwcvocuherdata.Count > 0)

{

try

{

SqlConnection connection = LSRetailPosis.Settings.ApplicationSettings.Database.LocalConnection;

string UpdateOfferCode = string.Empty;

if (connection.State == ConnectionState.Closed)

{

connection.Open();

}

foreach (var data in listtxwcvocuherdata)

{

string qurery = "UPDATE ax.RETAILTRANSACTIONPAYMENTTRANS SET [PAYMENTAUTHORIZATION]= '" + data.PAYMENTCAPTUREID + "'," + Environment.NewLine;

qurery += "[AUTHENTICATIONCODE]= '" + data.AUTHENTICATIONCODE + "'" + Environment.NewLine;

qurery += " Where LINENUM='" + data.LINENUM + "' AND TENDERTYPE=15 AND DESCRIPTION='" + data.VOUCHERID + "' AND TRANSACTIONID='" + posTransaction.TransactionId.ToString() + "' ";

using (SqlCommand sqlCmd = new SqlCommand(qurery, connection))

{

sqlCmd.ExecuteNonQuery();

}

}

listtxwcvocuherdata = new List<TXWCVoucherData>();

string updQuery = "UPDATE ax.RETAILTRANSACTIONTABLE SET [Acx\_WEBORDERREFER]= 'TXWC'" + Environment.NewLine;

updQuery += " Where TRANSACTIONID='" + posTransaction.TransactionId.ToString() + "' ";

using (SqlCommand sqlCmd = new SqlCommand(updQuery, connection))

{

sqlCmd.ExecuteNonQuery();

}

}

catch (Exception ee)

{

using (LSRetailPosis.POSProcesses.frmMessage dialog = new LSRetailPosis.POSProcesses.frmMessage(ee.Message.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Information))

{

LSRetailPosis.POSProcesses.POSFormsManager.ShowPOSForm(dialog);

}

}

}

#endregion

}