1. **Tại quầy**
2. **Tại quầy – Tạo giao dịch**



**Step 3:** Gọi Service để lấy thông tin khách hàng:

* Gọi Service1.getCustomerFromDienLuc() → Gọi Service EVNHCM\_SVC.LVBService.BankRequest(makh, madiemthu, BankID)

Return:

dt[khachhang]

* + - Makh, tenkh, diachikh, madl, masothuekh

dt[hoadon]

* + - mahd, sotien, mota, tungay, denngay, giabieu, hoadonid, soho, magiaodich, DNTT, tiendien, tienthue, thuesuat, kyhieuhd, seryhd, HDDT, NCC

**Step** 13: Gọi Service để lấy thông tin tài khoản điện lực

* Gọi Service Service1.getCustAccFromFlexCube\_Acc(Account\_No)

**Step 15:** Lấy thông tin tài khoản điện lực

* dt[Account]
  + Select from table: sttm\_cust\_account c
  + Column: c.CUST\_AC\_NO, c.AC\_DESC, c.ACY\_AVL\_BAL, c.BRANCH\_CODE, BRANCH\_NAME, c.CUST\_NO
  + Where: c.CR\_GL like '4211%' OR C.cust\_no in ( select B.WALKIN\_CUSTOMER from sttm\_branch b where B.RECORD\_STAT = 'O') and c.RECORD\_STAT = 'O' and c.CUST\_AC\_NO = '" + Account\_No + "'
* sCIF = dt.Rows[0]["CUST\_NO"].ToString();
* dt[Customer]
  + Select from table: sttm\_customer " + DB\_Link + "s left join sttm\_cust\_personal p on s.customer\_no = p.customer\_no
  + Where: s.CUSTOMER\_NO = '" + sCIF + "' and record\_stat = 'O'

**Step 20:** Gọi Service để lấy thông tin tài khoản khách hàng

* Gọi Service Service1.getCustAccFromFlexCube(Customer\_No)

**Step 22:** Lấy thông tin tài khoản khách hàng:

* dt[customer]:
  + Select from table: sttm\_customer s left join sttm\_cust\_personal p on s.customer\_no = p.customer\_no
  + Column: s.CUSTOMER\_NO,s.CUSTOMER\_NAME1, s.UNIQUE\_ID\_VALUE,s.UDF\_1 NGAY\_CAP,s.UDF\_2 NOI\_CAP, P.TELEPHONE, s.ADDRESS\_LINE1 || s.ADDRESS\_LINE2 || s.ADDRESS\_LINE3 || s.ADDRESS\_LINE4 ADDRESS
  + Where: s.CUSTOMER\_NO = '" + Customer\_No + "' and record\_stat = 'O'
* dt[account]:
  + Select from table: sttm\_cust\_account c
  + Column: c.CUST\_AC\_NO,c.AC\_DESC,c.BRANCH\_CODE, ACY\_AVL\_BAL
  + Where: c.ACCOUNT\_CLASS IN ('V0CN30', 'V0TKCN') OR c.CR\_GL LIKE '4211%' OR c.cust\_no IN (SELECT B.WALKIN\_CUSTOMER FROM sttm\_branch b WHERE B.RECORD\_STAT = 'O')) AND c.RECORD\_STAT = 'O' AND c.CUST\_NO ='" + Customer\_No + "' " AND c.ACCOUNT\_CLASS IN ('V0CN30', 'V0CD26', 'V0CD45', 'V0CNTN', 'V0DNTN', 'V0ID44', 'VMWALK', 'V0TKCN') AND c.CCY = 'VND'"

**Step 28:** Gọi Service tạo giao dịch

* Gọi Service Service1.importTransaction()

**Step 30:** Lưu thông tin giao dịch

* Lấy số hóa đơn trong hệ thống, thêm hóa đơn vào database

Package: pkg\_EVN\_HCM.dlf\_getMaGiaoDich

Parameter:

p\_Maker\_ID  
 p\_HoaDon\_Info xmltype

Variables:

v\_Rcd\_HD  
 v\_Home\_Branch  
 v\_Sign  
 v\_Year  
 v\_Min\_ID  
 v\_Max\_ID  
 v\_current\_id  
 v\_Rcd\_Err  
 v\_Sys\_Date

* For r in xmltable('/HoaDon/HD\_Info')
  + if r.HDDT = 1
    - Insert into DLTBD\_HOADON\_INFO
      * Column: MAHD, MAKH, SOTIEN, MOTA, TUNGAY, DENNGAY, GIABIEU, HOADONID, SOHO, NGAYGIO, MAGIAODICH, BANKID, DAINHD, TRN\_CHANEL, YEAR, HOADON\_ID, DNTT, TIENDIEN, TIENTHUE, THUESUAT, KYHIEUHD, SERYHD, HDDT
      * Values: r.MAHD, trim(r.MAKH), r.SOTIEN, r.MOTA, r.TUNGAY, r.DENNGAY, r.GIABIEU, r.HOADONID, r.SOHO, v\_Sys\_Date, r.MaGiaoDich, c\_Bank\_ID, '1', c\_Chanel\_TELLER, v\_Year, SUBSTR(r.MaGiaoDich,1,8), r.DNTT, r.TIENDIEN, r.TIENTHUE, r.THUESUAT, r.KYHIEUHD, r.SERYHD,r.HDDT
      * Returning: rowid, MAHD, MAKH, SOTIEN, HOADONID, MAGIAODICH

into v\_Rcd\_HD.row\_id, v\_Rcd\_HD.MAHD, v\_Rcd\_HD.MAKH, v\_Rcd\_HD.SOTIEN, v\_Rcd\_HD.HOADONID, v\_Rcd\_HD.MAGIAODICH;

* + if r.MaHD like '%2'
    - Insert into DLTBD\_HOADON\_INFO
      * Column: MAHD, MAKH, SOTIEN, MOTA, TUNGAY, DENNGAY, GIABIEU, HOADONID, SOHO, NGAYGIO, BANKID, DAINHD, TRN\_CHANEL, DNTT, TIENDIEN, TIENTHUE, THUESUAT, KYHIEUHD, SERYHD, HDDT
      * Values: r.MAHD, trim(r.MAKH), r.SOTIEN,r.MOTA, r.TUNGAY,r.DENNGAY,r.GIABIEU,r.HOADONID,r.SOHO, v\_Sys\_Date, c\_Bank\_ID, '1', c\_Chanel\_TELLER, r.DNTT, r.TIENDIEN, r.TIENTHUE, r.THUESUAT, r.KYHIEUHD, r.SERYHD, r.HDDT
      * Returning: rowid, MAHD, MAKH, SOTIEN, HOADONID

into v\_Rcd\_HD.row\_id, v\_Rcd\_HD.MAHD, v\_Rcd\_HD.MAKH, v\_Rcd\_HD.SOTIEN, v\_Rcd\_HD.HOADONID;

* + Else
    - Insert into DLTBD\_HOADON\_INFO
      * Column: MAHD, MAKH, SOTIEN, MOTA, TUNGAY, DENNGAY, GIABIEU, HOADONID, SOHO, NGAYGIO, MAGIAODICH, BANKID,DAINHD, TRN\_CHANEL, YEAR, HOADON\_ID, DNTT, TIENDIEN, TIENTHUE, THUESUAT, KYHIEUHD, SERYHD, HDDT
      * Value: r.MAHD, trim(r.MAKH), r.SOTIEN,r.MOTA, r.TUNGAY,r.DENNGAY,r.GIABIEU,r.HOADONID,r.SOHO, v\_Sys\_Date, lpad(v\_Rcd\_HD.HOADON\_ID,10, '0') || v\_Sign, c\_Bank\_ID, '1', c\_Chanel\_TELLER, v\_Year, v\_Rcd\_HD.HOADON\_ID, r.DNTT,r.TIENDIEN, r.TIENTHUE, r.THUESUAT, r.KYHIEUHD, r.SERYHD, r.HDDT
      * Returning: rowid, MAHD, MAKH, SOTIEN, HOADONID, MAGIAODICH

into v\_Rcd\_HD.row\_id, v\_Rcd\_HD.MAHD, v\_Rcd\_HD.MAKH, v\_Rcd\_HD.SOTIEN, v\_Rcd\_HD.HOADONID, v\_Rcd\_HD.MAGIAODICH;

Return: v\_Rcd\_HD;

* Gọi package pkg\_EVN\_HCM.DLP\_importTransaction

Parameters:

p\_Maker\_ID  
 p\_Trn\_Desc  
 p\_KhachHang\_Info  
 p\_HoaDon\_Info  
 p\_HachToan\_Info  
 p\_Trn\_ID  
 p\_Err\_String

Variables:

v\_Home\_Branch  
 v\_Trn\_Dt  
 v\_Check  
 v\_Current\_ID  
 v\_Min\_ID  
 v\_Max\_ID  
 V\_Branch\_Code  
 V\_BRANCH  
 V\_COUNT

* Insert into DATBD\_TRANSACTION
  + Column: TRANSACTION\_ID, TRN\_BRN, TRN\_DESC, TRN\_DT, VALUE\_DT, TRN\_CODE, MODULE, EVENT\_CODE, RECORD\_STATUS, MAKER\_ID, MAKER\_DT, APP\_TYPE, LAST\_EVENT\_SEQ\_NO, XREF, COMMON\_USER
  + Vaulues: p\_Trn\_ID, v\_Home\_Branch, p\_Trn\_Desc, v\_Trn\_Dt, trunc(sysdate), 'LIQUID', 'DL', 'INIT', 'O', p\_Maker\_ID, sysdate, 'DOM\_WEB', 1, 'EBK'||loadESBMsgId(), c\_Common\_User
* For r in xmltable ('/Transaction/Post\_Info')
  + IF r.Ac\_No IS NULL //TH giao dịch TM
    - Insert into DLTBD\_TRANSACTION\_POST
    - Column: TRANSACTION\_ID, AC\_NO, AC\_CCY, AC\_BRANCH, CUST\_GL, DRCR\_IND, AMOUNT, AMOUNT\_TAG, TRN\_CODE, EVENT, EVENT\_SEQ\_NO
    - Values: p\_Trn\_ID, '101100001','VND', v\_Home\_Branch, 'G', R.DrCr\_Ind, R.Amount, R.Amount\_Tag,'LIQUID','INIT', 1
  + Else //TH giao dịch CK
    - Select from table STTM\_CUST\_ACCOUNT
      * Column: BRANCH\_CODE INTO V\_BRANCH
      * Where: CUST\_AC\_NO = R.AC\_NO AND RECORD\_STAT = 'O' AND AUTH\_STAT ='A';
    - Insert into table DLTBD\_TRANSACTION\_POST
      * Columns: TRANSACTION\_ID, AC\_NO, AC\_CCY, AC\_BRANCH, CUST\_GL, DRCR\_IND, AMOUNT, AMOUNT\_TAG, TRN\_CODE, EVENT, EVENT\_SEQ\_NO
      * Values: p\_Trn\_ID, r.Ac\_No,'VND', V\_BRANCH, 'A', R.DrCr\_Ind, R.Amount, R.Amount\_Tag, 'LIQUID','INIT', 1
* For r in xmltable (' /KhachHang')
  + Update table DLTB\_CUSTOMER\_INFO
    - Set: TENKH = r.TENKH,  
      DIACHIKH = r.DIACHIKH,  
      MADL = r.MADL  
      MASOTHUEKH = r.MASOTHUE  
      PHIEN = r.PHIEN  
      LOTRINH = r.LOTRINH  
      SOGHICSCMIS = r.SOGHICSCMIS  
      DANHSO = r.DANHSO  
      SOCONGTO = r.SOCONGTO  
      NGANHNGHE = r.NGANHNGHE
    - Where: MAKH = r.MAKH;
  + Nếu update ko được
    - Insert into table DLTB\_CUSTOMER\_INFO
      * Columns: MAKH, TENKH, DIACHIKH, MADL, MASOTHUEKH, PHIEN, LOTRINH, SOGHICSCMIS, DANHSO, SOCONGTO, NGANHNGHE
      * Values: trim(r.MAKH), r.TENKH, r.DIACHIKH, r.MADL, r.MASOTHUE, r.PHIEN, r.LOTRINH, r.SOGHICSCMIS, r.DANHSO, r.SOCONGTO, r.NGANHNGHE
* For r in xmltable (' /HoaDon/HD\_Info')
  + Update table DLTBD\_HOADON\_INFO h
    - Set: H.TRN\_ID = p\_Trn\_ID
    - Where: rowid = r.row\_id;

1. **Duyệt đơn**  
   

**Step 2:** Gọi Service lấy list giao dịch cần duyệt

* Gọi Service1. getTransactionsToAuth()

**Step 4:** Select list giao dịch cần duyệt

* Select from table: DATBD\_TRANSACTION d, STTM\_DATES" + DB\_Link + " e "
  + Columns: d.\*, NCC, TRN\_CODE\_NAME
  + Where: D.TRN\_BRN = '" + Branch\_Code + "'

and D.EVENT\_CODE = 'INIT'   
and d.MODULE in ('DL', 'EVNHCM', 'EVNCPC', 'EVNSPC', 'EVNHNI','EVNNPC')   
and d.TRN\_CODE = 'LIQUID'   
and D.CHECKER\_ID is null   
and D.RECORD\_STATUS = 'O'   
and D.AUTH\_STATUS is null   
and D.TRN\_DT = E.TODAY   
and D.TRN\_BRN = E.BRANCH\_CODE   
and D.MAKER\_DT >= SYSDATE - " + sDATE

* + Order by: D.TRANSACTION\_ID DESC
  + Return: dt

NCC = case d.MODULE WHEN 'DL' THEN 'EVNHCM' ELSE d.MODULE END

TRN\_CODE\_NAME = decode (( select ac\_no from dltbd\_transaction\_post where transaction\_id = d.transaction\_id and drcr\_ind = 'D' and event\_seq\_no =1), '101100001', 'Tiền mặt', 'Chuyển khoản')

**Step 8:** Gọi Service lấy chi tiết giao dịch

* Gọi Service1.getTransactionDetail(Trn\_ID)

**Step 10:** Select chi tiết giao dịch

* Select from table DATBD\_TRANSACTION d, DLTBD\_TRANSACTION\_POST p
  + Columns: p.\*, d.MAKER\_ID, d.TRN\_DT, d.TRN\_BRN, d.TRN\_CODE, d.MODULE, NAME
  + Where: to\_char(d.TRANSACTION\_ID) = " + Trn\_ID + "   
    and d.RECORD\_STATUS = 'O'   
    and NVL(p.TRN\_STATUS,'O') <> 'C'   
    and d.TRANSACTION\_ID = p.TRANSACTION\_ID   
    and d.LAST\_EVENT\_SEQ\_NO = p.EVENT\_SEQ\_NO
  + Order by: d.TRANSACTION\_ID

NAME = SELECT NAME FROM ASTB\_USER WHERE USERNAME = d.MAKER\_ID

**Step 14:** Gọi Service duyệt giao dịch

* Gọi Service1.authTransaction()

**Step 16 + 18:**

Gọipkg\_EVN\_HCM.DLP\_authTransaction

Parameter:

p\_Checker\_ID  
p\_Trn\_ID  
p\_NgayGio  
p\_Err\_String

Constant:

c\_DB\_Link  
 c\_txn\_code  
 c\_Function\_Name

Variable:

v\_Home\_Branch  
 v\_Trn\_Desc  
 v\_Maker\_ID  
 v\_Rcd\_Err  
 v\_Value\_Date  
 v\_Last\_Event\_Seq\_No  
 v\_MSGID  
 v\_Rcd\_Trn

1. Get Transaction to Authorize

* Update table DATBD\_TRANSACTION d
  + Set: d.AUTH\_STATUS = 'A',  
     d.CHECKER\_DT = sysdate,  
     d.CHECKER\_ID = p\_Checker\_ID
  + Where: D.TRANSACTION\_ID = p\_Trn\_ID   
     and D.TRN\_BRN = v\_Home\_Branch   
     and D.EVENT\_CODE = 'INIT'  
     and D.RECORD\_STATUS = 'O'   
     and D.AUTH\_STATUS is null   
     and D.CHECKER\_ID is null
* Select from table DATBD\_TRANSACTION d
  + Columns: d.TRN\_DESC, d.MAKER\_ID, d.LAST\_EVENT\_SEQ\_NO
  + Where: D.TRANSACTION\_ID = p\_Trn\_ID   
     and D.TRN\_BRN = v\_Home\_Branch   
     and D.EVENT\_CODE = 'INIT'  
     and D.RECORD\_STATUS = 'O'   
     and D.AUTH\_STATUS is null   
     and D.CHECKER\_ID is null
  + Returning into: v\_Trn\_Desc, v\_Maker\_ID, v\_Last\_Event\_Seq\_No

1. Process authorize

Variable:

v\_lc\_SQL\_CMD  
 v\_lc\_Trn\_Ref\_No  
 v\_lc\_source\_code

v\_lc\_Cr\_Ac\_No  
 v\_lc\_Cr\_Ac\_Branch  
 v\_lc\_Cr\_Ac\_Ccy  
 v\_lc\_Cr\_Amount

v\_lc\_Db\_Ac\_No  
 v\_lc\_Db\_Ac\_Branch

v\_lc\_Db\_Ac\_Ccy  
 v\_lc\_DB\_Amount

* Get post info

//Tài khoản ghi nợ

* + Select from table DLTBD\_TRANSACTION\_POST p
    - Columns: P.AC\_NO,P.AC\_BRANCH,P.AC\_CCY,P.AMOUNT
    - Returning into: v\_lc\_Db\_Ac\_No,v\_lc\_Db\_Ac\_Branch, v\_lc\_Db\_Ac\_Ccy, v\_lc\_DB\_Amount

//Tài khoản ghi có

* + Select from table DLTBD\_TRANSACTION\_POST p
    - Columns: P.AC\_NO,P.AC\_BRANCH,P.AC\_CCY,P.AMOUNT
    - Returning into v\_lc\_Cr\_Ac\_No,v\_lc\_Cr\_Ac\_Branch, v\_lc\_Cr\_Ac\_Ccy, v\_lc\_Cr\_Amount
* Post to CoreBanking
  + if v\_lc\_Db\_Ac\_No like '1011%' //TM

v\_lc\_source\_code := 'THCK';

v\_lc\_SQL\_CMD:= 'select '|| '''' || v\_lc\_Cr\_Ac\_No || '''' || ','

|| '''' || v\_lc\_Cr\_Ac\_Branch || '''' || ','  
 || '''' || v\_lc\_Db\_Ac\_No || '''' || ','  
 || '''' || v\_lc\_Db\_Ac\_Branch || '''' || ',';

v\_lc\_SQL\_CMD:= v\_lc\_SQL\_CMD || '''' || v\_lc\_Db\_Ac\_Ccy || '''' || ','

|| v\_lc\_DB\_Amount || ','

|| '''' || v\_Trn\_Desc || '''' || ','

|| '''' || v\_lc\_source\_code || '''' || ','

|| '''' || c\_txn\_code || '''' || ','

|| '''' || upper(v\_Maker\_ID) || '''' || ','

|| '''' || c\_Common\_User || '''' || ' '

|| 'from dual '

|| 'where rownum = 1 ';

* + else //CK

v\_lc\_source\_code := 'CKCK';

v\_lc\_SQL\_CMD:= 'select '|| '''' || v\_lc\_Db\_Ac\_No || '''' || ','

|| '''' || v\_lc\_Db\_Ac\_Branch || '''' || ','

|| '''' || v\_lc\_Cr\_Ac\_No || '''' || ','

|| '''' || v\_lc\_Cr\_Ac\_Branch || '''' || ',';

v\_lc\_SQL\_CMD:= v\_lc\_SQL\_CMD || '''' || v\_lc\_Db\_Ac\_Ccy || '''' || ','

|| v\_lc\_DB\_Amount || ','

|| '''' || v\_Trn\_Desc || '''' || ','

|| '''' || v\_lc\_source\_code || '''' || ','

|| '''' || c\_txn\_code || '''' || ','

|| '''' || null || '''' || ','

|| '''' ||c\_Common\_User || '''' || ' '

|| 'from dual '

|| 'where rownum = 1 ';

* Gọi package ebank.PKG\_CORE\_SWITCH.PR\_RT\_TRANSFER\_V3 post to CoreBanking
* If substr(v\_Rcd\_Err.MESSAGE,3,2) <> '00'
  + if (substr(v\_Rcd\_Err.MESSAGE,3,2) = '80' or substr(v\_Rcd\_Err.MESSAGE,3,2) = '90' or substr(v\_Rcd\_Err.MESSAGE,3,2) = '91')
    - Update table: DLTBD\_TRANSACTION\_POST p
      * Set: P.TRN\_STATUS = 'S',  
         P.POST\_DATE = sysdate,  
         P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)
    - Update table: DLTBD\_HOADON\_INFO b
      * Set: B.NGAYGIO = to\_date(p\_NgayGio, 'mm/dd/yyyy hh:mi:ss am')  
         B.Time\_Out = v\_Rcd\_Trn.XREF
    - v\_Rcd\_Err.ERROR\_CODE := '1'||substr(p\_Err\_String,3,2);
    - v\_Rcd\_Err.ERROR\_TYPE := pkg\_Dom\_Util.c\_gl\_Debug\_Msg;
  + else
    - v\_Rcd\_Err.ERROR\_CODE := '0099';
    - v\_Rcd\_Err.ERROR\_TYPE := pkg\_Dom\_Util.c\_gl\_Sys\_Err;
* else
  + update table: datbd\_transaction d
    - Set: d.AUTH\_STATUS = 'A',  
       d.CHECKER\_DT = sysdate,  
       d.CHECKER\_ID = p\_Checker\_ID  
       d.RECORD\_STATUS = 'O'  
       d.CORE\_REF\_NO = v\_lc\_Trn\_Ref\_No
  + Update table: DLTBD\_TRANSACTION\_POST p
    - Set: P.TRN\_STATUS = 'S'  
       P.POST\_DATE = sysdate  
       P.CORE\_REF\_NO = v\_lc\_Trn\_Ref\_No  
       P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch
  + Update table: DLTBD\_HOADON\_INFO b
    - Set: B.NGAYGIO = to\_date (p\_NgayGio , 'mm/dd/yyyy hh:mi:ss am')

**Step 19:** Gọi Service gạch nợ:

* Gọi LVBService.BankRequest()

**Step 24:** Update Database đã gạch nợ

* Update table: DLTBD\_HOADON\_INFO
* Nếu EVN return gạch nợ thành công
  + Update Column: TRANGTHAIGD, TRANGTHAIHUYGD, NGAYGIO
  + Value: TRANGTHAIGD = 0, TRANGTHAIHUYGD = 9, NGAYGIO = Date.Now()
* Nếu EVN return False
  + Update Column: TRANGTHAIGD, TRANGTHAIHUYGD
  + Value: TRANGTHAIGD = 1, TRANGTHAIHUYGD = 9
* Nếu xảy ra Exception
  + Update Column: TRANGTHAIGD, TRANGTHAIHUYGD, NGAYGIO
  + Value: TRANGTHAIGD = 2, TRANGTHAIHUYGD = 9, NGAYGIO = Date.Now()

1. **Gạch nợ bổ sung**



**Step 2:** Gọi Service lấy hóa đơn cần gạch nợ bổ sung

* Gọi Service1.getHoaDonGachNoBoSung(Branch\_code, Checker\_ID)

**Step 4:** Select hóa đơn cần gạch nợ bổ sung

select b.MAKH, b.MAHD, b.HOADONID, b.SOTIEN, b.NGAYGIO, b.TRANGTHAIGD,   
t.MAKER\_ID, t.TRN\_BRN, t.TRANSACTION\_ID, t.module, decode (t.module, 'DL', 'EVNHCM', t.module) module\_name  
from   
(

select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID   
from DLTBD\_HOADON\_INFO   
where not (NVL(HDDT, '2') = 1

)  
and TRN\_CHANEL in ('TELLER', 'AUTO')   
and NGAYGIO >= SYSDATE - " + sDATE

union all  
select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID   
from DLTBD\_MN\_HOADON\_INFO   
Where NGAYGIO >= SYSDATE - " + sDATE

union all  
select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID   
from DLTBD\_MT\_HOADON\_INFO   
Where NGAYGIO >= SYSDATE - " + sDATE

union all  
select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID   
from DLTBD\_MB\_HOADON\_INFO   
Where NGAYGIO >= SYSDATE - " + sDATE

union all  
select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID   
from DLTBD\_HN\_HOADON\_INFO   
Where NGAYGIO >= SYSDATE - " + sDATE

) b, DATBD\_TRANSACTION t  
where (NVL(B.TRANGTHAIGD, '1') <>'0')  
AND B.TRANGTHAIHUYGD = '9'  
AND T.MODULE in ('DL','EVNHCM','EVNCPC','EVNSPC', 'EVNHNI','EVNNPC')  
AND T.TRN\_BRN = '" + Branch\_code + "'  
AND b.TRN\_ID = to\_char(t.TRANSACTION\_ID)  
and t.EVENT\_CODE='INIT'  
AND T.RECORD\_STATUS = 'O'  
AND T.AUTH\_STATUS = 'A'

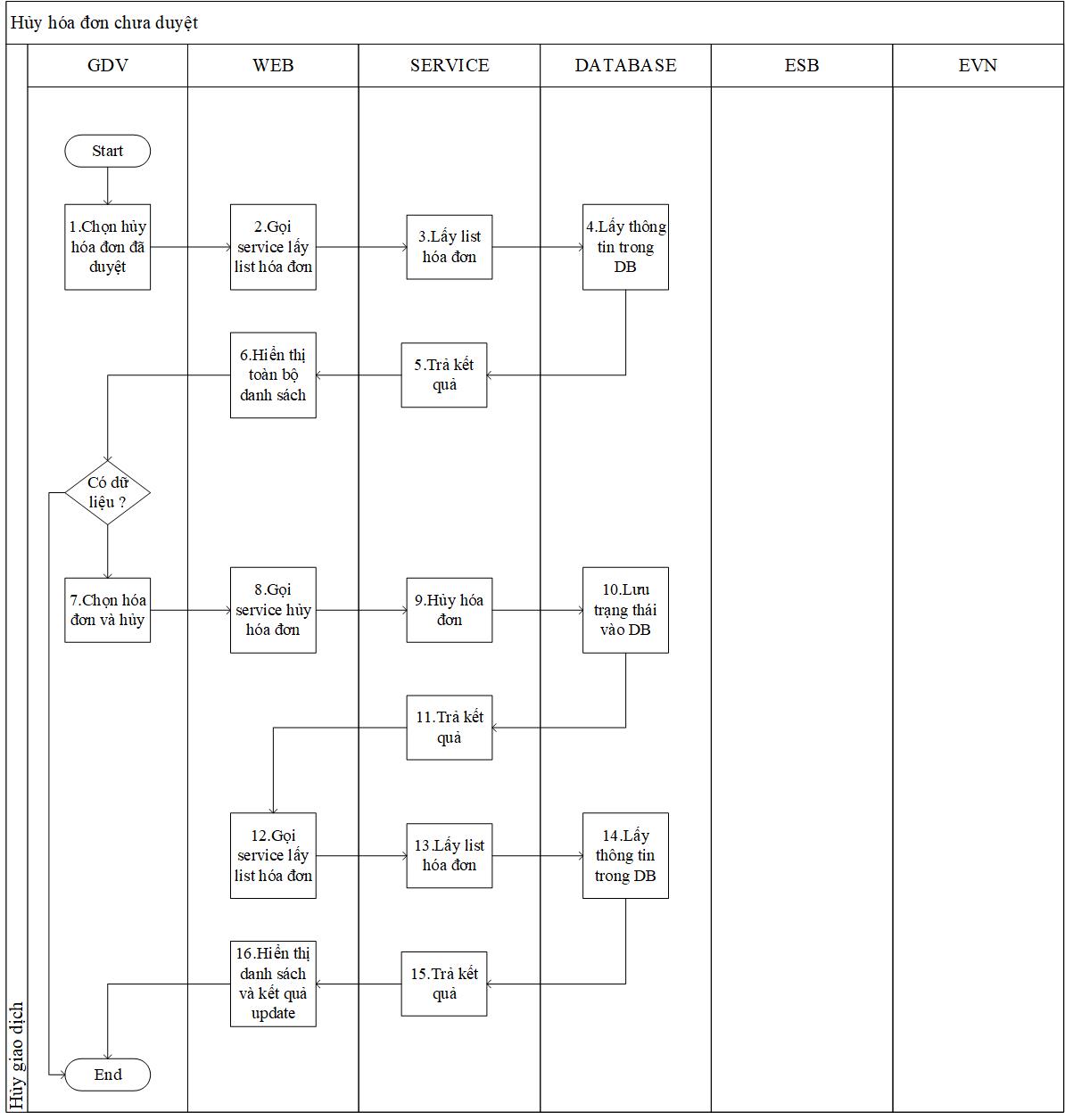
**Step 8:** Gọi Service gạch nợ bổ sung

* Gọi Service1.GachNoBoSungAll() → LVBService.BankRequest()

**Step 14:** Update trạng thái giao dịch

* Update table: DLTBD\_HOADON\_INFO
* Nếu EVN return gạch nợ thành công
  + Update Column: TRANGTHAIGD, TRANGTHAIHUYGD, NGAYGIO
  + Value: TRANGTHAIGD = 0, TRANGTHAIHUYGD = 9, NGAYGIO = Date.Now()
* Nếu EVN return False
  + Update Column: TRANGTHAIGD, TRANGTHAIHUYGD
  + Value: TRANGTHAIGD = 1, TRANGTHAIHUYGD = 9
* Nếu xảy ra Exception
  + Update Column: TRANGTHAIGD, TRANGTHAIHUYGD, NGAYGIO
  + Value: TRANGTHAIGD = 2, TRANGTHAIHUYGD = 9, NGAYGIO = Date.Now()

1. **Hủy Hóa đơn chưa duyệt**



Bước 2.Gọi service lấy list hóa đơn chưa duyệt :

* Lấy list các nhà cung cấp : Service1.getNhaCungCap() -> EVN\_Library.getNhaCungCap()

Return DataTable dt1 = new DataTable("NCC")

|  |  |
| --- | --- |
| madl | tendl |
| EVNHCM | Tổng công ty Điện Lực TPHCM |
| EVNCPC | Tổng công ty Điện Lực Miền Trung |
| EVNSPC | Tổng công ty Điện Lực Miền Nam |
| EVNHNI | Tổng công ty Điện Lực Hà Nội |
| EVNNPC | Tổng công ty Điện Lực Miền Bắc |

ddl\_electric.Items.Add(madl -- tendl)

* Lấy list hóa đơn: Service1. getTransactionsToCancel() -> EVN\_Library. getTransactionsToCancel(Maker\_ID, NCC, PE, MAHD, HDID, TK, AMOUNT)

Bước 4.Lấy list hóa đơn trong DB:

SELECT D.\*,

decode(D.AUTH\_STATUS,'D','Từ chối duyệt',null,'Chưa duyệt') status,

decode (d.module,'DL','EVNHCM',d.module) module\_name

FROM DATBD\_TRANSACTION d where

D.MAKER\_ID=Maker\_ID

and D.MODULE IN ('DL','EVNHCM','EVNCPC','EVNSPC','EVNHNI','EVNNPC')

and D.EVENT\_CODE = 'INIT'

and D.CHECKER\_ID is null

and D.RECORD\_STATUS = 'O'

and (D.AUTH\_STATUS is null or D.AUTH\_STATUS = 'D')

and D.TRN\_DT = (select TODAY from sttm\_dates where branch\_code = d.trn\_brn)

and decode(d.module,'DL','EVNHCM',d.module) = NCC// Nếu chọn Nhà cung cấp (mặc định HCM)

and A.MAKH = PE// Nếu Mã KH != “ ”

and A.MAHD =MAHD // Nếu Mã Hóa đơn != “ ”

and A.HOADONID = HDID// Nếu hóa đơn id != “ ”

and A.SOTIEN =AMOUNT// Nếu so tien != “ ”

and P.AC\_NO = TK// Nếu sotaikhoan != “ ”

order by D.TRANSACTION\_ID

Bước 8.Gọi service hủy hóa đơn: Service1. cancelTransactionAll() ->EVN\_Library. cancelTransactionAll() -> EVNHCM\_Library. cancelTransactionAll(Maker\_ID, Trn\_ID)

Bước 10.Lưu trạng thái vào DB

* Package : pkg\_EVN\_HCM

PROCEDURE: DLP\_CANCELTRANSACTION ( p\_Maker\_ID, p\_Trn\_ID, p\_Err\_String)

v\_Home\_Branch varchar2(5);

v\_Check number;

p\_Err\_String := null;

if p\_Err\_String is null then

update DATBD\_TRANSACTION d

set D.RECORD\_STATUS = 'C',D.EVENT\_CODE = 'CANCEL'

where D.TRANSACTION\_ID = p\_Trn\_ID

and D.MAKER\_ID = p\_Maker\_ID

and D.TRN\_BRN = v\_Home\_Branch

and D.EVENT\_CODE = 'INIT'

and D.CHECKER\_ID is null

and D.RECORD\_STATUS ='O'

and NVL(D.AUTH\_STATUS, 'D') = 'D';

if sql%rowcount = 0 then

p\_Err\_String := '1#0099#Trn\_Ref\_no '||p\_Trn\_ID||'is authed while Cancelling (DLP\_cancelTransaction)';

else

update DLTBD\_TRANSACTION\_POST p

set P.TRN\_STATUS = 'C'

where P.TRANSACTION\_ID = p\_Trn\_ID

and P.EVENT = 'INIT'

and P.EVENT\_SEQ\_NO = (SELECT LAST\_EVENT\_SEQ\_NO FROM DATBD\_TRANSACTION WHERE TRANSACTION\_ID = p\_Trn\_ID AND RECORD\_STATUS ='O');

p\_Err\_String := '0#0000#Cancel Post is successful (DLP\_cancelTransaction)

when others then

p\_Err\_String:='1#0099#updatePostfaied(DLP\_cancelTransaction)'

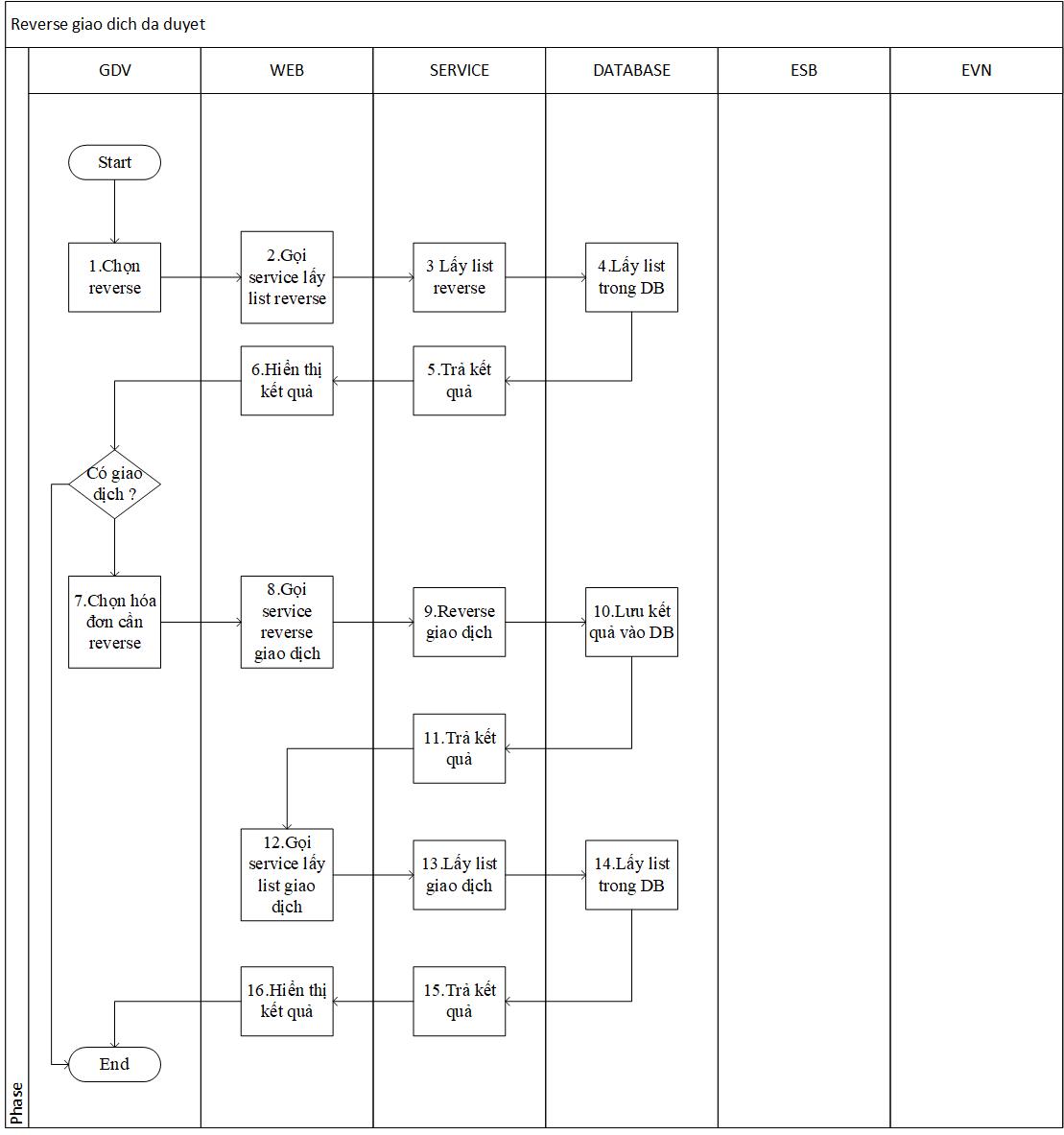
exception

when others then

p\_Err\_String := '1#0099#System Error (DLP\_cancelTransaction)

Bước 12.Gọi service lấy list hóa đơn giống bước 2.

1. **Reverse**



Bước 2.Gọi service lấy list hóa đơn để reverse :

* Lấy list các nhà cung cấp : Service1.getNhaCungCap() -> EVN\_Library.getNhaCungCap()

Return DataTable dt1 = new DataTable("NCC")

|  |  |
| --- | --- |
| Madl | tendl |
| EVNHCM | Tổng công ty Điện Lực TPHCM |
| EVNCPC | Tổng công ty Điện Lực Miền Trung |
| EVNSPC | Tổng công ty Điện Lực Miền Nam |
| EVNHNI | Tổng công ty Điện Lực Hà Nội |
| EVNNPC | Tổng công ty Điện Lực Miền Bắc |

ddl\_electric.Items.Add(madl -- tendl)

* Lấy list cần reverse: : Service1. getAuthTransToRevert() -> EVN\_Library. getAuthTransToRevert(String Maker\_ID, String NCC, String PE, String MAHD, String HDID, String TK, String AMOUNT, String AMOUNT\_Max).

Bước 4.Lấy thông tin trong DB:

* select distinct d.\*, (case d.MODULE WHEN 'DL' THEN 'EVNHCM' ELSE d.MODULE END) NCC

from

(select MAHD, MAKH, SOTIEN, HOADONID, TRN\_ID from DLTBD\_HOADON\_INFO WHERE TIME\_OUT IS NULL AND NGAYGIO >= TRUNC(SYSDATE)

union all

select MAHD, MAKH, SOTIEN, HOADONID, TRN\_ID from DLTBD\_MT\_HOADON\_INFO WHERE TIME\_OUT IS NULL AND NGAYGIO >= TRUNC(SYSDATE)

union all

select MAHD, MAKH, SOTIEN, HOADONID, TRN\_ID from DLTBD\_MB\_HOADON\_INFO WHERE TIME\_OUT IS NULL AND NGAYGIO >= TRUNC(SYSDATE)

union all select MAHD, MAKH, SOTIEN, HOADONID, TRN\_ID from DLTBD\_MN\_HOADON\_INFO WHERE TIME\_OUT IS NULL AND NGAYGIO >= TRUNC(SYSDATE) ) a,

DLTBD\_TRANSACTION\_POST p, DATBD\_TRANSACTION d, STTM\_DATES" + DB\_Link + e

WHERE

a.TRN\_ID = to\_char(D.TRANSACTION\_ID) AND P.TRANSACTION\_ID = D.TRANSACTION\_ID AND P.DRCR\_IND = 'D'

AND P.EVENT\_SEQ\_NO = D.LAST\_EVENT\_SEQ\_NO

AND d.MAKER\_ID = ' Maker\_ID'

and d.EVENT\_CODE = 'INIT'

and d.MODULE in ('DL', 'EVNHCM', 'EVNCPC', 'EVNHNI','EVNNPC', 'EVNSPC')

and d.CHECKER\_ID is not null

and d.RECORD\_STATUS = 'O'

and d.AUTH\_STATUS = 'A'

and d.TRN\_DT = e.TODAY

and d.TRN\_BRN = e.BRANCH\_CODE

and (CASE d.MODULE WHEN 'DL' THEN 'EVNHCM' ELSE d.MODULE END) = NCC // NCC.Trim() != “”

and A.MAKH =PE //PE.Trim()!= “”

and A.MAHD = MAHD // MAHD.Trim() != “”

and A.HOADONID = HDID // HDID.Trim() != “”

and A.SOTIEN >= AMOUNT // AMOUNT.Trim() != “”

and A.SOTIEN <= AMOUNT\_Max // AMOUNT\_Max.Trim() != “”

and P.AC\_NO = TK //TK.Trim() != “”

order by D.TRANSACTION\_ID

Bước 8.Gọi service reverse giao dịch:

* Service1.generateRevertTrans() -> EVN\_Library.generateRevertTrans()-> EVNHCM\_Library.generateRevertTrans(Maker\_ID, Trn\_ID)

Bước 10.Lưu kết quả vào DB

* Package : pkg\_EVN\_HCM
* PROCEDURE DLP\_REVERTTRANSACTION(p\_Maker\_ID varchar2

p\_Trn\_ID number

p\_Err\_String out varchar2)

*Variable*

v\_Home\_Branch varchar2(5);

v\_Check number;

SELECT Home\_Branch into v\_Home\_Branch

FROM ASTB\_USER u

WHERE u.USERNAME = p\_Maker\_ID AND u.RECORD\_STATUS <> 'C';

* IF v\_Home\_Branch is null then

p\_Err\_String:= ' 1#0002#(pkg\_EVN.DLP\_REVERTTRANSACTION) kHÔNG TÌM THẤY USER: p\_Maker\_ID;

RETURN;

END IF;

* if p\_Err\_String is null then

update DATBD\_TRANSACTION d

set D.AUTH\_STATUS = null

,D.EVENT\_CODE = 'REVERT'

,D.LAST\_EVENT\_SEQ\_NO = D.LAST\_EVENT\_SEQ\_NO + 1

where D.TRANSACTION\_ID = p\_Trn\_ID

and D.MAKER\_ID = p\_Maker\_ID

and D.TRN\_BRN = v\_Home\_Branch

and D.EVENT\_CODE = 'INIT'

and D.CHECKER\_ID is not null

and D.RECORD\_STATUS = 'O'

and D.AUTH\_STATUS= 'A';

if sql%rowcount = 0 then

p\_Err\_String := '1#0099#Trn\_Ref\_no ' || p\_Trn\_ID || ' is invalid to Revert or miss match Maker (DLP\_revertTransaction)

else

insert into DLTBD\_TRANSACTION\_POST(TRANSACTION\_ID,AC\_NO,AC\_CCY,AC\_BRANCH,CUST\_GL,DRCR\_IND,AMOUNT,AMOUNT\_TAG,TRN\_CODE,EVENT,EVENT\_SEQ\_NO)

select P.TRANSACTION\_ID, P.AC\_NO, P.AC\_CCY, P.AC\_BRANCH, P.CUST\_GL, P.DRCR\_IND, -1\*P.AMOUNT, P.AMOUNT\_TAG, P.TRN\_CODE, 'REVERT', P.EVENT\_SEQ\_NO+1

from DLTBD\_TRANSACTION\_POST p

where P.TRANSACTION\_ID = p\_Trn\_ID

p\_Err\_String := '0#0000#Revert Transaction is generated successfully (DLP\_revertTransaction)';

end if

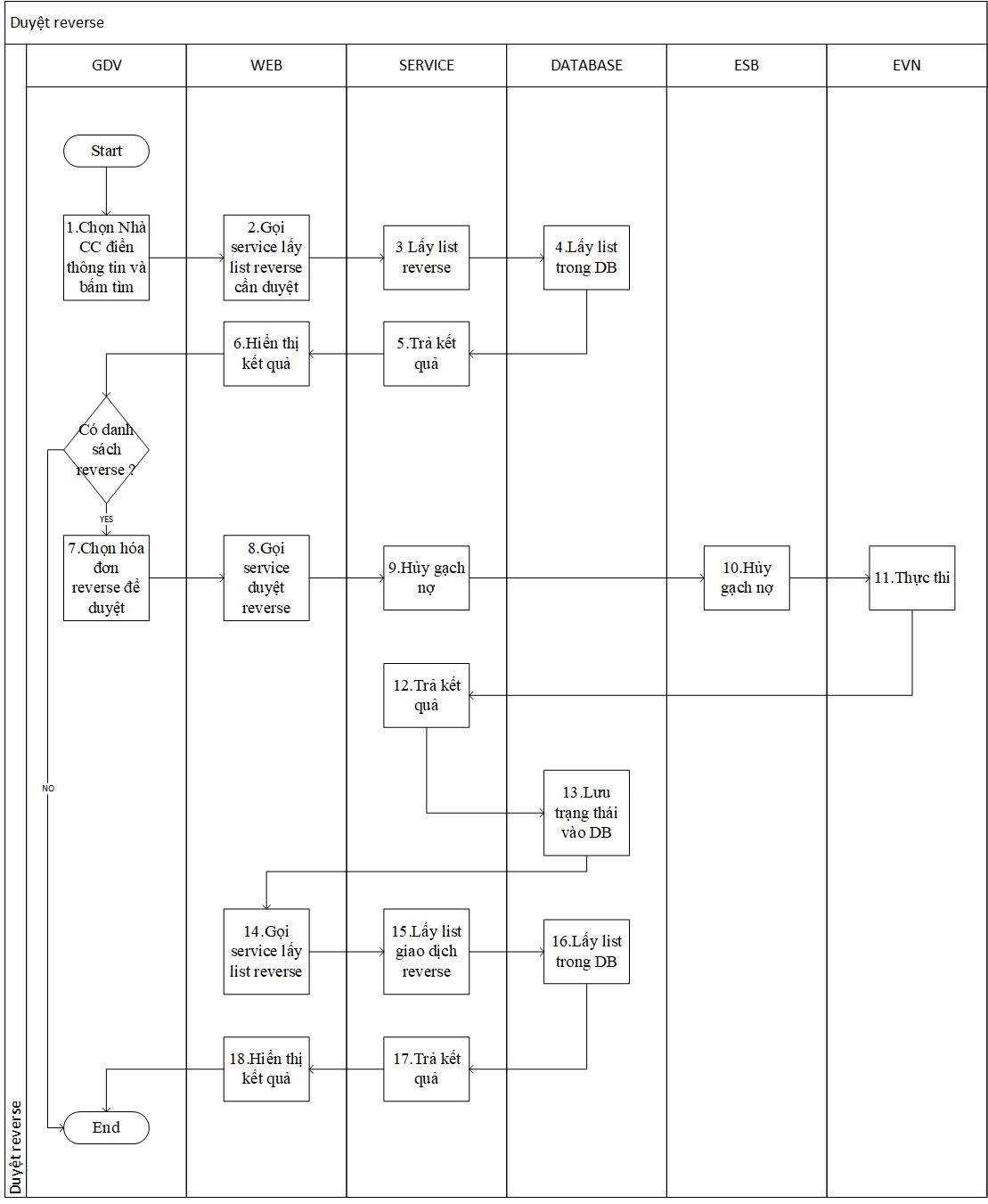
end if

* exception

p\_Err\_String := '1#0099#System Error(DLP\_revertTransaction) ' || sqlerrm;

Bước 12.Lấy list giống bước 2

1. **Duyệt Reverse**



Bước 2.Lấy list reverse để duyệt :

Service1.getRevertTransToAuth() -> EVN\_Library.getRevertTransToAuth(String Checker\_ID, String NCC, String PE, String MAHD, String HDID, String TK, String AMOUNT)

Bước 4.lấy thông tin trong DB:

- Brand\_Code =getBrand(Checker\_ID)

select Home\_Branch from ASTB\_USER u where u.USERNAME =User\_ID and u.RECORD\_STATUS <> 'C'

-

select distinct d.\*, (case d.MODULE WHEN 'DL' THEN 'EVNHCM' ELSE d.MODULE END) NCC

from DATBD\_TRANSACTION d, STTM\_DATES" + DB\_Link + " e “

(SELECT MAKH, MAHD, HOADONID, SOTIEN, TRN\_ID FROM DLTBD\_HOADON\_INFO WHERE TRANGTHAIGD = '0' and TRANGTHAIHUYGD = '9' "

+ @" and trunc(sysdate) = trunc(NGAYGIO)

and 1 =

CASE WHEN to\_number(to\_char(NGAYGIO,'HH24MI')) > 1630 THEN 1

ELSE

CASE WHEN to\_number(to\_char(sysdate,'HH24MI')) > 1630 THEN 0

ELSE 1

END

END "

UNION ALL SELECT MAKH, MAHD, HOADONID, SOTIEN, TRN\_ID FROM DLTBD\_MT\_HOADON\_INFO WHERE TRANGTHAIGD = '0' and TRANGTHAIHUYGD = '9' AND NGAYGIO >= TRUNC(SYSDATE)"

UNION ALL SELECT MAKH, MAHD, HOADONID, SOTIEN, TRN\_ID FROM DLTBD\_HN\_HOADON\_INFO WHERE TRANGTHAIGD = '0' and TRANGTHAIHUYGD = '9' AND NGAYGIO >= TRUNC(SYSDATE)"

UNION ALL SELECT MAKH, MAHD, HOADONID, SOTIEN, TRN\_ID FROM DLTBD\_MB\_HOADON\_INFO WHERE TRANGTHAIHUYGD = '9' AND NGAYGIO >= TRUNC(SYSDATE)"

UNION ALL SELECT MAKH, MAHD, HOADONID, SOTIEN, TRN\_ID FROM DLTBD\_MN\_HOADON\_INFO WHERE TRANGTHAIGD = '0' and TRANGTHAIHUYGD = '9' AND NGAYGIO >= TRUNC(SYSDATE)) A "

", DLTBD\_TRANSACTION\_POST p "

where to\_char(d.TRANSACTION\_ID) = A.TRN\_ID

AND p.TRANSACTION\_ID = d.TRANSACTION\_ID

AND p.DRCR\_IND = 'D'

AND p.EVENT\_SEQ\_NO = d.LAST\_EVENT\_SEQ\_NO

AND p.TRN\_STATUS is null

and d.TRN\_BRN = Branch\_Code +

and d.EVENT\_CODE ='REVERT'

and d.MODULE in ('DL', 'EVNHCM', 'EVNCPC', 'EVNSPC', 'EVNHNI', 'EVNNPC')

and d.CHECKER\_ID is not null

and d.RECORD\_STATUS = 'O'

and d.AUTH\_STATUS is null

and d.TRN\_DT = e.TODAY

and d.TRN\_BRN = e.BRANCH\_CODE

and decode(d.MODULE, 'DL', 'EVNHCM', d.MODULE) = NCC

// NCC.Trim() != “”

and A.MAKH =PE //PE.Trim()!= “”

and A.MAHD = MAHD // MAHD.Trim() != “”

and A.HOADONID = HDID // HDID.Trim() != “”

and A.SOTIEN = AMOUNT // AMOUNT.Trim() != “”

and P.AC\_NO = TK //TK.Trim() != “”

order by D.TRANSACTION\_ID

Bước 8.Gọi service duyệt :

Service1.authRevertTransaction() -> EVN\_Library.authRevertTransaction() -> EVNHCM\_Library.authRevertTransaction(String Checker\_ID,String Trn\_ID).

* SQL\_CMD:select d.\*,rowid row\_id

from DLTBD\_HOADON\_INFO d

where d.TRN\_ID =Trn\_ID

and TRANGTHAIGD = '0'

and TRANGTHAIHUYGD = '9'

and trunc(sysdate) = trunc(NGAYGIO

and 1 =

CASE

WHENto\_number(to\_char(NGAYGIO,'HH24MI'))>1630THEN 1

ELSE

CASE

WHEN to\_number(to\_char(sysdate,'HH24MI'))> 1630 THEN 0 ELSE 1

END

END

String temp = db.SelectCommand(SQL\_CMD, dt\_HoaDon);

if(temp.Substring(0, 1) != "0"||dt\_HoaDon.Rows.Count == 0)

if (temp.Substring(0, 1) != "0"

return "1#0099#System Error in Service authRevertTransaction";

if (dt\_HoaDon.Rows.Count == 0)

return "1#0008#Khong co giao dich nao de huy"

if (dt\_HoaDon.Rows[0]["HDDT"].ToString() == "1")//HDDT

-Gọi service hủy gạch nợ :

**BankCancel**= LVBService.BankCancelHDDT()

- update DLTBD\_HOADON\_INFO

set TRANGTHAIHUYGD = **BankCancel**

where TRN\_ID = Trn\_ID

else

Gọi service : LVBService1.BankCancel() = BankCancel

Update Database

update DLTBD\_HOADON\_INFO

set TRANGTHAIHUYGD = BankCancel

where rowid= dt\_HoaDon.Rows[i]["ROW\_ID"].ToString()

Bước 13.Lưu vào DB:

Package: pkg\_EVN\_HCM

PROCEDURE DLP\_AUTHREVERTTRANS

( p\_Checker\_ID varchar2

p\_Trn\_ID number

p\_Err\_String out nocopy varchar2

)

Variable

v\_Home\_Branch varchar2(5);

v\_Trn\_Desc DATBD\_TRANSACTION.TRN\_DESC%TYPE;

v\_Maker\_ID DATBD\_TRANSACTION.MAKER\_ID%TYPE;

v\_Last\_Event\_Seq\_No DATBD\_TRANSACTION.LAST\_EVENT\_SEQ\_NO%TYPE;

v\_Trn\_Ref\_No DLTBD\_TRANSACTION\_POST.CORE\_REF\_NO%TYPE;

v\_Rcd\_Err PKG\_DOM\_UTIL.RCD\_ERROR;

v\_Value\_Date varchar2(10);

v\_MSGID varchar2(255);

v\_Rcd\_Trn Datbd\_Transaction%ROWTYPE;

* SELECT t.\*

INTO v\_Rcd\_Trn

FROM Datbd\_Transaction t

WHERE t.Transaction\_Id = p\_Trn\_ID

AND t.Record\_Status = 'O'

AND t.Trn\_Brn = v\_Home\_Branch

AND t.Auth\_Status IS NULL;

update

(

select d.AUTH\_STATUS, d.CHECKER\_DT, d.CHECKER\_ID,

d.TRN\_DESC, d.MAKER\_ID, d.LAST\_EVENT\_SEQ\_NO

from DATBD\_TRANSACTION d

where D.TRANSACTION\_ID = p\_Trn\_ID

and D.TRN\_BRN = v\_Home\_Branch

and D.EVENT\_CODE = 'REVERT'

and D.RECORD\_STATUS = 'O'

and D.AUTH\_STATUS is null

) d

set d.AUTH\_STATUS = 'A'

, d.CHECKER\_DT = sysdate

, d.CHECKER\_ID = p\_Checker\_ID

returning d.TRN\_DESC, d.MAKER\_ID, d.LAST\_EVENT\_SEQ\_NO

into v\_Trn\_Desc, v\_Maker\_ID, v\_Last\_Event\_Seq\_No;

-Lấy CORE\_REF\_NO

select P.CORE\_REF\_NO

into **v\_Trn\_Ref\_No**

from DLTBD\_TRANSACTION\_POST P

where P.TRANSACTION\_ID = p\_Trn\_ID

and P.DRCR\_IND = 'C'

and P.EVENT\_SEQ\_NO = v\_Last\_Event\_Seq\_No - 1

and P.TRN\_STATUS = 'S';

-Lấy V\_Value\_Date : select to\_char(today,'dd/MM/yyyy') into v\_Value\_Date from sttm\_dates where branch\_code =v\_Home\_Branch;

- if v\_Value\_Date is not null

+)if (not ebank.PKG\_CORE\_SWITCH.fn\_rt\_reverse\_V3(**v\_Trn\_Ref\_No**,v\_Value\_Date, v\_Trn\_Desc,v\_Maker\_ID, v\_Home\_Branch,c\_Common\_User, p\_Err\_String,v\_MSGID,v\_Rcd\_Trn.XREF))andsubstr(p\_Err\_String,3,20) != '98#FAILURE@GW-REV-04' then

if (substr(p\_Err\_String,3,2) = '80' or substr(p\_Err\_String,3,2) = '90' or substr(p\_Err\_String,3,2) = '91') //Nghi ngờ

update DLTBD\_TRANSACTION\_POST p

set P.TRN\_STATUS = 'S'

P.POST\_DATE = sysdate

P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)

where P.TRANSACTION\_ID = p\_Trn\_ID

and P.EVENT\_SEQ\_NO = v\_Last\_Event\_Seq\_No

and P.TRN\_STATUS is null;

update DLTBD\_HOADON\_INFO b

set

B.Time\_Out = v\_Rcd\_Trn.XREF (!= NULL)

where B.TRN\_ID = to\_char(p\_Trn\_ID);

+) else if (substr(p\_Err\_String,3,20)='98#FAILURE@GW-REV-04')

update datbd\_transaction d

set d.AUTH\_STATUS = 'A'

d.CHECKER\_DT = sysdate

d.CHECKER\_ID = p\_Checker\_ID

d.RECORD\_STATUS = 'O'

d.CORE\_REF\_NO = v\_Trn\_Ref\_No

where d.TRANSACTION\_ID = p\_Trn\_ID

and RECORD\_STATUS = 'B';

update DLTBD\_TRANSACTION\_POST p

set P.TRN\_STATUS = 'S'

P.POST\_DATE = sysdate

P.CORE\_REF\_NO = v\_Trn\_Ref\_No

P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)

where P.TRANSACTION\_ID = p\_Trn\_ID

and P.EVENT\_SEQ\_NO = (select D.LAST\_EVENT\_SEQ\_NO from DATBD\_TRANSACTION d where D.TRANSACTION\_ID = p\_Trn\_ID);

+) ELSE

update datbd\_transaction d

set d.AUTH\_STATUS = 'A'

d.CHECKER\_DT = sysdate

d.CHECKER\_ID = p\_Checker\_ID

d.RECORD\_STATUS = 'O'

d.CORE\_REF\_NO = v\_Trn\_Ref\_No

where d.TRANSACTION\_ID = p\_Trn\_ID

and RECORD\_STATUS = 'B';

update DLTBD\_TRANSACTION\_POST p

set P.TRN\_STATUS = 'S'

P.POST\_DATE = sysdate

P.CORE\_REF\_NO = v\_Trn\_Ref\_No

P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)

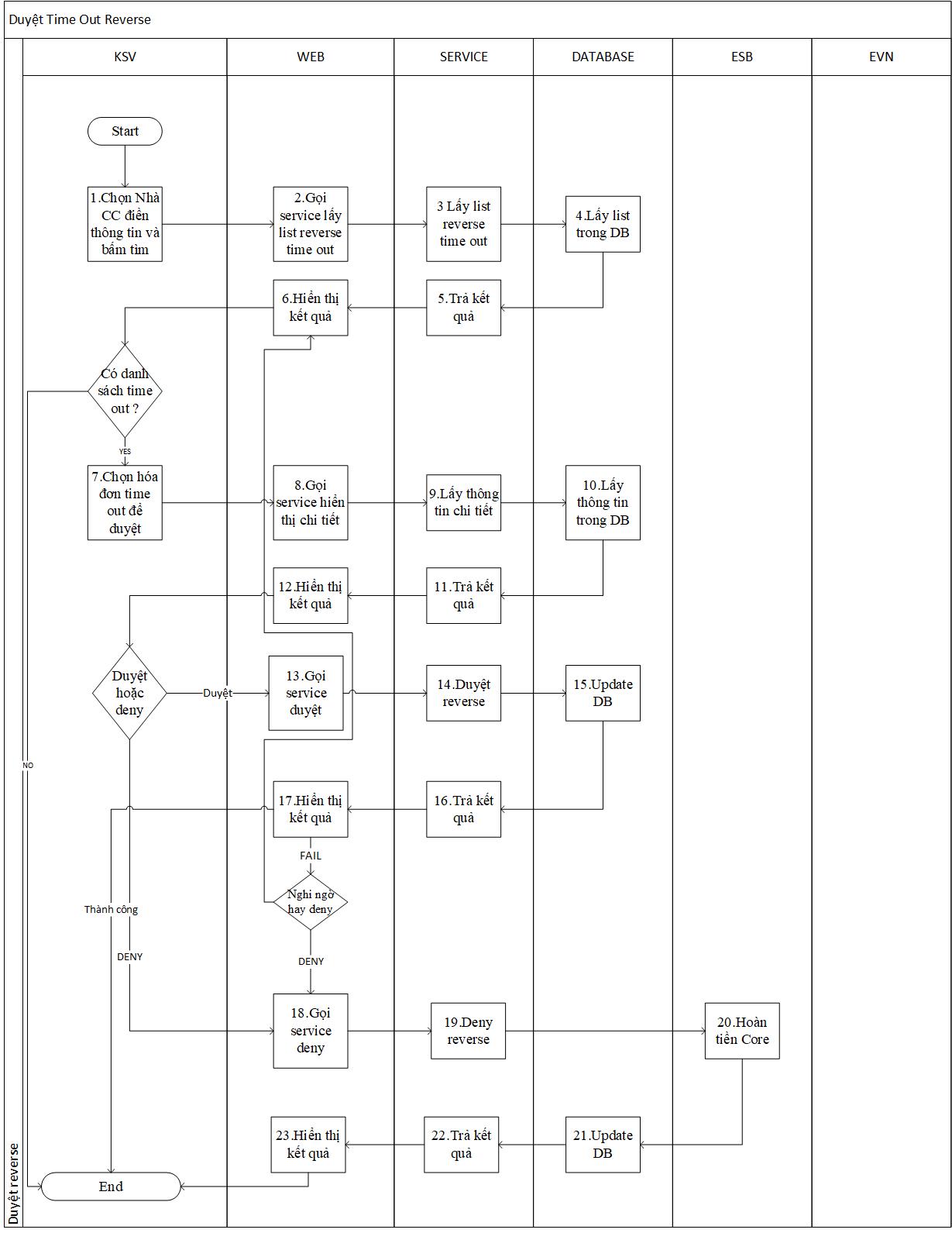
where P.TRANSACTION\_ID = p\_Trn\_ID

and P.EVENT\_SEQ\_NO = v\_Last\_Event\_Seq\_No

and P.TRN\_STATUS is null;

Bước 14.Lấy list giao dịch reverse : Giống bước 2

1. **Duyệt Time Out Reverse**



Bước 2 :lấy list reverse time out :

getRevertTransToAuth\_TimeOut(String Checker\_ID, String NCC, String PE, String MAHD, String HDID, String TK, String AMOUNT)  
Bước 4: Lấy list trong DB:

* Branch\_Code = getBranch(Checker\_ID):

+ ) select Home\_Branch from ASTB\_USER u where u.USERNAME=User\_ID and u.RECORD\_STATUS <> 'C'

+) DataTable("dt")

select distinct d.\*, (case d.MODULE WHEN 'DL' THEN 'EVNHCM' ELSE d.MODULE END) NCC

from DATBD\_TRANSACTION d, DLTBD\_TRANSACTION\_POST p STTM\_DATES" DB\_Link e

**(**SELECT MAKH, MAHD, HOADONID, SOTIEN, TRN\_ID FROM DLTBD\_HOADON\_INFO WHERE TRANGTHAIGD = '0' and **TRANGTHAIHUYGD = '0'** AND **TIME\_OUT IS NOT NULL**

and trunc(sysdate) = trunc(NGAYGIO)

and 1 =

CASE WHEN to\_number(to\_char(NGAYGIO,'HH24MI')) > 1630 THEN 1

ELSE

CASE WHEN to\_number(to\_char(sysdate,'HH24MI')) > 1630 THEN 0

ELSE 1

END

END

UNION ALL SELECT MAKH, MAHD, HOADONID, SOTIEN, TRN\_ID FROM DLTBD\_MT\_HOADON\_INFO WHERE TRANGTHAIGD = '0' and TRANGTHAIHUYGD = '0' AND TIME\_OUT IS NOT NULL

UNION ALL SELECT MAKH, MAHD, HOADONID, SOTIEN, TRN\_ID FROM DLTBD\_MB\_HOADON\_INFO WHERE TRANGTHAIGD = '0' and TRANGTHAIHUYGD = '0' AND TIME\_OUT IS NOT NULL

UNION ALL SELECT MAKH, MAHD, HOADONID, SOTIEN, TRN\_ID FROM DLTBD\_HN\_HOADON\_INFO WHERE TRANGTHAIGD = '0' and TRANGTHAIHUYGD = '0' AND TIME\_OUT IS NOT NULL

UNION ALL SELECT MAKH, MAHD, HOADONID, SOTIEN, TRN\_ID FROM DLTBD\_MN\_HOADON\_INFO WHERE TRANGTHAIGD = '0' and TRANGTHAIHUYGD = '0' AND TIME\_OUT IS NOT NULL **)**A

WHERE to\_char(d.TRANSACTION\_ID) = A.TRN\_ID

AND p.TRANSACTION\_ID = d.TRANSACTION\_ID

AND p.DRCR\_IND = 'D'

AND p.EVENT\_SEQ\_NO = d.LAST\_EVENT\_SEQ\_NO

AND p.TRN\_STATUS ='S' "

and d.TRN\_BRN = Branch\_Code

and d.EVENT\_CODE = 'REVERT' "

and d.MODULE in('DL','EVNHCM','EVNCPC','EVNSPC','EVNHNI','EVNNPC')

and d.CHECKER\_ID = Checker\_ID

and d.RECORD\_STATUS = 'O'

and d.AUTH\_STATUS ='A'

and d.TRN\_DT = e.TODAY

and d.TRN\_BRN = e.BRANCH\_CODE

and decode(d.MODULE, 'DL', 'EVNHCM', d.MODULE) = NCC

// NCC.Trim() != “”

and A.MAKH =PE //PE.Trim()!= “”

and A.MAHD = MAHD // MAHD.Trim() != “”

and A.HOADONID = HDID // HDID.Trim() != “”

and A.SOTIEN = AMOUNT // AMOUNT.Trim() != “”

and P.AC\_NO = TK //TK.Trim() != “”

order by D.TRANSACTION\_ID

Bước 8: Lấy List Time Out :

* Service1.getTransactionDetail() - > EVN\_Library.getTransactionDetail()
* Service1.getCustAccFromFlexCube\_Acc() - > EVN\_Library.getCustAccFromFlexCube\_Acc()
* Service1.getGDInDay() -> Evn\_Library.getGDInDay()

Bước 10 : Lấy thông tin trong DB:

* getTransactionDetail(Request["transaction\_id"].ToString()) :

+ DataTable("TRANSACTION")

select p.\*, d.MAKER\_ID, d.TRN\_DT, d.TRN\_BRN, d.TRN\_CODE, d.MODULE, (SELECT NAME FROM ASTB\_USER WHERE USERNAME = d.MAKER\_ID ) NAME

from DATBD\_TRANSACTION d, DLTBD\_TRANSACTION\_POST p

where to\_char(d.TRANSACTION\_ID) = Trn\_ID

and d.RECORD\_STATUS = 'O'

and NVL(p.TRN\_STATUS,'O') <> 'C'

and d.TRANSACTION\_ID = p.TRANSACTION\_ID

and d.LAST\_EVENT\_SEQ\_NO = p.EVENT\_SEQ\_NO

order by d.TRANSACTION\_ID

+ DataTable("HOADON")

select h.MAKH, h.MAHD, h.SOTIEN, h.HOADONID, h.NGAYGIO, h.MAGIAODICH, h.DAINHD, h.TRANGTHAIGD, h.TRANGTHAIHUYGD, h.TRN\_CHANEL, (select max(MADL) from DLTB\_CUSTOMER\_INFO where MAKH = h.MAKH ) MADL, (select max(NGAYGIO) from DLTBD\_HOADON\_INFO where MAKH = h.MAKH and MAHD = h.MAHD and TRN\_ID <> h.TRN\_ID ) M\_NGAYGIO, 'EVNHCM' AS EVN "

from DLTBD\_HOADON\_INFO h

where h.TRN\_ID =Trn\_ID

union all

select k.MAKH, k.MAHD, k.SOTIEN, k.HOADONID, k.NGAYGIO, k.MAGIAODICH, k.DAINHD, k.TRANGTHAIGD, k.TRANGTHAIHUYGD, k.TRN\_CHANEL, (select max(MADL) from DLTB\_MT\_CUSTOMER\_INFO where MAKH = k.MAKH ) MADL, (select max(NGAYGIO) from DLTBD\_MT\_HOADON\_INFO where MAKH = k.MAKH and MAHD = k.MAHD and TRN\_ID <> k.TRN\_ID ) M\_NGAYGIO, 'EVNCPC' as EVN "

from DLTBD\_MT\_HOADON\_INFO k

where k.TRN\_ID =Trn\_ID

union all

select k.MAKH, k.MAHD, k.SOTIEN, k.HOADONID, k.NGAYGIO, k.MAGIAODICH, k.DAINHD, k.TRANGTHAIGD, k.TRANGTHAIHUYGD, k.TRN\_CHANEL, (select max(MADL) from DLTB\_HN\_CUSTOMER\_INFO where MAKH = k.MAKH ) MADL, (select max(NGAYGIO) from DLTBD\_HN\_HOADON\_INFO where MAKH = k.MAKH and MAHD = k.MAHD and TRN\_ID <> k.TRN\_ID ) M\_NGAYGIO, 'EVNHNI' as EVN "

from DLTBD\_HN\_HOADON\_INFO k

where k.TRN\_ID =Trn\_ID

union all

select k.MAKH, k.MAHD, k.SOTIEN, k.HOADONID, k.NGAYGIO, k.MAGIAODICH, k.DAINHD, k.TRANGTHAIGD, k.TRANGTHAIHUYGD, k.TRN\_CHANEL, (select max(MADL) from DLTB\_MB\_CUSTOMER\_INFO where MAKH = k.MAKH ) MADL, (select max(NGAYGIO) from DLTBD\_MB\_HOADON\_INFO where MAKH = k.MAKH and MAHD = k.MAHD and TRN\_ID <> k.TRN\_ID ) M\_NGAYGIO, 'EVNNPC' as EVN

from DLTBD\_MB\_HOADON\_INFO k

where k.TRN\_ID = Trn\_ID

union all

select l.MAKH, l.MAHD, l.SOTIEN, l.HOADONID, l.NGAYGIO, l.MAGIAODICH, l.DAINHD, l.TRANGTHAIGD, l.TRANGTHAIHUYGD, l.TRN\_CHANEL, (select max(MADL) from DLTB\_MN\_CUSTOMER\_INFO where MAKH = l.MAKH ) MADL, (select max(NGAYGIO) from DLTBD\_MN\_HOADON\_INFO where MAKH = l.MAKH and MAHD = l.MAHD and TRN\_ID <> l.TRN\_ID ) M\_NGAYGIO, 'EVNSPC' as EVN

from DLTBD\_MN\_HOADON\_INFO l

where l.TRN\_ID = Trn\_ID

* getCustAccFromFlexCube\_Acc(String Account\_No)

+ DataTable("Account") //dt

select c.CUST\_AC\_NO,c.AC\_DESC,c.ACY\_AVL\_BAL,c.BRANCH\_CODE, (select BRANCH\_NAME from sttm\_branch where BRANCH\_CODE = c.BRANCH\_CODE) BRANCH\_NAME, c.CUST\_NO

from sttm\_cust\_account c

where (c.CR\_GL like '4211%' OR C.cust\_no in (select B.WALKIN\_CUSTOMER from sttm\_branch b where B.RECORD\_STAT ='O'))

and c.RECORD\_STAT = 'O'

and c.CUST\_AC\_NO = Account\_No

sCIF = dt.Rows[0]["CUST\_NO"].ToString()

+ DataTable("Customer")

select s.CUSTOMER\_NO,s.CUSTOMER\_NAME1, s.UNIQUE\_ID\_VALUE,s.UDF\_1 NGAY\_CAP,s.UDF\_2 NOI\_CAP, P.TELEPHONE,s.ADDRESS\_LINE1 || s.ADDRESS\_LINE2 || s.ADDRESS\_LINE3 || s.ADDRESS\_LINE4 ADDRESS

from sttm\_customer" + DB\_Link + " s left join sttm\_cust\_personal p

on s.customer\_no =p.customer\_no where s.CUSTOMER\_NO =sCIF

and record\_stat = 'O'

* getGDInDay(Trn\_ID, NCC):

+ DataTable("dt")

select d.\*

from DLTBD\_HOADON\_INFO d, DATBD\_TRANSACTION t "

where d.TRN\_ID = t.TRANSACTION\_ID "

and (d.TRN\_CHANEL = 'TELLER' or d.TRN\_CHANEL = 'AUTO')

and t.MODULE = 'DL'

and t.TRANSACTION\_ID <> Trn\_ID

and d.MAKH = (SELECT DISTINCT MAKH FROM DLTBD\_HOADON\_INFO WHERE TRN\_ID=Trn\_ID

and d.MAHD IN (SELECT MAHD FROM DLTBD\_HOADON\_INFO WHERE TRN\_ID =Trn\_ID

and trunc(sysdate) = trunc(d.NGAYGIO)

-Duyệt :

Bước 13. Gọi service duyệt : authRevertTransaction\_TimeOut()

Bước 15 :Update DB

Package: pkg\_EVN\_HCM

PROCEDURE DLP\_AUTHREVERTTRANS

( p\_Checker\_ID varchar2

p\_Trn\_ID number

p\_Err\_String out nocopy varchar2

)

Variable

v\_Home\_Branch varchar2(5);

v\_Trn\_Desc DATBD\_TRANSACTION.TRN\_DESC%TYPE;

v\_Maker\_ID DATBD\_TRANSACTION.MAKER\_ID%TYPE;

v\_Last\_Event\_Seq\_No DATBD\_TRANSACTION.LAST\_EVENT\_SEQ\_NO%TYPE;

v\_Trn\_Ref\_No DLTBD\_TRANSACTION\_POST.CORE\_REF\_NO%TYPE;

v\_Rcd\_Err PKG\_DOM\_UTIL.RCD\_ERROR;

v\_Value\_Date varchar2(10);

v\_MSGID varchar2(255);

v\_Rcd\_Trn Datbd\_Transaction%ROWTYPE;

* SELECT t.\*

INTO v\_Rcd\_Trn

FROM Datbd\_Transaction t

WHERE t.Transaction\_Id = p\_Trn\_ID

AND t.Record\_Status = 'O'

AND t.Trn\_Brn = v\_Home\_Branch

AND t.Auth\_Status IS NULL;

update

(

select d.AUTH\_STATUS, d.CHECKER\_DT, d.CHECKER\_ID,

d.TRN\_DESC, d.MAKER\_ID, d.LAST\_EVENT\_SEQ\_NO

from DATBD\_TRANSACTION d

where D.TRANSACTION\_ID = p\_Trn\_ID

and D.TRN\_BRN = v\_Home\_Branch

and D.EVENT\_CODE = 'REVERT'

and D.RECORD\_STATUS = 'O'

and D.AUTH\_STATUS is null

) d

set d.AUTH\_STATUS = 'A'

, d.CHECKER\_DT = sysdate

, d.CHECKER\_ID = p\_Checker\_ID

returning d.TRN\_DESC, d.MAKER\_ID, d.LAST\_EVENT\_SEQ\_NO

into v\_Trn\_Desc, v\_Maker\_ID, v\_Last\_Event\_Seq\_No;

-Lấy CORE\_REF\_NO

select P.CORE\_REF\_NO

into **v\_Trn\_Ref\_No**

from DLTBD\_TRANSACTION\_POST P

where P.TRANSACTION\_ID = p\_Trn\_ID

and P.DRCR\_IND = 'C'

and P.EVENT\_SEQ\_NO = v\_Last\_Event\_Seq\_No - 1

and P.TRN\_STATUS = 'S';

-Lấy V\_Value\_Date : select to\_char(today,'dd/MM/yyyy') into v\_Value\_Date from sttm\_dates where branch\_code =v\_Home\_Branch;

- if v\_Value\_Date is not null

+)if (not ebank.PKG\_CORE\_SWITCH.fn\_rt\_reverse\_V3(**v\_Trn\_Ref\_No**,v\_Value\_Date, v\_Trn\_Desc,v\_Maker\_ID, v\_Home\_Branch,c\_Common\_User, p\_Err\_String,v\_MSGID,v\_Rcd\_Trn.XREF))andsubstr(p\_Err\_String,3,20) != '98#FAILURE@GW-REV-04' then

if (substr(p\_Err\_String,3,2) = '80' or substr(p\_Err\_String,3,2) = '90' or substr(p\_Err\_String,3,2) = '91') //Nghi ngờ

update DLTBD\_TRANSACTION\_POST p

set P.TRN\_STATUS = 'S'

P.POST\_DATE = sysdate

P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)

where P.TRANSACTION\_ID = p\_Trn\_ID

and P.EVENT\_SEQ\_NO = v\_Last\_Event\_Seq\_No

and P.TRN\_STATUS is null;

update DLTBD\_HOADON\_INFO b

set

B.Time\_Out = v\_Rcd\_Trn.XREF (!= NULL)

where B.TRN\_ID = to\_char(p\_Trn\_ID);

+) else if (substr(p\_Err\_String,3,20)='98#FAILURE@GW-REV-04')

update datbd\_transaction d

set d.AUTH\_STATUS = 'A'

d.CHECKER\_DT = sysdate

d.CHECKER\_ID = p\_Checker\_ID

d.RECORD\_STATUS = 'O'

d.CORE\_REF\_NO = v\_Trn\_Ref\_No

where d.TRANSACTION\_ID = p\_Trn\_ID

and RECORD\_STATUS = 'B';

update DLTBD\_TRANSACTION\_POST p

set P.TRN\_STATUS = 'S'

P.POST\_DATE = sysdate

P.CORE\_REF\_NO = v\_Trn\_Ref\_No

P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)

where P.TRANSACTION\_ID = p\_Trn\_ID

and P.EVENT\_SEQ\_NO = (select D.LAST\_EVENT\_SEQ\_NO from DATBD\_TRANSACTION d where D.TRANSACTION\_ID = p\_Trn\_ID);

+) ELSE

update datbd\_transaction d

set d.AUTH\_STATUS = 'A'

d.CHECKER\_DT = sysdate

d.CHECKER\_ID = p\_Checker\_ID

d.RECORD\_STATUS = 'O'

d.CORE\_REF\_NO = v\_Trn\_Ref\_No

where d.TRANSACTION\_ID = p\_Trn\_ID

and RECORD\_STATUS = 'B';

update DLTBD\_TRANSACTION\_POST p

set P.TRN\_STATUS = 'S'

P.POST\_DATE = sysdate

P.CORE\_REF\_NO = v\_Trn\_Ref\_No

P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)

where P.TRANSACTION\_ID = p\_Trn\_ID

and P.EVENT\_SEQ\_NO = v\_Last\_Event\_Seq\_No

and P.TRN\_STATUS is null;

+Nếu **result\_AuthTrn[0]** (= authRevertTransaction\_TimeOut()) = 0 là thành công -> END

+!= 0 thì:

* result\_AuthTrn[0].substring(4,2) =80 ,90,91 -> nghi ngờ
* Sang Deny : Gọi service authRevertTransaction\_Erro

\*service authRevertTransaction\_Erro(String Checker\_ID, String Trn\_ID):

result[0] = "0000"

update DATBD\_TRANSACTION d

set D.RECORD\_STATUS = 'O'

D.AUTH\_STATUS = NULL ,

d.CHECKER\_DT =NULL ,

d.CHECKER\_ID= NULL

where to\_char(d.TRANSACTION\_ID) = Trn\_ID

and D.EVENT\_CODE = 'REVERT'

and D.AUTH\_STATUS = 'A'

* if (db.DMLCommand(SQL).Substring(0, 1) != "0")

result[0] = "1#0009#authRevertTransaction\_Erro Loi Cap nhat DATBD\_TRANSACTION: TRANSACTION\_ID: " + Trn\_ID

UPDATE DLTBD\_TRANSACTION\_POST p

SET P.TRN\_STATUS = NULL, P.POST\_DATE = NULL, P.CORE\_TRN\_DT = NULL

WHERE P.TRANSACTION\_ID = 'Trn\_ID

AND P.EVENT\_SEQ\_NO = (SELECT D.LAST\_EVENT\_SEQ\_NO

FROM DATBD\_TRANSACTION d

WHERE to\_char(d.TRANSACTION\_ID) = Trn\_ID

* if (db.DMLCommand(SQL\_CMD).Substring(0, 1) != "0")

result[0] = "1#0009#authRevertTransaction\_Erro Loi Cap nhat core ref no: TRANSACTION\_ID: " + Trn\_ID;

return result

1.Nếu result. Substring(0, 1) == "0"

Xác nhận revert core không thành công: OK.

2. !=0

Xác nhận revert core không thành công: FAIL

**DENY :**

\*service authRevertTransaction\_Erro(String Checker\_ID, String Trn\_ID):

result[0] = "0000"

update DATBD\_TRANSACTION d

set D.RECORD\_STATUS = 'O'

D.AUTH\_STATUS = NULL ,

d.CHECKER\_DT =NULL ,

d.CHECKER\_ID= NULL

where to\_char(d.TRANSACTION\_ID) = Trn\_ID

and D.EVENT\_CODE = 'REVERT'

and D.AUTH\_STATUS = 'A'

* if (db.DMLCommand(SQL).Substring(0, 1) != "0")

result[0] = "1#0009#authRevertTransaction\_Erro Loi Cap nhat DATBD\_TRANSACTION: TRANSACTION\_ID: " + Trn\_ID

UPDATE DLTBD\_TRANSACTION\_POST p

SET P.TRN\_STATUS = NULL, P.POST\_DATE = NULL, P.CORE\_TRN\_DT = NULL

WHERE P.TRANSACTION\_ID = 'Trn\_ID

AND P.EVENT\_SEQ\_NO = (SELECT D.LAST\_EVENT\_SEQ\_NO

FROM DATBD\_TRANSACTION d

WHERE to\_char(d.TRANSACTION\_ID) = Trn\_ID

* if (db.DMLCommand(SQL\_CMD).Substring(0, 1) != "0")

result[0] = "1#0009#authRevertTransaction\_Erro Loi Cap nhat core ref no: TRANSACTION\_ID: " + Trn\_ID;

return result

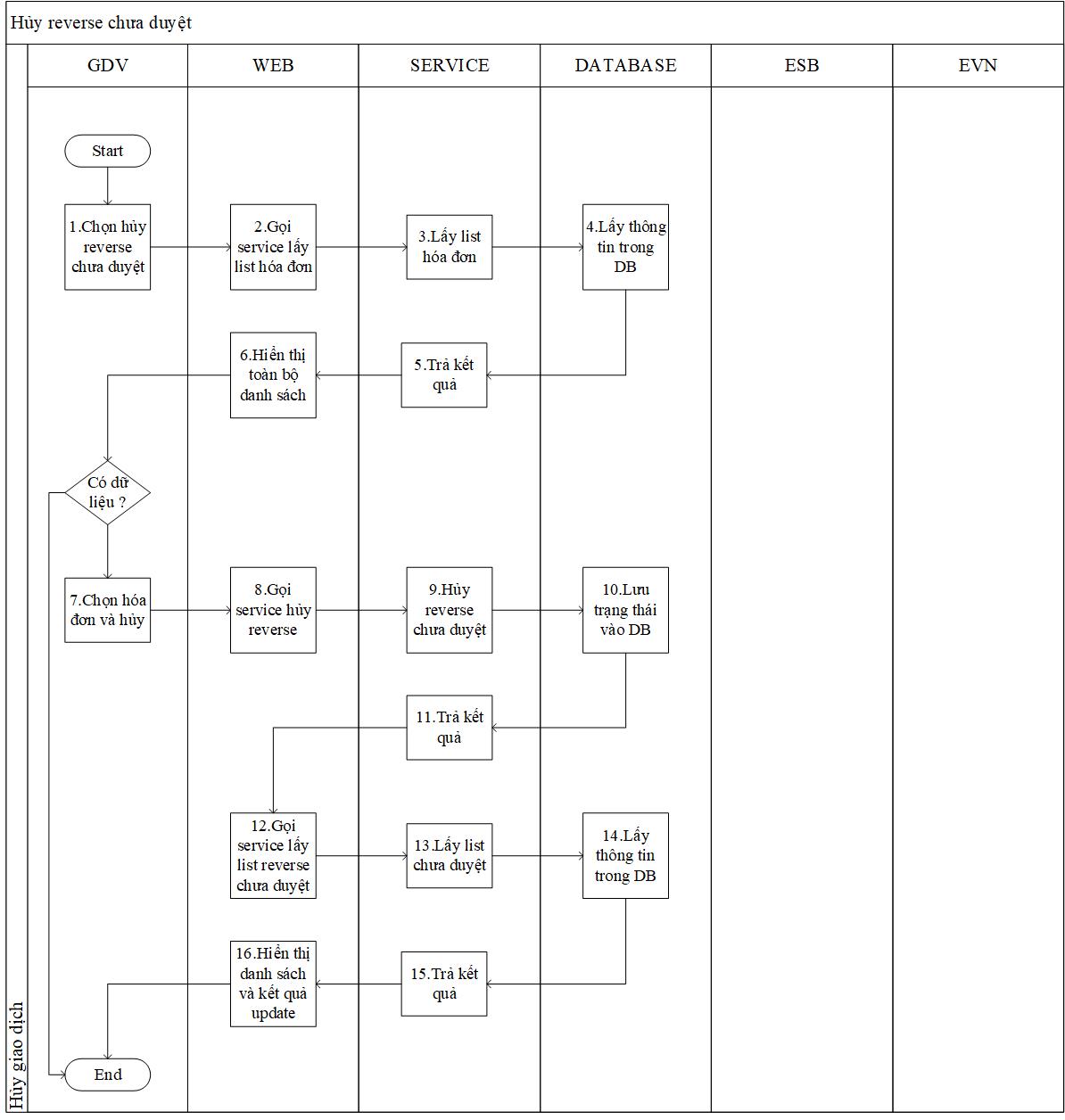
1.Nếu result. Substring(0, 1) == "0"

Xác nhận revert core không thành công: OK.

2. !=0

Xác nhận revert core không thành công: FAIL

1. **Hủy Reverse**



Bước 2: Gọi service lấy list gd reverse chưa duyệt

Service1.getRevertTransToCancel() -> EVN\_Library.getRevertTransToCancel(String Maker\_ID)

Bước 4 : Lấy thông tin trong DB:

select d.\*, decode (d.module,'DL','EVNHCM',d.module) module\_name

from DATBD\_TRANSACTION d, STTM\_DATES" + DB\_Link + " e

where D.MAKER\_ID = Maker\_ID

and D.EVENT\_CODE = 'REVERT'

and d.MODULE in ('DL','EVNCPC', 'EVNSPC','EVNNPC','EVNHNI')

and D.RECORD\_STATUS = 'O'

and D.AUTH\_STATUS is null

and D.TRN\_DT = E.TODAY

and D.TRN\_BRN = E.BRANCH\_CODE

order by D.TRANSACTION\_ID

Bước 8 : Gọi service hủy reverse chưa duyệt :

Service1.cancelRevertTransactionAll() 🡪 EVN\_Library.cancelRevertTransactionAll() 🡪 EVNHCM\_Library.cancelRevertTransaction(String Maker\_ID, String Trn\_ID)

Bước 10 :Lưu thông tin vào DB :

Package : pkg\_EVN\_HCM

PROCEDURE : DLP\_cancelRevertTrans

(p\_Maker\_ID varchar2

p\_Trn\_ID number

p\_Err\_String out varchar2 )

Variable

v\_Home\_Branch varchar2(5);

v\_Check number;

p\_Err\_String := null;

if p\_Err\_String is null then

update DATBD\_TRANSACTION d

set D.EVENT\_CODE = 'INIT'

D.AUTH\_STATUS = 'A'

D.LAST\_EVENT\_SEQ\_NO = D.LAST\_EVENT\_SEQ\_NO - 1

where D.TRANSACTION\_ID = p\_Trn\_ID

and D.MAKER\_ID = p\_Maker\_ID

and D.TRN\_BRN = v\_Home\_Branch

and D.EVENT\_CODE = 'REVERT'

and D.RECORD\_STATUS = 'O'

and D.AUTH\_STATUS is null

if sql%rowcount = 0 then

p\_Err\_String := '1#0099#Trn\_Ref\_no p\_Trn\_ID is authed while Cancelling

else

update DLTBD\_TRANSACTION\_POST p

set P.TRN\_STATUS = 'C'

where P.TRANSACTION\_ID = p\_Trn\_ID

and P.EVENT = 'REVERT'

and P.EVENT\_SEQ\_NO = (SELECT LAST\_EVENT\_SEQ\_NO+1 FROM DATBD\_TRANSACTION WHERE TRANSACTION\_ID = p\_Trn\_ID);

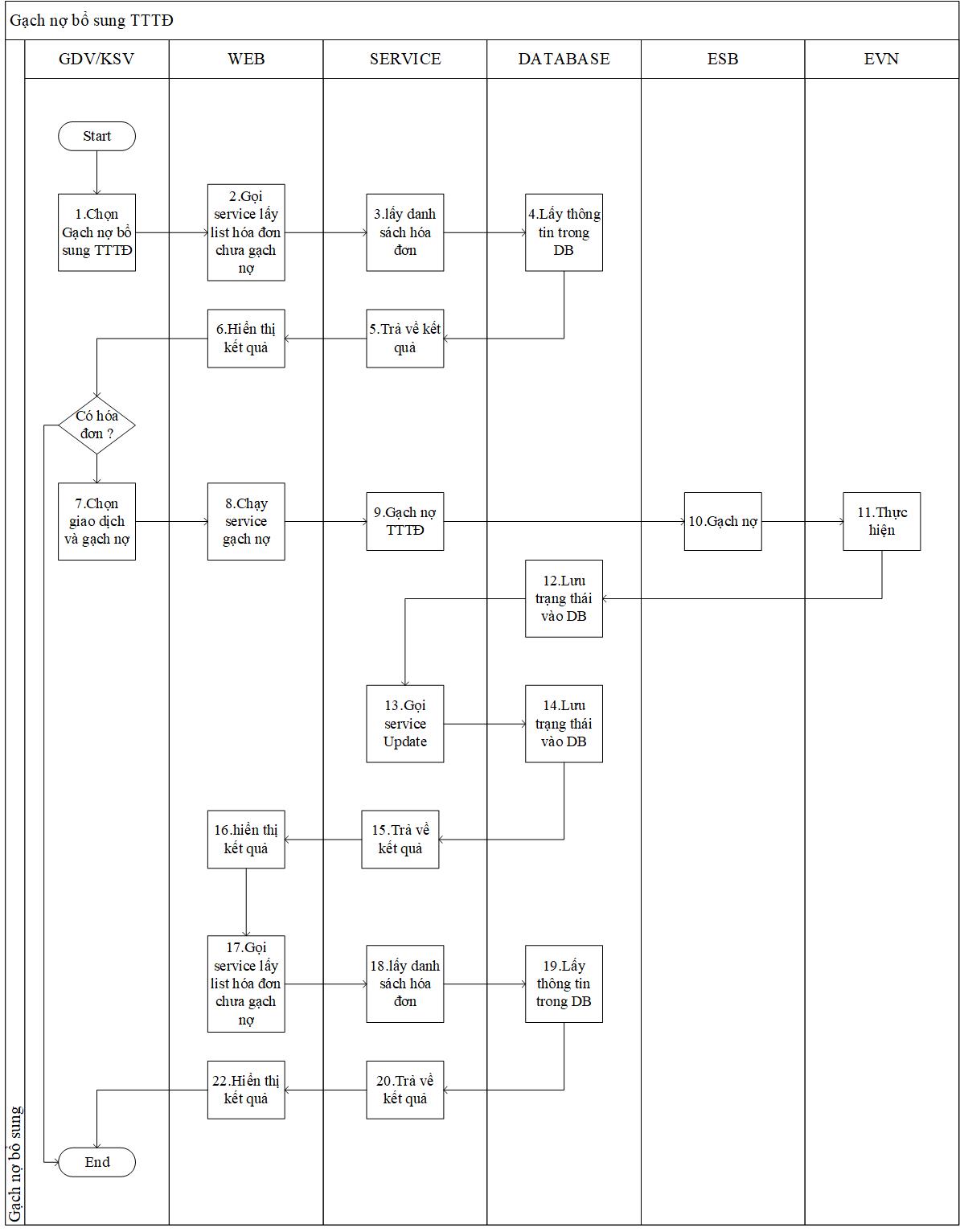
p\_Err\_String := '0#0000#Cancel Post is successful

exception: p\_Err\_String := '1#0099#update Post faied

exception

p\_Err\_String := '1#0099#System Error (DLP\_cancelRevertTrans)

1. **Gạch Nợ Bổ sung HDDT**



Bước 2: Gọi service lấy list hóa đơn chưa gạch nợ :

EVN\_Auto\_Service.Service1.getHoaDonGachNoBoSung\_Ver2() 🡪 EVN\_AUTO. getHoaDonGachNoBoSung(String maker\_ID, string branchCode)

Bước 4 : Lấy thông tin trong DB:

SQL\_CMD = "select \* from DLTBD\_SCHEDULES where RECORD\_STAT='O'";

result = db.SelectCommand(SQL\_CMD, dt\_DATE)

if (result.Substring(0, 1) =="0"&& dt\_DATE.Rows.Count== 1)

sDATE = dt\_DATE.Rows[0]["NUMBER\_DAY"].ToString()

else return dt // “ “

SQL\_CMD :

select b.MAKH, b.MAHD, b.HOADONID, b.SOTIEN, b.NGAYGIO, b.TRANGTHAIGD, t.MAKER\_ID, t.TRN\_BRN, t.TRANSACTION\_ID, t.module,

decode (t.module,'DL','EVNHCM',t.module) module\_name

from ( select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD,TRN\_CHANEL, TRN\_ID,TIME\_OUT

from DLTBD\_HOADON\_INFO where not (NVL(HDDT, '2') = 1 )

and TRN\_CHANEL in ('TELLER', 'AUTO') and NGAYGIO >= SYSDATE - sDATE

union all

select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID, TIME\_OUT

from DLTBD\_MN\_HOADON\_INFO Where NGAYGIO >= SYSDATE sDATE

union all

select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID, TIME\_OUT

from DLTBD\_MT\_HOADON\_INFO Where NGAYGIO >= SYSDATE sDATE

union all

select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID, TIME\_OUT

from DLTBD\_MB\_HOADON\_INFO Where NGAYGIO >= SYSDATE sDATE

union all

select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID, TIME\_OUT \n"

from DLTBD\_HN\_HOADON\_INFO Where NGAYGIO >=SYSDATE-sDATE)b, DATBD\_TRANSACTION t

WHERE(NVL(B.TRANGTHAIGD, '1') <>'0')

AND B.TRANGTHAIHUYGD = '9'

AND B.Time\_Out is null

AND T.MODULE in ('DL','EVNHCM','EVNCPC','EVNSPC', 'EVNHNI','EVNNPC')

AND T.MAKER\_ID = maker\_ID

AND T.TRN\_BRN = branchCode

AND b.TRN\_ID = to\_char(t.TRANSACTION\_ID)

and t.EVENT\_CODE='INIT'

AND T.RECORD\_STATUS = 'O'

AND T.AUTH\_STATUS = 'A'

db.SelectCommand(SQL\_CMD, dt\_HoaDon)

return dt\_HoaDon

Bước 8: Gọi service gạch nợ bổ sung :

EVN\_auto\_Service.Service1.GachNoBoSungAll\_Ver2() 🡪 EVN\_Library.EVN\_AUTO.GachNoBoSungAll()🡪EVNHCM\_Library.GachNoBoSung(

🡪 LVBService.BankConfirm() //Gạch nợ

Bước 12: Lưu trạng thái vào DB :

TBL: DLTBD\_HOADON\_INFO

Thành công : TRANGTHAIGD =0;

TRANGTHAIHUYGD =9;

NGAYGIO = DateTime.Now

where TRN\_ID = Trn\_ID

Thất bại : TRANGTHAIGD =1;

TRANGTHAIHUYGD =9;

where TRN\_ID = Trn\_ID

Exception : TRANGTHAIGD =2;

TRANGTHAIHUYGD =9;

NGAYGIO = DateTime.Now

where TRN\_ID = Trn\_ID

Bước 13 : Gọi service Update: updateAutoLogVer2(string Trn\_ID, string status, string coreStatus, string pgStatus)

Bước 14: Lưu kết quả vào DB

Update DLTBD\_AUTOPAY\_DETAILS

set AUTO\_STATUS = status,

POSTCORE\_MSG = coreStatus, //if coreStatus!= null or empty

PAYBILLS\_MSG = pgStatus //if pgStatus !=null or empty

where TRANSACTION\_ID = Trn\_ID

Bước 17 : Như bước 2

1. Chăn thanh toán theo CIF



Bước 3: Gọi service thêm mới :

string[] arr = (string[])Session["up\_evn"];

- data createXMLData(ddl\_SearchEVN.SelectedValue,txt\_FromDT.Text,txt\_ToDT.Text,DateTime.Now.Year.ToString(), txt\_CIF.Text.Trim(), "O")

- EVN\_AUTO.Service1.insertParam( arr[2],data )

Bước 6 Lưu dữ liệu vào DB:

Package : PKG\_DLTBD\_TQ

PROCEDURE SP\_INSERT\_PARAM(

p\_UserID varchar2,

p\_Data xmltype,

p\_Err\_String out varchar2

)

v\_ID varchar2(30)

v\_CountLoop number

for r in (

SELECT a.\*

FROM XMLTABLE ('/LIST/INFO'

PASSING p\_data

COLUMNS

EVN\_TYPE varchar2(100) PATH '/INFO/EVN\_TYPE',

FROM\_DT VARCHAR2(100) PATH '/INFO/FROM\_DT',

TO\_DT VARCHAR2(100) PATH '/INFO/TO\_DT',

YEAR\_APPLY varchar2(10) PATH '/INFO/YEAR\_APPLY',

CIF VARCHAR2(500) PATH '/INFO/CIF',

STATUS VARCHAR2(10) PATH '/INFO/STATUS'

)a

select count(\*) into v\_CountLoop from LVBSMS.DLTBD\_PARAMETERS

where CIF = r.CIF

and RECORD\_STAT='O'

and ( trunc(to\_date(r.FROM\_DT,'dd/mm/yyyy hh24:mi:ss')) between trunc(FROM\_DT) and trunc(TO\_DT)

or trunc(to\_date(r.TO\_DT,'dd/mm/yyyy hh24:mi:ss')) between trunc(FROM\_DT) and trunc(TO\_DT) )

if(v\_CountLoop > 0) then

p\_Err\_String := '1#98#CIF da bi chan trong khoang thoi gian ' || r.FROM\_DT || ' va ' || r.TO\_DT;

end if

INSERT INTO LVBSMS.DLTBD\_PARAMETERS(

ID,

FROM\_DT,

TO\_DT,

EVN\_TYPE,

YEAR\_APPLY,

CIF,

RECORD\_STAT,

MAKER\_ID,

MAKER\_DT

)

VALUES(

v\_ID,

to\_date(r.FROM\_DT, 'dd/mm/yyyy hh24:mi:ss'),

to\_date(r.TO\_DT, 'dd/mm/yyyy hh24:mi:ss'),

r.EVN\_TYPE,

r.YEAR\_APPLY,

r.CIF,

r.STATUS,

p\_UserID,

sysdate

)

p\_Err\_String := '0#00#Successful

Bước 8: Gọi service lấy list chặn thanh toán :

EVN\_AUTO\_service.Service1.listParam(string yearApply, string evnType, string cif)

Bước 10 Lấy thông tin trong DB:

condition = "and cif like '%" + cif + "%'";

select a.ID, a.CIF, a.RECORD\_STAT, a.MAKER\_ID, a.MAKER\_DT, a.CHECKER\_ID,a.CHECKER\_DT, a.EVN\_TYPE, to\_char(a.FROM\_DT, 'dd/mm/yyyy') as FROM\_DT , to\_char(a.TO\_DT, 'dd/mm/yyyy') as TO\_DT

,decode(a.RECORD\_STAT, 'O', 'Đang áp dụng', 'C', 'Đóng') as RECORD\_STATUS\_NAME

from LVBSMS.DLTBD\_PARAMETERS a

where a.YEAR\_APPLY = yearApply and a.EVN\_TYPE = evnType + condition + order by a.ID desc

Bước 14: Giống bước 8

Bước 16 giống bước 10

Bước 19: Gọi service hủy chặn thanh toán :

EVN\_AUTO\_Service.Service1.updateStatus(id, arr[2], "C")

Bước 21 cập nhật DB :

update LVBSMS.DLTBD\_PARAMETERS

set RECORD\_STAT = 'C',

checker\_id=arr[2],

checker\_dt=sysdate

where ID = id

Bước 26: Gọi service update :

EVN\_AUTO\_Service.Service1. updateParam(hd\_IdParam.Value, arr[2], txt\_FromDT.Text.Trim(), txt\_ToDT.Text.Trim(), DateTime.Now.Year.ToString(), txt\_CIF.Text.Trim(), string.Empty)

//Check

select \* from LVBSMS.DLTBD\_PARAMETERS

where CIF = 'cifNo'

and RECORD\_STAT='O' and ID != 'paramID'

and ( trunc(to\_date('fromDate','dd/mm/yyyy')) between trunc(FROM\_DT) and trunc(TO\_DT)

or trunc(to\_date('toDate','dd/mm/yyyy')) between trunc(FROM\_DT) and trunc(TO\_DT)

Bước 28 Lưu vào DB :

update LVBSMS.DLTBD\_PARAMETERS

set FROM\_DT = to\_date(' fromDate 00:00:00', 'dd/mm/yyyy hh24:mi:ss'),

TO\_DT = to\_date(' toDate 23:59:00', 'dd/mm/yyyy hh24:mi:ss'),

YEAR\_APPLY = ' yearApply ',

CIF = ' cifNo ',

CHECKER\_ID=' userID ', CHECKER\_DT = sysdate

where ID = 'paramID'

Bước 30 Giống bước 8

# **Tự Động**

1. **Đăng ký thu tự động**



**Step** **2**: Gọi Service lấy thông tin khách hàng

* Gọi Service1.getCustomerFromDienLuc(sMaKH, sMaDiemThu, sNCC)

→ Gọi EVNHCM\_SVC.LVBService.BankRequest(MaKH, MaDiemThu, BankID)

* Return:
  + dt[khachhang]
    - Makh, tenkh, diachikh, madl, masothuekh
  + dt[hoadon]
    - mahd, sotien, mota, tungay, denngay, giabieu, hoadonid, soho, magiaodich, DNTT, tiendien, tienthue, thuesuat, kyhieuhd, seryhd, HDDT, NCC

**Step 8:** Gọi Service check auto regist

* Gọi Service1.CheckAutoRegisterToCreat(String MaKH, String MaDL)

**Step 10:** Check auto regist

* Select \* from
  + Select from DLTBD\_MN\_AUTO\_REGISTER R
    - Columns: R.\*
    - Where: RECORD\_STAT <> 'C'  
       and (Close\_DT is null or Close\_DT>= sysdate
  + Union all select from DLTBD\_MB\_AUTO\_REGISTER R
    - Columns: R.\*
    - Where: RECORD\_STAT <> 'C'  
       and (Close\_DT is null or Close\_DT>= sysdate
  + Union all select from DLTBD\_AUTO\_REGISTER R
    - Columns: R.\*
    - Where: RECORD\_STAT <> 'C'  
       and (Close\_DT is null or Close\_DT>= sysdate
  + Union all select from DLTBD\_HN\_AUTO\_REGISTER R
    - Columns: R.\*
    - Where: RECORD\_STAT <> 'C'  
       and (Close\_DT is null or Close\_DT>= sysdate
  + Union all select from DLTBD\_MT\_AUTO\_REGISTER R
    - Columns: R.\*
    - Where: RECORD\_STAT <> 'C'  
       and (Close\_DT is null or Close\_DT>= sysdate

**Step 14:** Gọi Service lấy thông tin tài khoản khách hàng

* Gọi Service Service1.getCustAccFromFlexCube(Customer\_No)

**Step 16:** Lấy thông tin tài khoản khách hàng

* dt[customer]:
  + Select from table: sttm\_customer s left join sttm\_cust\_personal p on s.customer\_no = p.customer\_no
  + Column: s.CUSTOMER\_NO,s.CUSTOMER\_NAME1, s.UNIQUE\_ID\_VALUE,s.UDF\_1 NGAY\_CAP,s.UDF\_2 NOI\_CAP, P.TELEPHONE, s.ADDRESS\_LINE1 || s.ADDRESS\_LINE2 || s.ADDRESS\_LINE3 || s.ADDRESS\_LINE4 ADDRESS
  + Where: s.CUSTOMER\_NO = '" + Customer\_No + "' and record\_stat = 'O'
* dt[account]:
  + Select from table: sttm\_cust\_account c
  + Column: c.CUST\_AC\_NO,c.AC\_DESC,c.BRANCH\_CODE, ACY\_AVL\_BAL
  + Where: c.ACCOUNT\_CLASS IN ('V0CN30', 'V0TKCN') OR c.CR\_GL LIKE '4211%' OR c.cust\_no IN (SELECT B.WALKIN\_CUSTOMER FROM sttm\_branch b WHERE B.RECORD\_STAT = 'O')) AND c.RECORD\_STAT = 'O' AND c.CUST\_NO ='" + Customer\_No + "' " AND c.ACCOUNT\_CLASS IN ('V0CN30', 'V0CD26', 'V0CD45', 'V0CNTN', 'V0DNTN', 'V0ID44', 'VMWALK', 'V0TKCN') AND c.CCY = 'VND'"

**Step 20**: Gọi Service thêm đăng ký thu tự động

* Gọi Service1. ImportAutoRegister()

**Step 22:** Thêm đăng ký thu tự động

Package: pkg\_EVN\_HCM.DLP\_IMPORTAUTOREGISTER

Parameter:

p\_MAKER\_ID  
p\_MAKH  
p\_AC\_NO  
p\_PAY\_DT  
p\_RECEIVE\_SMS  
p\_RECEIVE\_EMAIL  
p\_PHONE  
p\_EMAIL  
p\_REG\_BRANCH  
p\_REG\_DATE  
p\_CLOSE\_DATE  
p\_PRINT\_BILL  
p\_KHACHHANG\_INFO xmltype  
p\_Err\_String

Variable:

v\_Home\_Branch  
v\_Count

* Insert into table DLTBD\_AUTO\_REGISTER
  + Columns: MAKH, CIF, AC\_NO, PAY\_DT, RECEIVE\_SMS, RECEIVE\_EMAIL, PHONE, EMAIL, REG\_BRANCH, REG\_DT, CLOSE\_DT, MAKER\_ID, MAKER\_DT, RECORD\_STAT, LAST\_AUDIT\_VER, PRINT\_BILL
  + Values: upper(p\_MAKH), p\_CIF, p\_AC\_NO, p\_PAY\_DT, p\_RECEIVE\_SMS, p\_RECEIVE\_EMAIL, p\_PHONE, p\_EMAIL, p\_REG\_BRANCH, to\_date(NVL(p\_REG\_DATE, sysdate), 'dd/mm/yyyy'), to\_date(p\_CLOSE\_DATE, 'dd/mm/yyyy'), p\_MAKER\_ID, SYSDATE, 'I', 0, p\_PRINT\_BILL
* Insert into table DLTBD\_AUTO\_REGISTER\_HIST
  + Columns: MAKH, CIF, AC\_NO, PAY\_DT, RECEIVE\_SMS, RECEIVE\_EMAIL, PHONE, EMAIL, REG\_BRANCH, REG\_DT, CLOSE\_DT, MAKER\_ID, MAKER\_DT, RECORD\_STAT, AUDIT\_VER, PRINT\_BILL
  + Values: upper(p\_MAKH), p\_CIF, p\_AC\_NO, p\_PAY\_DT, p\_RECEIVE\_SMS, p\_RECEIVE\_EMAIL, p\_PHONE, p\_EMAIL, p\_REG\_BRANCH, to\_date(NVL(p\_REG\_DATE, sysdate), 'dd/mm/yyyy'), to\_date(p\_CLOSE\_DATE, 'dd/mm/yyyy'), p\_MAKER\_ID, SYSDATE, 'I', 0, p\_PRINT\_BILL
* For r in xmltable (' /KhachHang')
  + Update table DLTB\_CUSTOMER\_INFO
    - Set: TENKH = r.TENKH

, DIACHIKH = r.DIACHIKH

, MADL = r.MADL

, MASOTHUEKH = r.MASOTHUE

, PHIEN = r.PHIEN

, LOTRINH = r.LOTRINH

, SOGHICSCMIS = r.SOGHICSCMIS

, DANHSO = r.DANHSO

, SOCONGTO = r.SOCONGTO

, NGANHNGHE = r.NGANHNGHE

* + Nếu ko update được
    - Insert into table DLTB\_CUSTOMER\_INFO
    - Columns: MAKH, TENKH, DIACHIKH, MADL, MASOTHUEKH, PHIEN, LOTRINH, SOGHICSCMIS, DANHSO, SOCONGTO, NGANHNGHE
    - Values: trim(r.MAKH), r.TENKH, r.DIACHIKH, r.MADL, r.MASOTHUE, r.PHIEN, r.LOTRINH, r.SOGHICSCMIS, r.DANHSO, r.SOCONGTO, r.NGANHNGHE

1. **Duyệt đăng ký thu tự động**



**Step 2:** Gọi Service lấy list đăng ký

* Gọi Service1. getData4AuthAutoRegist(Checker\_ID, Checker\_BranchID)

**Step 4:** Select list đăng ký

* Select from: DLTBD\_AUTO\_REGISTER A
  + Columns: A.ROWID ROW\_ID, A.\*, RECORD\_STAT\_DESC, 'EVNHCM' NCC
  + Where: A.CHECKER\_ID IS NULL  
     AND (A.RECORD\_STAT = 'I' OR RECORD\_STAT='C')  
     AND A.MAKER\_ID <> '" + Checker\_ID + "'  
     AND EXISTS(SELECT 1 FROM LVBSMS.ASTB\_USER U WHERE U.USERNAME = A.MAKER\_ID AND U.HOME\_BRANCH = '" + Checker\_BranchID + "')
* Union all Select from DLTBD\_MT\_AUTO\_REGISTER
  + Columns: A.ROWID ROW\_ID, A.\*, RECORD\_STAT\_DESC, 'EVNCPC' NCC
  + Where: A.CHECKER\_ID IS NULL  
     AND (A.RECORD\_STAT = 'I' OR RECORD\_STAT='C')  
     AND A.MAKER\_ID <> '" + Checker\_ID + "'  
     AND EXISTS(SELECT 1 FROM LVBSMS.ASTB\_USER U WHERE U.USERNAME = A.MAKER\_ID AND U.HOME\_BRANCH = '" + Checker\_BranchID + "')
* Union all Select from DLTBD\_MB\_AUTO\_REGISTER
  + Columns: A.ROWID ROW\_ID, A.\*, RECORD\_STAT\_DESC, 'EVNCPC' NCC
  + Where: A.CHECKER\_ID IS NULL  
     AND (A.RECORD\_STAT = 'I' OR RECORD\_STAT='C')  
     AND A.MAKER\_ID <> '" + Checker\_ID + "'  
     AND EXISTS(SELECT 1 FROM LVBSMS.ASTB\_USER U WHERE U.USERNAME = A.MAKER\_ID AND U.HOME\_BRANCH = '" + Checker\_BranchID + "')
* Union all Select from DLTBD\_HN\_AUTO\_REGISTER
  + Columns: A.ROWID ROW\_ID, A.\*, RECORD\_STAT\_DESC, 'EVNCPC' NCC
  + Where: A.CHECKER\_ID IS NULL  
     AND (A.RECORD\_STAT = 'I' OR RECORD\_STAT='C')  
     AND A.MAKER\_ID <> '" + Checker\_ID + "'  
     AND EXISTS(SELECT 1 FROM LVBSMS.ASTB\_USER U WHERE U.USERNAME = A.MAKER\_ID AND U.HOME\_BRANCH = '" + Checker\_BranchID + "')
* Union all Select from DLTBD\_MN\_AUTO\_REGISTER
  + Columns: A.ROWID ROW\_ID, A.\*, RECORD\_STAT\_DESC, 'EVNCPC' NCC
  + Where: A.CHECKER\_ID IS NULL  
     AND (A.RECORD\_STAT = 'I' OR RECORD\_STAT='C')  
     AND A.MAKER\_ID <> '" + Checker\_ID + "'  
     AND EXISTS(SELECT 1 FROM LVBSMS.ASTB\_USER U WHERE U.USERNAME = A.MAKER\_ID AND U.HOME\_BRANCH = '" + Checker\_BranchID + "')

RECORD\_STAT\_DESC = DECODE(NVL(LAST\_AUDIT\_VER,0), 0, 'Tạo mới','Chỉnh sửa'), 'Đóng')

**Step 8:** Gọi Service duyệt đăng ký

* Gọi Service1.authAutoRegistAll(MaKH, Checker\_ID, NCC)

**Step 10:**

Package:pkg\_EVN\_HCM.DLP\_authAutoRegist

Parameters:

p\_MaKH,  
p\_Checker\_ID,  
p\_Err\_String

Variables:

c\_Function\_Name   
v\_Home\_Branch  
v\_Status  
v\_Rcd\_Err

* Update table DLTBD\_AUTO\_REGISTER A
  + Set: A.CHECKER\_DT = sysdate,  
     A.CHECKER\_ID = p\_Checker\_ID,  
     A.RECORD\_STAT = DECODE( A.RECORD\_STAT, 'I', 'O', RECORD\_STAT)
  + Where: A.MAKH = p\_MaKH  
     and A.MAKER\_ID <> p\_Checker\_ID

1. **Hạch toán hóa đơn tự động**



**Step 2:** Gọi Service lấy list khách hàng đã đăng ký thanh toán tự động

* Gọi Service1.getCustomerForAutoPay(pay\_day, branch\_code, maxRow.ToString(), PageNum)

**Step 4:** Select list khách hàng

* Select \* from Select List\_KH.\*, rownum as Rn
  + From Select pay\_list.\*, ACY\_AVL\_BAL
    - From Pay\_list

Pay\_list = Select A.\*, 'EVNHCM' NCC, HDDT  
 From DLTBD\_AUTO\_REGISTER A  
 Where A.REG\_BRANCH = '" + reg\_branch + @"'  
 and A.RECORD\_STAT = 'O'  
 and A.CHECKER\_ID is not null  
 and trunc(sysdate) between trunc(NVL(A.REG\_DT, sysdate)) and NVL(trunc(A.CLOSE\_DT), to\_date('09/09/2099', 'dd/mm/yyyy'))  
 and case when A.PAY\_DT >   
 to\_number(to\_char(LAST\_DAY(to\_date('" + pay\_day +   
 @"','dd/mm/yyyy')),'dd'))   
 then LAST\_DAY(to\_date('" + pay\_day + @"','dd/mm/yyyy'))

else to\_date(lpad(A.PAY\_DT,2,'0') || '/' || substr('" + pay\_day + @"',4,7),'dd/mm/yyyy')

end

between

DECODE(to\_char(to\_date('" + pay\_day +   
 @"','dd/mm/yyyy'),'D'),'2', to\_date('" + pay\_day +   
 @"','dd/mm/yyyy')-1 , to\_date('" + pay\_day +   
 @"','dd/mm/yyyy') )   
 and to\_date('" + pay\_day + @"','dd/mm/yyyy')

* Union all Select pay\_list.\*, ACY\_AVL\_BAL
  + From Pay\_list
* Pay\_list = Select A.\*, ' EVNCPC' NCC, HDDT  
   From DLTBD\_MT\_AUTO\_REGISTER A  
   Where A.REG\_BRANCH = '" + reg\_branch + @"'  
   and A.RECORD\_STAT = 'O'  
   and A.CHECKER\_ID is not null  
   and trunc(sysdate) between trunc(NVL(A.REG\_DT, sysdate)) and NVL(trunc(A.CLOSE\_DT), to\_date('09/09/2099', 'dd/mm/yyyy'))  
   and case when A.PAY\_DT >   
   to\_number(to\_char(LAST\_DAY(to\_date('" + pay\_day +   
   @"','dd/mm/yyyy')),'dd'))   
   then LAST\_DAY(to\_date('" + pay\_day + @"','dd/mm/yyyy'))

else to\_date(lpad(A.PAY\_DT,2,'0') || '/' || substr('" + pay\_day + @"',4,7),'dd/mm/yyyy')

end

between

DECODE(to\_char(to\_date('" + pay\_day +   
 @"','dd/mm/yyyy'),'D'),'2', to\_date('" + pay\_day +   
 @"','dd/mm/yyyy')-1 , to\_date('" + pay\_day +   
 @"','dd/mm/yyyy') )   
 and to\_date('" + pay\_day + @"','dd/mm/yyyy')

* Union all Select pay\_list.\*, ACY\_AVL\_BAL
  + From Pay\_list
* Pay\_list = Select A.\*, ' EVNCPC' NCC, HDDT  
   From DLTBD\_MB\_AUTO\_REGISTER A  
   Where A.REG\_BRANCH = '" + reg\_branch + @"'  
   and A.RECORD\_STAT = 'O'  
   and A.CHECKER\_ID is not null  
   and trunc(sysdate) between trunc(NVL(A.REG\_DT, sysdate)) and NVL(trunc(A.CLOSE\_DT), to\_date('09/09/2099', 'dd/mm/yyyy'))  
   and case when A.PAY\_DT >   
   to\_number(to\_char(LAST\_DAY(to\_date('" + pay\_day +   
   @"','dd/mm/yyyy')),'dd'))   
   then LAST\_DAY(to\_date('" + pay\_day + @"','dd/mm/yyyy'))

else to\_date(lpad(A.PAY\_DT,2,'0') || '/' || substr('" + pay\_day + @"',4,7),'dd/mm/yyyy')

end

between

DECODE(to\_char(to\_date('" + pay\_day +   
 @"','dd/mm/yyyy'),'D'),'2', to\_date('" + pay\_day +   
 @"','dd/mm/yyyy')-1 , to\_date('" + pay\_day +   
 @"','dd/mm/yyyy') )   
 and to\_date('" + pay\_day + @"','dd/mm/yyyy')

* Union all Select pay\_list.\*, ACY\_AVL\_BAL
  + From Pay\_list
* Pay\_list = Select A.\*, ' EVNCPC' NCC, HDDT  
   From DLTBD\_HN\_AUTO\_REGISTER A  
   Where A.REG\_BRANCH = '" + reg\_branch + @"'  
   and A.RECORD\_STAT = 'O'  
   and A.CHECKER\_ID is not null  
   and trunc(sysdate) between trunc(NVL(A.REG\_DT, sysdate)) and NVL(trunc(A.CLOSE\_DT), to\_date('09/09/2099', 'dd/mm/yyyy'))  
   and case when A.PAY\_DT >   
   to\_number(to\_char(LAST\_DAY(to\_date('" + pay\_day +   
   @"','dd/mm/yyyy')),'dd'))   
   then LAST\_DAY(to\_date('" + pay\_day + @"','dd/mm/yyyy'))

else to\_date(lpad(A.PAY\_DT,2,'0') || '/' || substr('" + pay\_day + @"',4,7),'dd/mm/yyyy')

end

between

DECODE(to\_char(to\_date('" + pay\_day +   
 @"','dd/mm/yyyy'),'D'),'2', to\_date('" + pay\_day +   
 @"','dd/mm/yyyy')-1 , to\_date('" + pay\_day +   
 @"','dd/mm/yyyy') )   
 and to\_date('" + pay\_day + @"','dd/mm/yyyy')

* Union all Select pay\_list.\*, ACY\_AVL\_BAL
  + From Pay\_list
* Pay\_list = Select A.\*, ' EVNCPC' NCC, HDDT  
   From DLTBD\_MN\_AUTO\_REGISTER A  
   Where A.REG\_BRANCH = '" + reg\_branch + @"'  
   and A.RECORD\_STAT = 'O'  
   and A.CHECKER\_ID is not null  
   and trunc(sysdate) between trunc(NVL(A.REG\_DT, sysdate)) and NVL(trunc(A.CLOSE\_DT), to\_date('09/09/2099', 'dd/mm/yyyy'))  
   and case when A.PAY\_DT >   
   to\_number(to\_char(LAST\_DAY(to\_date('" + pay\_day +   
   @"','dd/mm/yyyy')),'dd'))   
   then LAST\_DAY(to\_date('" + pay\_day + @"','dd/mm/yyyy'))

else to\_date(lpad(A.PAY\_DT,2,'0') || '/' || substr('" + pay\_day + @"',4,7),'dd/mm/yyyy')

end

between

DECODE(to\_char(to\_date('" + pay\_day +   
 @"','dd/mm/yyyy'),'D'),'2', to\_date('" + pay\_day +   
 @"','dd/mm/yyyy')-1 , to\_date('" + pay\_day +   
 @"','dd/mm/yyyy') )   
 and to\_date('" + pay\_day + @"','dd/mm/yyyy')

ACY\_AVL\_BAL = SELECT ACY\_AVL\_BAL - nvl(min\_balance,0)  
 FROM XMLTABLE (  
 '/FCUBS\_BODY/Get\_Account\_List\_IO/ACCOUNT\_LIST/ACCOUNT'  
 PASSING xmltype (   
 ebank.pkg\_core\_switch.  
 loadValBal (pay\_list.AC\_NO, 1, 1))  
 COLUMNS ACY\_AVL\_BAL VARCHAR2 (200) PATH 'ACY\_AVL\_BAL',  
 min\_balance VARCHAR2 (200) PATH 'MIN\_BALANCE')

HDDT = case when TRIM(A.PRINT\_BILL) = '1' AND SUBSTR(MAKH, 0, 4) = 'PE01' then '1' else '0' end

**Step 7:** Gọi Service lấy thông tin khách hàng

Gọi Service Service1.getCustomerFromDienLuc\_getBill(MaKH,MaDiemThu,NCC)

→ EVNHCM\_SVC.LVBService.BankRequest(MaKH, MaDiemThu, BankID)

* Return:
  + dt[khachhang]
    - Makh, tenkh, diachikh, madl, masothuekh
  + dt[hoadon]
    - mahd, sotien, mota, tungay, denngay, giabieu, hoadonid, soho, magiaodich, DNTT, tiendien, tienthue, thuesuat, kyhieuhd, seryhd, HDDT, NCC

**Step 16:** Gọi Service tạo giao dịch

* Gọi Service Service1.importTransaction\_Auto()

**Step 18:** Lấy số hóa đơn trong hệ thống

* Lấy số hóa đơn trong hệ thống, thêm hóa đơn vào database

Package: pkg\_EVN\_HCM.dlf\_getMaGiaoDich\_auto

Parameter:

p\_Maker\_ID  
 p\_HoaDon\_Info xmltype

Variables:

v\_Rcd\_HD  
 v\_Home\_Branch  
 v\_Sign  
 v\_Year  
 v\_Min\_ID  
 v\_Max\_ID  
 v\_current\_id

v\_MaGiaoDich  
 v\_Rcd\_Err  
 v\_Sys\_Date

* For r in xmltable('/HoaDon/HD\_Info')
  + If r.DAINHD = '1'
    - If r.HDDT = 1
      * Insert into DLTBD\_HOADON\_INFO
      * Columns: MAHD, MAKH, SOTIEN, MOTA, TUNGAY, DENNGAY, GIABIEU, HOADONID, SOHO, NGAYGIO, MAGIAODICH, BANKID,DAINHD,TRN\_CHANEL,YEAR,HOADON\_ID,DNTT,TIENDIEN, TIENTHUE, THUESUAT, KYHIEUHD, SERYHD, HDDT
      * Values: r.MAHD, trim(r.MAKH), r.SOTIEN,r.MOTA, r.TUNGAY,r.DENNGAY,r.GIABIEU,r.HOADONID,r.SOHO, v\_Sys\_Date, r.MaGiaoDich, c\_Bank\_ID, '0', c\_Chanel\_AUTO, v\_Year, SUBSTR(r.MaGiaoDich,1,8), r.DNTT,r.TIENDIEN, r.TIENTHUE, r.THUESUAT, r.KYHIEUHD, r.SERYHD, r.HDDT
      * Return: rowid, MAHD,MAKH,SOTIEN,HOADONID,MAGIAODICH
      * Into: v\_Rcd\_HD.row\_id, v\_Rcd\_HD.MAHD, v\_Rcd\_HD.MAKH, v\_Rcd\_HD.SOTIEN, v\_Rcd\_HD.HOADONID, v\_Rcd\_HD.MAGIAODICH
    - Else If r.MaHD like '%2' //HD phat
      * Insert into DLTBD\_HOADON\_INFO
      * Columns: MAHD, MAKH, SOTIEN, MOTA, TUNGAY, DENNGAY, GIABIEU, HOADONID, SOHO, NGAYGIO, BANKID,DAINHD, TRN\_CHANEL, DNTT, TIENDIEN, TIENTHUE, THUESUAT, KYHIEUHD, SERYHD, HDDT
      * Values: r.MAHD, trim(r.MAKH), r.SOTIEN,r.MOTA, r.TUNGAY,r.DENNGAY,r.GIABIEU,r.HOADONID,r.SOHO, v\_Sys\_Date, c\_Bank\_ID, '0', c\_Chanel\_AUTO, r.DNTT,r.TIENDIEN, r.TIENTHUE, r.THUESUAT, r.KYHIEUHD, r.SERYHD, r.HDDT
      * Return: rowid, MAHD, MAKH, SOTIEN, HOADONID
      * Values: v\_Rcd\_HD.row\_id, v\_Rcd\_HD.MAHD, v\_Rcd\_HD.MAKH, v\_Rcd\_HD.SOTIEN, v\_Rcd\_HD.HOADONID;
    - Else
      * Select from dual
        + Columns: LVBSMS.SEQ\_PG\_030002.NEXTVAL
        + Into: v\_Rcd\_HD.HOADON\_ID
      * Insert into DLTBD\_HOADON\_INFO
      * Columns: MAHD, MAKH, SOTIEN, MOTA, TUNGAY, DENNGAY, GIABIEU, HOADONID, SOHO, NGAYGIO, MAGIAODICH, BANKID,DAINHD, TRN\_CHANEL, YEAR, HOADON\_ID, DNTT, TIENDIEN, TIENTHUE, THUESUAT, KYHIEUHD, SERYHD, HDDT
      * Values: r.MAHD, trim(r.MAKH), r.SOTIEN,r.MOTA, r.TUNGAY,r.DENNGAY,r.GIABIEU,r.HOADONID,r.SOHO, v\_Sys\_Date, lpad(v\_Rcd\_HD.HOADON\_ID,10, '0') || v\_Sign, c\_Bank\_ID, '0', c\_Chanel\_AUTO, v\_Year, v\_Rcd\_HD.HOADON\_ID, r.DNTT,r.TIENDIEN, r.TIENTHUE, r.THUESUAT, r.KYHIEUHD, r.SERYHD, r.HDDT
      * Return: rowid, MAHD,MAKH,SOTIEN,HOADONID,MAGIAODICH
      * Into: v\_Rcd\_HD.row\_id, v\_Rcd\_HD.MAHD, v\_Rcd\_HD.MAKH, v\_Rcd\_HD.SOTIEN, v\_Rcd\_HD.HOADONID, v\_Rcd\_HD.MAGIAODICH;
  + Else //Khong in hoa don
    - Select from dual
      * Columns: EBANK.SEQ\_PG\_030002.NEXTVAL
      * Into v\_Rcd\_HD.HOADON\_ID

Return: v\_Rcd\_HD;

**Step 20:** Lưu thông tin giao dịch

Gọi package pkg\_EVN\_HCM.DLP\_importTransaction\_auto

Parameters:

p\_Maker\_ID  
 p\_Trn\_Desc  
 p\_KhachHang\_Info  
 p\_HoaDon\_Info  
 p\_HachToan\_Info  
 p\_Trn\_ID  
 p\_Err\_String

Constants:

c\_txn\_code  
 c\_source\_code

Variables:

v\_SQL\_CMD  
 v\_Home\_Branch  
 v\_Trn\_Ref\_No  
 v\_Trn\_Dt  
 v\_Check  
 v\_Amount  
 v\_Cr\_Ac\_No  
 v\_Cr\_Ac\_Branch  
 v\_Db\_Ac\_No  
 v\_Db\_Ac\_Branch  
 v\_CCY  
 V\_BRANCH  
 V\_COUNT

* p\_Trn\_ID := pkg\_Global\_Dom\_Application.get\_Dom\_Transaction\_ID();
* Insert into: DATBD\_TRANSACTION
  + Columns: TRANSACTION\_ID, TRN\_BRN, TRN\_DESC, TRN\_DT, VALUE\_DT, TRN\_CODE, MODULE, EVENT\_CODE, RECORD\_STATUS, MAKER\_ID, MAKER\_DT, APP\_TYPE, LAST\_EVENT\_SEQ\_NO, XREF, COMMON\_USER
  + Values: p\_Trn\_ID, v\_Home\_Branch, p\_Trn\_Desc, v\_Trn\_Dt, trunc(sysdate), 'AUTO\_LIQUID', 'DL', 'INIT', 'O', p\_Maker\_ID, sysdate, 'DOM\_WEB', 1 ,'EBK'||loadESBMsgId(), c\_Common\_User
* For r in xmltable ('/Transaction/Post\_Info')
  + SELECT COUNT(\*) INTO V\_COUNT
    - FROM STTM\_CUST\_ACCOUNT
    - WHERE CUST\_AC\_NO = R.AC\_NO
    - AND RECORD\_STAT = 'O' AND AUTH\_STAT ='A'
  + If V\_COUNT >0
    - SELECT BRANCH\_CODE INTO V\_BRANCH FROM STTM\_CUST\_ACCOUNT
  + Insert into DLTBD\_TRANSACTION\_POST
    - Columns: TRANSACTION\_ID, AC\_NO, AC\_CCY, AC\_BRANCH, CUST\_GL, DRCR\_IND, AMOUNT, AMOUNT\_TAG, TRN\_CODE, EVENT, EVENT\_SEQ\_NO
    - Values: p\_Trn\_ID, r.Ac\_No, 'VND', V\_BRANCH, 'A', R.DrCr\_Ind, R.Amount, R.Amount\_Tag, 'AUTO\_LIQUID', 'INIT',1
* For r in xmltable ('/ KhachHang\_Info')
  + Update: DLTB\_CUSTOMER\_INFO
  + Set: TENKH = r.TENKH,  
     DIACHIKH = r.DIACHIKH,  
     MADL = r.MADL,  
     MASOTHUEKH = r.MASOTHUE,  
     PHIEN = r.PHIEN,  
     LOTRINH = r.LOTRINH,  
     SOGHICSCMIS = r.SOGHICSCMIS,  
     DANHSO = r.DANHSO,  
     SOCONGTO = r.SOCONGTO,  
     NGANHNGHE = r.NGANHNGHE
  + Nếu không update được:
    - Insert into DLTB\_CUSTOMER\_INFO
      * Columns: MAKH, TENKH, DIACHIKH, MADL, MASOTHUEKH, PHIEN, LOTRINH, SOGHICSCMIS, DANHSO, SOCONGTO, NGANHNGHE
      * Values: trim(r.MAKH), r.TENKH, r.DIACHIKH, r.MADL, r.MASOTHUE, r.PHIEN, r.LOTRINH, r.SOGHICSCMIS, r.DANHSO, r.SOCONGTO, r.NGANHNGHE
* For r in xmltable ('/HoaDon\_Info')
  + Update DLTBD\_HOADON\_INFO
    - Set: H.TRN\_ID = p\_Trn\_ID

1. **Duyệt hóa đơn thu tự động**



**Step 2:** Gọi Service load list giao dịch chờ duyệt

* Gọi Service1.getTransactionsToAuth\_Auto(checkerID)

**Step 4:** Select list giao dịch chờ duyệt

* Select from DATBD\_TRANSACTION d, "STTM\_DATES" + DB\_Link + " e "
  + Columns: d.\*, (case d.MODULE WHEN 'DL' THEN 'EVNHCM' ELSE d.MODULE END) NCC
  + Where: D.TRN\_BRN = '" + Branch\_Code + "' "   
     and D.EVENT\_CODE = 'INIT'   
     and d.MODULE in ('DL', 'EVNHCM', 'EVNCPC', 'EVNSPC', 'EVNHNI', 'EVNNPC')   
     and d.TRN\_CODE = 'AUTO\_LIQUID'   
     and D.CHECKER\_ID is null   
     and D.RECORD\_STATUS = 'O'   
     and D.AUTH\_STATUS is null   
     and D.TRN\_DT = E.TODAY   
     and D.TRN\_BRN = E.BRANCH\_CODE   
     and D.MAKER\_DT >= SYSDATE - " + sDATE
  + order by D.TRANSACTION\_ID DESC"
* Return: dt\

**Step 8:** Gọi Service duyệt giao dịch

* Gọi Service1.authTransaction(Checker\_ID, Trn\_ID, sMaDL)

**Step 11+13:**

Gọipkg\_EVN\_HCM.DLP\_authTransaction

Parameter:

p\_Checker\_ID  
p\_Trn\_ID  
p\_NgayGio  
p\_Err\_String

Constant:

c\_DB\_Link  
 c\_txn\_code  
 c\_Function\_Name

Variable:

v\_Home\_Branch  
 v\_Trn\_Desc  
 v\_Maker\_ID  
 v\_Rcd\_Err  
 v\_Value\_Date  
 v\_Last\_Event\_Seq\_No  
 v\_MSGID  
 v\_Rcd\_Trn

1. Get Transaction to Authorize

* Update table DATBD\_TRANSACTION d
  + Set: d.AUTH\_STATUS = 'A',  
     d.CHECKER\_DT = sysdate,  
     d.CHECKER\_ID = p\_Checker\_ID
* Select from table DATBD\_TRANSACTION d
  + Columns: d.TRN\_DESC, d.MAKER\_ID, d.LAST\_EVENT\_SEQ\_NO
  + Returning into: v\_Trn\_Desc, v\_Maker\_ID, v\_Last\_Event\_Seq\_No
* Where: D.TRANSACTION\_ID = p\_Trn\_ID

and D.TRN\_BRN = v\_Home\_Branch   
and D.EVENT\_CODE = 'INIT'  
and D.RECORD\_STATUS = 'O'   
and D.AUTH\_STATUS is null   
and D.CHECKER\_ID is null

1. Process authorize

Variable:

v\_lc\_SQL\_CMD  
 v\_lc\_Trn\_Ref\_No  
 v\_lc\_source\_code

v\_lc\_Cr\_Ac\_No  
 v\_lc\_Cr\_Ac\_Branch  
 v\_lc\_Cr\_Ac\_Ccy  
 v\_lc\_Cr\_Amount

v\_lc\_Db\_Ac\_No  
 v\_lc\_Db\_Ac\_Branch

v\_lc\_Db\_Ac\_Ccy  
 v\_lc\_DB\_Amount

* Get post info

//Tài khoản ghi nợ

* + Select from table DLTBD\_TRANSACTION\_POST p
    - Columns: P.AC\_NO,P.AC\_BRANCH,P.AC\_CCY,P.AMOUNT
    - Returning into: v\_lc\_Db\_Ac\_No,v\_lc\_Db\_Ac\_Branch, v\_lc\_Db\_Ac\_Ccy, v\_lc\_DB\_Amount

//Tài khoản ghi có

* + Select from table DLTBD\_TRANSACTION\_POST p
    - Columns: P.AC\_NO,P.AC\_BRANCH,P.AC\_CCY,P.AMOUNT
    - Returning into v\_lc\_Cr\_Ac\_No,v\_lc\_Cr\_Ac\_Branch, v\_lc\_Cr\_Ac\_Ccy, v\_lc\_Cr\_Amount
* Post to CoreBanking
  + if v\_lc\_Db\_Ac\_No like '1011%' //TM

v\_lc\_source\_code := 'THCK';

v\_lc\_SQL\_CMD:= 'select '|| '''' || v\_lc\_Cr\_Ac\_No || '''' || ','

|| '''' || v\_lc\_Cr\_Ac\_Branch || '''' || ','  
 || '''' || v\_lc\_Db\_Ac\_No || '''' || ','  
 || '''' || v\_lc\_Db\_Ac\_Branch || '''' || ',';

v\_lc\_SQL\_CMD:= v\_lc\_SQL\_CMD || '''' || v\_lc\_Db\_Ac\_Ccy || '''' || ','

|| v\_lc\_DB\_Amount || ','

|| '''' || v\_Trn\_Desc || '''' || ','

|| '''' || v\_lc\_source\_code || '''' || ','

|| '''' || c\_txn\_code || '''' || ','

|| '''' || upper(v\_Maker\_ID) || '''' || ','

|| '''' || c\_Common\_User || '''' || ' '

|| 'from dual '

|| 'where rownum = 1 ';

* + else //CK

v\_lc\_source\_code := 'CKCK';

v\_lc\_SQL\_CMD:= 'select '|| '''' || v\_lc\_Db\_Ac\_No || '''' || ','

|| '''' || v\_lc\_Db\_Ac\_Branch || '''' || ','

|| '''' || v\_lc\_Cr\_Ac\_No || '''' || ','

|| '''' || v\_lc\_Cr\_Ac\_Branch || '''' || ',';

v\_lc\_SQL\_CMD:= v\_lc\_SQL\_CMD || '''' || v\_lc\_Db\_Ac\_Ccy || '''' || ','

|| v\_lc\_DB\_Amount || ','

|| '''' || v\_Trn\_Desc || '''' || ','

|| '''' || v\_lc\_source\_code || '''' || ','

|| '''' || c\_txn\_code || '''' || ','

|| '''' || null || '''' || ','

|| '''' ||c\_Common\_User || '''' || ' '

|| 'from dual '

|| 'where rownum = 1 ';

* Gọi package ebank.PKG\_CORE\_SWITCH.PR\_RT\_TRANSFER\_V3(v\_lc\_SQL\_CMD, v\_Home\_Branch,v\_Value\_Date,0,v\_lc\_Trn\_Ref\_No, v\_Rcd\_Err.MESSAGE ,v\_MSGID,v\_Rcd\_Trn.XREF) post to CoreBanking
* If substr(v\_Rcd\_Err.MESSAGE,3,2) <> '00'
  + if (substr(v\_Rcd\_Err.MESSAGE,3,2) = '80' or substr(v\_Rcd\_Err.MESSAGE,3,2) = '90' or substr(v\_Rcd\_Err.MESSAGE,3,2) = '91')
    - Update table: DLTBD\_TRANSACTION\_POST p
      * Set: P.TRN\_STATUS = 'S',  
         P.POST\_DATE = sysdate,  
         P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)
    - Update table: DLTBD\_HOADON\_INFO b
      * Set: B.NGAYGIO = to\_date(p\_NgayGio, 'mm/dd/yyyy hh:mi:ss am')  
         B.Time\_Out = v\_Rcd\_Trn.XREF
    - v\_Rcd\_Err.ERROR\_CODE := '1'||substr(p\_Err\_String,3,2);
    - v\_Rcd\_Err.ERROR\_TYPE := pkg\_Dom\_Util.c\_gl\_Debug\_Msg;
  + else
    - v\_Rcd\_Err.ERROR\_CODE := '0099';
    - v\_Rcd\_Err.ERROR\_TYPE := pkg\_Dom\_Util.c\_gl\_Sys\_Err;
* else
  + update table: datbd\_transaction d
    - Set: d.AUTH\_STATUS = 'A',  
       d.CHECKER\_DT = sysdate,  
       d.CHECKER\_ID = p\_Checker\_ID  
       d.RECORD\_STATUS = 'O'  
       d.CORE\_REF\_NO = v\_lc\_Trn\_Ref\_No
  + Update table: DLTBD\_TRANSACTION\_POST p
    - Set: P.TRN\_STATUS = 'S'  
       P.POST\_DATE = sysdate  
       P.CORE\_REF\_NO = v\_lc\_Trn\_Ref\_No  
       P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)
  + Update table: DLTBD\_HOADON\_INFO b
    - Set: B.NGAYGIO = to\_date (p\_NgayGio , 'mm/dd/yyyy hh:mi:ss am')

**Step 19:** Gọi Service gạch nợ:

* Gọi LVBService.BankRequest()

**Step 24:** Update Database đã gạch nợ

* Update table: DLTBD\_HOADON\_INFO

Nếu EVN return gạch nợ thành công

* + Update Column: TRANGTHAIGD, TRANGTHAIHUYGD, NGAYGIO
  + Value: TRANGTHAIGD = 0, TRANGTHAIHUYGD = 9, NGAYGIO = Date.Now()

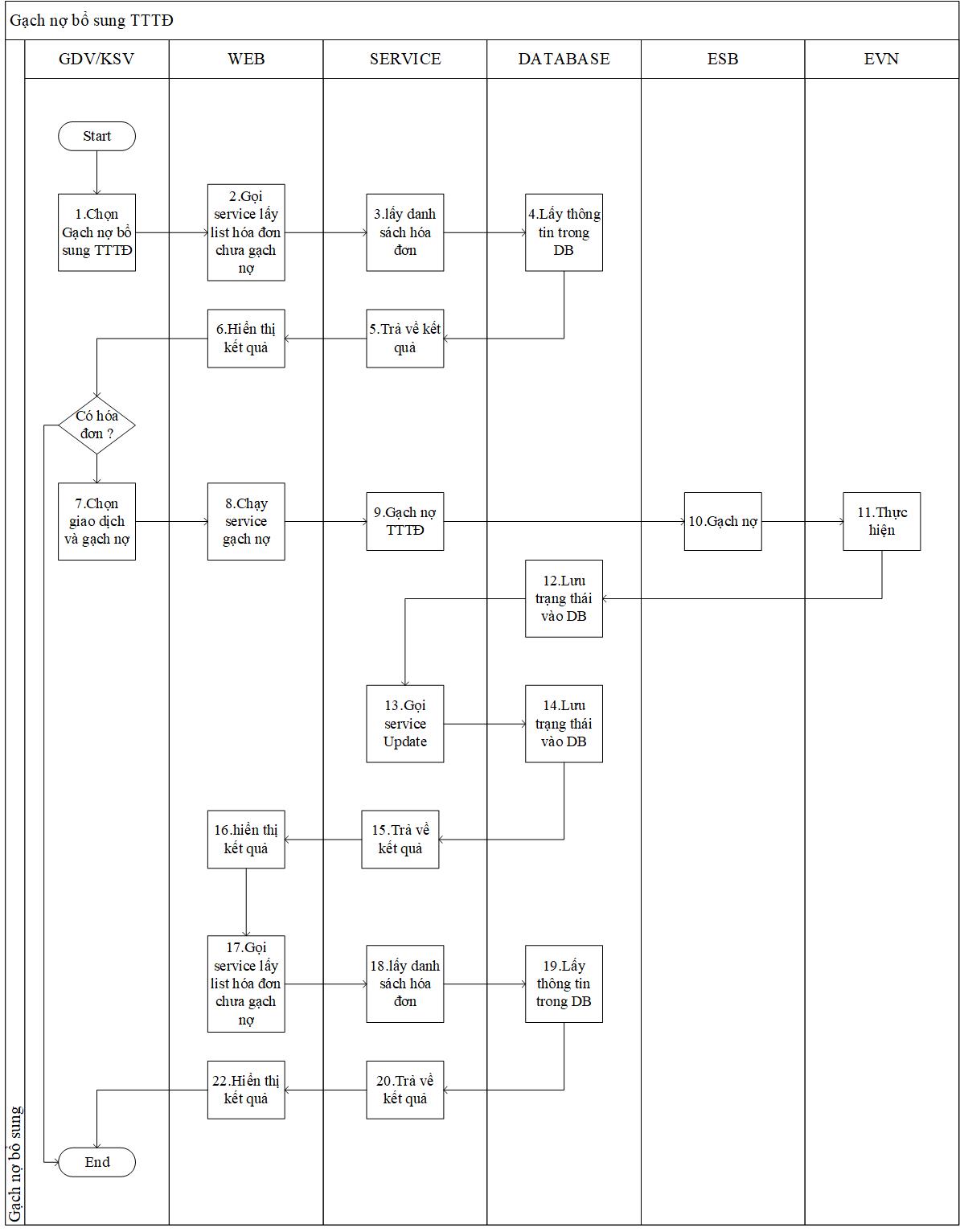
Nếu EVN return False

* + Update Column: TRANGTHAIGD, TRANGTHAIHUYGD
  + Value: TRANGTHAIGD = 1, TRANGTHAIHUYGD = 9

Nếu xảy ra Exception

* + Update Column: TRANGTHAIGD, TRANGTHAIHUYGD, NGAYGIO
  + Value: TRANGTHAIGD = 2, TRANGTHAIHUYGD = 9, NGAYGIO = Date.Now()

1. **Gạch nợ bổ sung TTTĐ**



Bước 2: Gọi service lấy list hóa đơn chưa gạch nợ :

EVN\_Auto\_Service.Service1.getHoaDonGachNoBoSung\_Ver2() 🡪 EVN\_AUTO. getHoaDonGachNoBoSung(String maker\_ID, string branchCode)

Bước 4 : Lấy thông tin trong DB:

SQL\_CMD = "select \* from DLTBD\_SCHEDULES where RECORD\_STAT='O'";

result = db.SelectCommand(SQL\_CMD, dt\_DATE)

if (result.Substring(0, 1) =="0"&& dt\_DATE.Rows.Count== 1)

sDATE = dt\_DATE.Rows[0]["NUMBER\_DAY"].ToString()

else return dt // “ “

SQL\_CMD :

select b.MAKH, b.MAHD, b.HOADONID, b.SOTIEN, b.NGAYGIO, b.TRANGTHAIGD, t.MAKER\_ID, t.TRN\_BRN, t.TRANSACTION\_ID, t.module,

decode (t.module,'DL','EVNHCM',t.module) module\_name

from ( select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD,TRN\_CHANEL, TRN\_ID,TIME\_OUT

from DLTBD\_HOADON\_INFO where not (NVL(HDDT, '2') = 1 )

and TRN\_CHANEL in ('TELLER', 'AUTO') and NGAYGIO >= SYSDATE - sDATE

union all

select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID, TIME\_OUT

from DLTBD\_MN\_HOADON\_INFO Where NGAYGIO >= SYSDATE sDATE

union all

select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID, TIME\_OUT

from DLTBD\_MT\_HOADON\_INFO Where NGAYGIO >= SYSDATE sDATE

union all

select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID, TIME\_OUT

from DLTBD\_MB\_HOADON\_INFO Where NGAYGIO >= SYSDATE sDATE

union all

select MAKH,MAHD, HOADONID, SOTIEN, NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, TRN\_CHANEL, TRN\_ID, TIME\_OUT \n"

from DLTBD\_HN\_HOADON\_INFO Where NGAYGIO >=SYSDATE-sDATE)b, DATBD\_TRANSACTION t

WHERE(NVL(B.TRANGTHAIGD, '1') <>'0')

AND B.TRANGTHAIHUYGD = '9'

AND B.Time\_Out is null

AND T.MODULE in ('DL','EVNHCM','EVNCPC','EVNSPC', 'EVNHNI','EVNNPC')

AND T.MAKER\_ID = maker\_ID

AND T.TRN\_BRN = branchCode

AND b.TRN\_ID = to\_char(t.TRANSACTION\_ID)

and t.EVENT\_CODE='INIT'

AND T.RECORD\_STATUS = 'O'

AND T.AUTH\_STATUS = 'A'

db.SelectCommand(SQL\_CMD, dt\_HoaDon)

return dt\_HoaDon

Bước 8: Gọi service gạch nợ bổ sung :

EVN\_auto\_Service.Service1.GachNoBoSungAll\_Ver2() 🡪 EVN\_Library.EVN\_AUTO.GachNoBoSungAll()🡪EVNHCM\_Library.GachNoBoSung(

🡪 LVBService.BankConfirm() //Gạch nợ

Bước 12: Lưu trạng thái vào DB :

TBL: DLTBD\_HOADON\_INFO

Thành công : TRANGTHAIGD =0;

TRANGTHAIHUYGD =9;

NGAYGIO = DateTime.Now

where TRN\_ID = Trn\_ID

Thất bại : TRANGTHAIGD =1;

TRANGTHAIHUYGD =9;

where TRN\_ID = Trn\_ID

Exception : TRANGTHAIGD =2;

TRANGTHAIHUYGD =9;

NGAYGIO = DateTime.Now

where TRN\_ID = Trn\_ID

Bước 13 : Gọi service Update: updateAutoLogVer2(string Trn\_ID, string status, string coreStatus, string pgStatus)

Bước 14: Lưu kết quả vào DB

Update DLTBD\_AUTOPAY\_DETAILS

set AUTO\_STATUS = status,

POSTCORE\_MSG = coreStatus, //if coreStatus!= null or empty

PAYBILLS\_MSG = pgStatus //if pgStatus !=null or empty

where TRANSACTION\_ID = Trn\_ID

Bước 17 : Như bước 2

## **6.Hủy đăng ký tự động**



1. Bước 1:

Nhập và chọn dữ liệu đầu vào :

Chọn nhà cung cấp, nhập mã KH, nhập CIF, nhập số tài khoản , chọn ngày thu tiền, nhập số điện thoại , nhập email, chọn ngày đăng ký

1. Bước 2:

Thực hiện gọi Service để tìm kiếm danh sách khách hàng đã đăng ký thu tự động

1. Bước 3:

Gọi Service evnall.searchAutoRegisterToClose(arr[2], txt\_makh\_s.Text.Trim(), txt\_cif\_s.Text.Trim(), txt\_account\_s.Text.Trim(),ddl\_tungaythutien\_s.SelectedItem.Value, ddl\_denngaythutien\_s.SelectedItem.Value, txt\_sodienthoai\_s.Text.Trim(),

txt\_email\_s.Text.Trim(), txt\_from\_reg\_dt.Value.Trim(), txt\_to\_reg\_dt.Value.Trim(), ddl\_electric.SelectedValue.ToString())→EVN\_Library. searchAutoRegisterToClose(p\_Maker\_ID, MaKH, Cif, Ac\_No, From\_Pay\_Dt, To\_Pay\_Dt, Phone, Email, From\_Reg\_Dt, To\_Reg\_Dt, NCC)

1. Bước 4:

Thực hiện select lấy danh sách khách hàng đã đăng ký tự động

SQL

select \* from (

select R.\*, 'Mở' RECORD\_STAT\_DESC, 'EVNSPC' module

from DLTBD\_MN\_AUTO\_REGISTER R where RECORD\_STAT = 'O' AND REG\_BRANCH = (SELECT NVL(U.ACTIVE\_BRANCH,U.HOME\_BRANCH) FROM ASTB\_USER U WHERE U.USERNAME='" + p\_Maker\_ID + "') "

UNION ALL

select R.\*, 'Mở' RECORD\_STAT\_DESC, 'EVNCPC' module

from DLTBD\_MT\_AUTO\_REGISTER R where RECORD\_STAT = 'O' AND REG\_BRANCH = (SELECT NVL(U.ACTIVE\_BRANCH,U.HOME\_BRANCH) FROM ASTB\_USER U WHERE U.USERNAME='" + p\_Maker\_ID + "')

UNION ALL

select R.\*, 'Mở' RECORD\_STAT\_DESC, 'EVNNPC' module

from DLTBD\_MB\_AUTO\_REGISTER R where RECORD\_STAT = 'O' AND REG\_BRANCH = (SELECT NVL(U.ACTIVE\_BRANCH,U.HOME\_BRANCH) FROM ASTB\_USER U WHERE U.USERNAME='" + p\_Maker\_ID + "') "

UNION ALL

select R.\*, 'Mở' RECORD\_STAT\_DESC, 'EVNHNI' module

from DLTBD\_HN\_AUTO\_REGISTER R where RECORD\_STAT = 'O' AND REG\_BRANCH = (SELECT NVL(U.ACTIVE\_BRANCH,U.HOME\_BRANCH) FROM ASTB\_USER U WHERE U.USERNAME='" + p\_Maker\_ID + "') "

UNION ALL

select R.\*, 'Mở' RECORD\_STAT\_DESC, 'EVNHCM' module

from DLTBD\_AUTO\_REGISTER R where RECORD\_STAT = 'O' AND REG\_BRANCH = (SELECT NVL(U.ACTIVE\_BRANCH,U.HOME\_BRANCH) FROM ASTB\_USER U WHERE U.USERNAME='" + p\_Maker\_ID + "') "

) where 1=1

Nếu NCC.Trim() != ""

AND MODULE = '" + NCC + "'

Nếu MaKH.Trim() != ""

AND MAKH like '" + MaKH + "%'

Nếu Cif.Trim() != ""

AND CIF ='" + Cif + "'

Nếu Ac\_No.Trim() != ""

AND AC\_NO ='" + Ac\_No + "'

Nếu Phone.Trim() != ""

AND PHONE ='" + Phone + "'

Nếu Email.Trim() != ""

AND EMAIL like '" + Email + "%'

Nếu From\_Pay\_Dt.Trim() != ""

AND to\_number(PAY\_DT) >= to\_number('" + From\_Pay\_Dt + "')

Nếu To\_Pay\_Dt.Trim() != ""

AND to\_number(PAY\_DT) <= to\_number('" + To\_Pay\_Dt + "')

Nếu From\_Reg\_Dt.Trim() != ""

AND trunc(REG\_DT) >= to\_date('" + From\_Reg\_Dt + "','dd/mm/yyyy')

1. Bước 5:

Trả lại danh sách khách hàng đã đăng ký thu tự động

DataTable dt = new DataTable();

string[] arr = (string[])Session["up\_evn"];

dt = evnall.searchAutoRegisterToClose(arr[2], txt\_makh\_s.Text.Trim(), txt\_cif\_s.Text.Trim(), txt\_account\_s.Text.Trim(),

ddl\_tungaythutien\_s.SelectedItem.Value, ddl\_denngaythutien\_s.SelectedItem.Value, txt\_sodienthoai\_s.Text.Trim(),

txt\_email\_s.Text.Trim(), txt\_from\_reg\_dt.Value.Trim(), txt\_to\_reg\_dt.Value.Trim(), ddl\_electric.SelectedValue.ToString());

1. Bước 6:

Hiển thị danh sách khác hàng đăng ký thu tự động

g\_kh.DataSource = dt;

g\_kh.DataBind();

1. Bước 6.1.1:

Khi click vào nút xem

1. Bước 6.1.2:

Sang trang u\_ViewAutoRegist.aspx?makh={0}&ncc={1}

1. Bước 6.1.3:

Gọi service để hiển thị chi tiết khách hàng đã đăng ký thu tự động

1. Bước 6.1.4:

Gọi Service evn.getAutoRegist4View("", makh, "", "", "", "", "", "", "", "", "", NCC) → EVN\_Library. getAutoRegist4View(p\_Maker\_ID, MaKH, Cif, Ac\_No, From\_Pay\_Dt, To\_Pay\_Dt, Phone,Email, Reg\_Branch, From\_Reg\_Dt, To\_Reg\_Dt, NCC)

1. Bước 6.1.5:

Select dữ liệu chi tiết khách hàng đã đăng ký thu tự động

select \* from (

SELECT A.\*, DECODE(A.CHECKER\_ID, null, 'Chưa duyệt', '', 'Chưa duyệt', 'Đã duyệt') AUTH\_STAT\_DESC, DECODE(A.RECORD\_STAT, 'O', 'Bình thường', 'I', DECODE(NVL(LAST\_AUDIT\_VER,0), 0, 'Tạo mới','Chỉnh sửa'), 'C', 'Đóng') RECORD\_STAT\_DESC,'EVNCPC' MODULE

FROM DLTBD\_MT\_AUTO\_REGISTER A "

union all

SELECT A.\*, DECODE(A.CHECKER\_ID, null, 'Chưa duyệt', '', 'Chưa duyệt', 'Đã duyệt') AUTH\_STAT\_DESC,DECODE(A.RECORD\_STAT, 'O', 'Bình thường', 'I', DECODE(NVL(LAST\_AUDIT\_VER,0), 0, 'Tạo mới','Chỉnh sửa'), 'C', 'Đóng') RECORD\_STAT\_DESC,'EVNNPC' MODULE

FROM DLTBD\_MB\_AUTO\_REGISTER A

union all

SELECT A.\*, DECODE(A.CHECKER\_ID, null, 'Chưa duyệt', '', 'Chưa duyệt', 'Đã duyệt') AUTH\_STAT\_DESC,DECODE(A.RECORD\_STAT, 'O', 'Bình thường', 'I', DECODE(NVL(LAST\_AUDIT\_VER,0), 0, 'Tạo mới','Chỉnh sửa'), 'C', 'Đóng') RECORD\_STAT\_DESC,'EVNHNI' MODULE

FROM DLTBD\_HN\_AUTO\_REGISTER A

union all

SELECT A.\*, DECODE(A.CHECKER\_ID, null, 'Chưa duyệt', '', 'Chưa duyệt', 'Đã duyệt') AUTH\_STAT\_DESC,DECODE(A.RECORD\_STAT, 'O', 'Bình thường', 'I', DECODE(NVL(LAST\_AUDIT\_VER,0), 0, 'Tạo mới','Chỉnh sửa'), 'C', 'Đóng') RECORD\_STAT\_DESC,'EVNHCM' MODULE

FROM DLTBD\_AUTO\_REGISTER A

union all

SELECT A.\*, DECODE(A.CHECKER\_ID, null, 'Chưa duyệt', '', 'Chưa duyệt', 'Đã duyệt') AUTH\_STAT\_DESC,DECODE(A.RECORD\_STAT, 'O', 'Bình thường', 'I', DECODE(NVL(LAST\_AUDIT\_VER,0), 0, 'Tạo mới','Chỉnh sửa'), 'C', 'Đóng') RECORD\_STAT\_DESC, 'EVNSPC' MODULE

FROM DLTBD\_MN\_AUTO\_REGISTER A

) WHERE 1=1

Nếu NCC.Trim() != ""

AND MODULE = '" + NCC + "'

Nếu MaKH.Trim() != ""

AND MAKH like '" + MaKH + "%'

Nếu Cif.Trim() != ""

AND CIF ='" + Cif + "'

Nếu Ac\_No.Trim() != ""

AND AC\_NO ='" + Ac\_No + "'

Nếu Phone.Trim() != ""

AND PHONE ='" + Phone + "'

Nếu Email.Trim() != ""

AND EMAIL like '" + Email + "%'

Nếu From\_Pay\_Dt.Trim() != ""

AND to\_number(PAY\_DT) >= to\_number('" + From\_Pay\_Dt + "')

Nếu To\_Pay\_Dt.Trim() != ""

AND to\_number(PAY\_DT) <= to\_number('" + To\_Pay\_Dt + "')

Nếu Reg\_Branch.Trim() != ""

AND REG\_BRANCH ='" + Reg\_Branch + "'

Nếu From\_Reg\_Dt.Trim() != ""

AND trunc(REG\_DT) >= to\_date('" + From\_Reg\_Dt + "','dd/mm/yyyy')

Nếu To\_Reg\_Dt.Trim() != ""

AND trunc(REG\_DT) <= to\_date('" + To\_Reg\_Dt + "','dd/mm/yyyy')

order by TO\_NUMBER(PAY\_DT)

1. Bước 6.1.6:

Trả lại thông tin chi tiết khách hàng đăng ký thu tự động

DataTable dt = new DataTable()

dt = evn.getAutoRegist4View("", makh, "", "", "", "", "", "", "", "", "", NCC)

1. Bước 6.1.7:

Hiển thị thông tin chi tiết khách hàng

lbMaKH.Text = dt.Rows[0]["MAKH"].ToString();

lbCif.Text = dt.Rows[0]["CIF"].ToString();

lbAcct.Text = dt.Rows[0]["AC\_NO"].ToString();

lbPayDate.Text = dt.Rows[0]["PAY\_DT"].ToString();

lbPhone.Text = dt.Rows[0]["PHONE"].ToString();

lbEmail.Text = dt.Rows[0]["EMAIL"].ToString();

lbInHD.Text = (dt.Rows[0]["PRINT\_BILL"].ToString() == "1") ? "Y" : "N";

lbl\_rec\_sms.Text = dt.Rows[0]["RECEIVE\_SMS"].ToString();

lbl\_rec\_email.Text = dt.Rows[0]["RECEIVE\_EMAIL"].ToString();

lbl\_reg\_branch.Text = dt.Rows[0]["REG\_BRANCH"].ToString();

if (!string.IsNullOrEmpty(dt.Rows[0]["REG\_DT"].ToString()))

lbl\_reg\_dt.Text = Util.MDYToDMY(dt.Rows[0]["REG\_DT"].ToString()); //tren live ngay thang bi nguoc, tren test ko bi

if (!string.IsNullOrEmpty(dt.Rows[0]["CLOSE\_DT"].ToString()))

lbl\_close\_dt.Text = Util.MDYToDMY(dt.Rows[0]["CLOSE\_DT"].ToString());

lbl\_close\_reason.Text = dt.Rows[0]["CLOSE\_REASON"].ToString();

lbl\_maker\_id.Text = dt.Rows[0]["MAKER\_ID"].ToString();

if (!string.IsNullOrEmpty(dt.Rows[0]["MAKER\_DT"].ToString()))

lbl\_maker\_dt.Text = Util.MDYToDMY(dt.Rows[0]["MAKER\_DT"].ToString());

lbl\_checker\_id.Text = dt.Rows[0]["CHECKER\_ID"].ToString();

if (!string.IsNullOrEmpty(dt.Rows[0]["CHECKER\_DT"].ToString()))

lbl\_checker\_dt.Text = Util.MDYToDMY(dt.Rows[0]["CHECKER\_DT"].ToString());

lbl\_record\_stat.Text = dt.Rows[0]["RECORD\_STAT\_DESC"].ToString();

lbl\_auth\_stat.Text = dt.Rows[0]["AUTH\_STAT\_DESC"].ToString();

1. Bước 6.2.1:

Khi click vào nút đóng

1. Bước 6.2.2:

Gọi Service để thực hiện hủy đăng ký thu tự động

1. Bước 6.2.3:

Gọi Service evnall.CloseAutoRegisterAll(arr[2], MaKH, ddl\_electric.SelectedValue.ToString() + MaDL) → EVN\_Library. CloseAutoRegisterAll(p\_Maker\_ID, p\_MaKH, NCC)

Nếu NCC.Substring(0, 6) == "EVNHCM"

if (hcm == null) hcm = new EVNHCM\_Library();

kq = hcm.CloseAutoRegister(p\_Maker\_ID, p\_MaKH);

1. Bước 6.2.4:

Thực hiện Select dữ liệu khách hàng đã đăng ký thu tự động

HCM:

Gọi đến pkg\_EVN\_HCM.DLP\_CLOSEAUTOREGISTER

Parameter:

p\_MAKER\_ID varchar2

p\_MAKH varchar2

p\_Err\_String out varchar2

p\_MaDL varchar2

Thực hiện select count(\*) into v\_Count

from DLTBD\_MB\_AUTO\_REGISTER

where MAKH = p\_MAKH AND MADL = p\_MaDL

and RECORD\_STAT = 'O';

Nếu v\_Count > 1: báo lỗi

Nếu v\_Count <1:

Nếu p\_Err\_String is null : thực hiện Update DLTBD\_MB\_AUTO\_REGISTER

Set MAKER\_ID = p\_MAKER\_ID,

MAKER\_DT = SYSDATE,

CHECKER\_ID = NULL,

CHECKER\_DT = NULL,

RECORD\_STAT = 'C',

CLOSE\_DT = trunc(SYSDATE),

LAST\_AUDIT\_VER = NVL(LAST\_AUDIT\_VER, 0) + 1

Where MAKH = p\_MAKH AND MADL = p\_MaDL

and RECORD\_STAT = 'O';

Thực hiện select \* into v\_Rcd\_Customer

from DLTBD\_MB\_AUTO\_REGISTER

where MAKH = p\_MAKH AND MADL = p\_MaDL

and RECORD\_STAT = 'C' ;

Thực hiện insert :

insert into DLTBD\_MB\_AUTO\_REGISTER\_HIST(MAKH,MADL, CIF, AC\_NO, PAY\_DT, RECEIVE\_SMS, RECEIVE\_EMAIL, PHONE, EMAIL, REG\_BRANCH, REG\_DT, CLOSE\_DT, CLOSE\_REASON, RECORD\_STAT, MAKER\_ID, MAKER\_DT, CHECKER\_ID, CHECKER\_DT, LAST\_PAY\_DT, LAST\_ENTRY\_SEQ\_NO, AUDIT\_VER)

values(v\_Rcd\_Customer.MAKH,v\_Rcd\_Customer.MADL, v\_Rcd\_Customer.CIF, v\_Rcd\_Customer.AC\_NO, v\_Rcd\_Customer.PAY\_DT, v\_Rcd\_Customer.RECEIVE\_SMS, v\_Rcd\_Customer.RECEIVE\_EMAIL, v\_Rcd\_Customer.PHONE, v\_Rcd\_Customer.EMAIL, v\_Rcd\_Customer.REG\_BRANCH, v\_Rcd\_Customer.REG\_DT, v\_Rcd\_Customer.CLOSE\_DT, v\_Rcd\_Customer.CLOSE\_REASON, v\_Rcd\_Customer.RECORD\_STAT, v\_Rcd\_Customer.MAKER\_ID, SYSDATE, NULL, NULL, v\_Rcd\_Customer.LAST\_PAY\_DT, v\_Rcd\_Customer.LAST\_ENTRY\_SEQ\_NO, v\_Rcd\_Customer.LAST\_AUDIT\_VER);

1. Bước 6.2.5:

Trả lại kết quả

String result = evnall.CloseAutoRegisterAll(arr[2], MaKH, ddl\_electric.SelectedValue.ToString() + MaDL);

1. Bước 6.2.6:

Thông báo kết quả

if (result.Substring(0, 1) == "1")

{

JSHelper.MsgBox(result);

//dr = ((DataTable)Session["dl\_err\_code"]).Select("ERROR\_CODE='" + result.Substring(0, 1) + result.Substring(2, 4) + "'");

//foreach (DataRow row in dr)

//{

// JSHelper.MsgBox(row["CUS\_MESSAGE"].ToString());

//}

}

else

{

JSHelper.MsgBox("Đóng thành công.");

}

1. Bước 6.2.7:

Gọi lại service evnall.searchAutoRegisterToClose(arr[2], txt\_makh\_s.Text.Trim(), txt\_cif\_s.Text.Trim(), txt\_account\_s.Text.Trim(),

ddl\_tungaythutien\_s.SelectedItem.Value, ddl\_denngaythutien\_s.SelectedItem.Value, txt\_sodienthoai\_s.Text.Trim(),

txt\_email\_s.Text.Trim(), txt\_from\_reg\_dt.Value.Trim(), txt\_to\_reg\_dt.Value.Trim(), ddl\_electric.SelectedValue.ToString())

g\_kh.DataSource = dt;

g\_kh.DataBind();

để hiển thị danh sách khách hàng thu tự động , làm tương tự từ bước 2 đến bước 6

## **Kích hoạt lại đăng ký thu tự động**



**Step 2:** Gọi service tìm kiếm khách hàng

Gọi Service1. searchAutoRegisterToOpen(String p\_Maker\_ID, String MaKH, String Cif, String Ac\_No, String From\_Pay\_Dt, String To\_Pay\_Dt, String Phone, String Email, String From\_Reg\_Dt, String To\_Reg\_Dt, String NCC)

**Step 4:** Select list khách hàng

SQL\_CMD =

select \* from   
(

select R.\*, 'Đóng' RECORD\_STAT\_DESC, 'EVNSPC' module  
from DLTBD\_MN\_AUTO\_REGISTER R where RECORD\_STAT = 'C' AND CHECKER\_ID IS NOT NULL AND CHECKER\_DT IS NOT NULL  
UNION ALL  
select R.\*, 'Đóng' RECORD\_STAT\_DESC, 'EVNNPC' module  
from DLTBD\_MB\_AUTO\_REGISTER R where RECORD\_STAT = 'C' AND CHECKER\_ID IS NOT NULL AND CHECKER\_DT IS NOT NULL  
UNION ALL  
select R.\*, 'Đóng' RECORD\_STAT\_DESC, 'EVNNPC' module  
from DLTBD\_AUTO\_REGISTER R where RECORD\_STAT = 'C' AND CHECKER\_ID IS NOT NULL AND CHECKER\_DT IS NOT NULL  
UNION ALL  
select R.\*, 'Đóng' RECORD\_STAT\_DESC, 'EVNNPC' module  
from DLTBD\_HN\_AUTO\_REGISTER R where RECORD\_STAT = 'C' AND CHECKER\_ID IS NOT NULL AND CHECKER\_DT IS NOT NULL  
UNION ALL  
select R.\*, 'Đóng' RECORD\_STAT\_DESC, 'EVNNPC' module  
from DLTBD\_MT\_AUTO\_REGISTER R where RECORD\_STAT = 'C' AND CHECKER\_ID IS NOT NULL AND CHECKER\_DT IS NOT NULL

) where 1=1

* if (NCC.Trim() != "")

SQL\_CMD = SQL\_CMD + "AND MODULE = '" + NCC + "'

* if (MaKH.Trim() != "")

SQL\_CMD = SQL\_CMD + "AND MAKH like '" + MaKH + "%' ";

* if (Cif.Trim() != "")

SQL\_CMD = SQL\_CMD + "AND CIF ='" + Cif + "' ";

* if (Ac\_No.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND AC\_NO ='" + Ac\_No + "' ";
* if (Phone.Trim() != "")

SQL\_CMD = SQL\_CMD + "AND PHONE ='" + Phone + "' ";

* if (Email.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND EMAIL like '" + Email + "%' ";
* if (From\_Pay\_Dt.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND to\_number(PAY\_DT) >= to\_number('" + From\_Pay\_Dt + "') ";
* if (To\_Pay\_Dt.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND to\_number(PAY\_DT) <= to\_number('" + To\_Pay\_Dt + "') ";
* if (From\_Reg\_Dt.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND trunc(REG\_DT) >= to\_date('" + From\_Reg\_Dt + "','dd/mm/yyyy') ";
* if (To\_Reg\_Dt.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND trunc(REG\_DT) <= to\_date('" + To\_Reg\_Dt + "','dd/mm/yyyy') ";

SQL\_CMD = SQL\_CMD + " order by TO\_NUMBER(PAY\_DT)";

**Step 8:** Gọi Service lấy chi tiết đăng ký

Gọi Service1. getAutoRegist4View(p\_Maker\_ID, MaKH, Cif, Ac\_No, From\_Pay\_Dt, To\_Pay\_Dt, Phone,Email, Reg\_Branch, From\_Reg\_Dt, To\_Reg\_Dt, NCC))

**Step 10:** Select chi tiết đăng ký

SQL\_CMD =

select \* from   
(

SELECT A.\*, DECODE(A.CHECKER\_ID, null, 'Chưa duyệt', '', 'Chưa duyệt', 'Đã duyệt') AUTH\_STAT\_DESC, DECODE(A.RECORD\_STAT, 'O', 'Bình thường', 'I', DECODE(NVL(LAST\_AUDIT\_VER,0), 0, 'Tạo mới','Chỉnh sửa'), 'C', 'Đóng') RECORD\_STAT\_DESC, 'EVNCPC' MODULE  
FROM DLTBD\_MT\_AUTO\_REGISTER A  
union all  
SELECT A.\*, DECODE(A.CHECKER\_ID, null, 'Chưa duyệt', '', 'Chưa duyệt', 'Đã duyệt') AUTH\_STAT\_DESC, DECODE(A.RECORD\_STAT, 'O', 'Bình thường', 'I', DECODE(NVL(LAST\_AUDIT\_VER,0), 0, 'Tạo mới','Chỉnh sửa'), 'C', 'Đóng') RECORD\_STAT\_DESC, 'EVNNPC' MODULE  
FROM DLTBD\_MB\_AUTO\_REGISTER A

union all  
SELECT A.\*, DECODE(A.CHECKER\_ID, null, 'Chưa duyệt', '', 'Chưa duyệt', 'Đã duyệt') AUTH\_STAT\_DESC, DECODE(A.RECORD\_STAT, 'O', 'Bình thường', 'I', DECODE(NVL(LAST\_AUDIT\_VER,0), 0, 'Tạo mới','Chỉnh sửa'), 'C', 'Đóng') RECORD\_STAT\_DESC, 'EVNNPC' MODULE  
FROM DLTBD\_HN\_AUTO\_REGISTER A

union all  
SELECT A.\*, DECODE(A.CHECKER\_ID, null, 'Chưa duyệt', '', 'Chưa duyệt', 'Đã duyệt') AUTH\_STAT\_DESC, DECODE(A.RECORD\_STAT, 'O', 'Bình thường', 'I', DECODE(NVL(LAST\_AUDIT\_VER,0), 0, 'Tạo mới','Chỉnh sửa'), 'C', 'Đóng') RECORD\_STAT\_DESC, 'EVNNPC' MODULE  
FROM DLTBD\_AUTO\_REGISTER A

union all  
SELECT A.\*, DECODE(A.CHECKER\_ID, null, 'Chưa duyệt', '', 'Chưa duyệt', 'Đã duyệt') AUTH\_STAT\_DESC, DECODE(A.RECORD\_STAT, 'O', 'Bình thường', 'I', DECODE(NVL(LAST\_AUDIT\_VER,0), 0, 'Tạo mới','Chỉnh sửa'), 'C', 'Đóng') RECORD\_STAT\_DESC, 'EVNNPC' MODULE  
FROM DLTBD\_MN\_AUTO\_REGISTER A

) WHERE 1=1

* if (NCC.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND MODULE = '" + NCC + "' "
* if (MaKH.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND MAKH like '" + MaKH + "%' "
* if (Cif.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND CIF ='" + Cif + "' ";
* if (Ac\_No.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND AC\_NO ='" + Ac\_No + "' ";
* if (Phone.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND PHONE ='" + Phone + "' ";
* if (Email.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND EMAIL like '" + Email + "%' ";
* if (From\_Pay\_Dt.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND to\_number(PAY\_DT) >= to\_number('" + From\_Pay\_Dt + "') ";
* if (To\_Pay\_Dt.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND to\_number(PAY\_DT) <= to\_number('" + To\_Pay\_Dt + "') ";
* if (Reg\_Branch.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND REG\_BRANCH ='" + Reg\_Branch + "' ";
* if (From\_Reg\_Dt.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND trunc(REG\_DT) >= to\_date('" + From\_Reg\_Dt + "','dd/mm/yyyy') ";
* if (To\_Reg\_Dt.Trim() != "")  
  SQL\_CMD = SQL\_CMD + "AND trunc(REG\_DT) <= to\_date('" + To\_Reg\_Dt + "','dd/mm/yyyy') ";

SQL\_CMD = SQL\_CMD + " order by TO\_NUMBER(PAY\_DT)";

**Step 14:** Gọi Service mở đăng ký thu tự động

Gọi Service1. OpenAutoRegisterAll(p\_Maker\_ID, p\_MaKH, NCC)

**Step 16:** Mở đăng ký thu tự động

pkg\_EVN\_HCM.DLP\_OPENAUTOREGISTER

Parameters:  
 p\_MAKER\_ID varchar2  
 p\_MAKH varchar2  
 p\_Err\_String out varchar2

Variables:  
 v\_Home\_Branch varchar2(5);  
 v\_Count number;   
 v\_Rcd\_Customer DLTBD\_AUTO\_REGISTER%ROWTYPE;

1. Kiểm tra khách hàng hủy đăng ký thu tự động

select count(\*) into v\_Count  
from DLTBD\_AUTO\_REGISTER  
where MAKH = p\_MAKH  
 and RECORD\_STAT = 'C'  
 and CHECKER\_ID IS NOT NULL  
 and CHECKER\_DT IS NOT NULL;

if v\_Count <> 1 then  
 p\_Err\_String := '1#0014#No customer found (DLP\_OPENAUTOREGISTER)';

1. Process

if p\_Err\_String is null then

update DLTBD\_AUTO\_REGISTER  
 set MAKER\_ID = p\_MAKER\_ID,  
 MAKER\_DT = SYSDATE,  
 CHECKER\_ID = NULL,  
 CHECKER\_DT = NULL,  
 RECORD\_STAT = 'I',  
 REG\_BRANCH = v\_Home\_Branch,  
 REG\_DT = trunc(sysdate),  
 CLOSE\_DT = NULL,  
 LAST\_AUDIT\_VER = NVL(LAST\_AUDIT\_VER, 0) +1  
 where MAKH = p\_MAKH  
 and RECORD\_STAT = 'C'  
 and CHECKER\_ID IS NOT NULL  
 and CHECKER\_DT IS NOT NULL;

p\_Err\_String := '0#0000#Mo thanh cong (DLP\_OPENAUTOREGISTER)';

1. Update history

select \* into v\_Rcd\_Customer  
from DLTBD\_AUTO\_REGISTER  
where MAKH = p\_MAKH  
and (RECORD\_STAT='I'   
AND CHECKER\_ID IS NULL   
AND MAKER\_ID = p\_MAKER\_ID);

exception  
 when others then  
 p\_Err\_String := '1#0099#pkg\_EVN: Read DLTBD\_AUTO\_REGISTER\_HIST ' || sqlerrm;

insert into DLTBD\_AUTO\_REGISTER\_HIST(MAKH, CIF, AC\_NO, PAY\_DT, RECEIVE\_SMS, RECEIVE\_EMAIL, PHONE, EMAIL, REG\_BRANCH, REG\_DT, CLOSE\_DT, CLOSE\_REASON, RECORD\_STAT, MAKER\_ID, MAKER\_DT, CHECKER\_ID, CHECKER\_DT, LAST\_PAY\_DT, LAST\_ENTRY\_SEQ\_NO, AUDIT\_VER)  
values(v\_Rcd\_Customer.MAKH, v\_Rcd\_Customer.CIF, v\_Rcd\_Customer.AC\_NO, v\_Rcd\_Customer.PAY\_DT, v\_Rcd\_Customer.RECEIVE\_SMS, v\_Rcd\_Customer.RECEIVE\_EMAIL, v\_Rcd\_Customer.PHONE, v\_Rcd\_Customer.EMAIL, v\_Rcd\_Customer.REG\_BRANCH, v\_Rcd\_Customer.REG\_DT, v\_Rcd\_Customer.CLOSE\_DT, v\_Rcd\_Customer.CLOSE\_REASON, v\_Rcd\_Customer.RECORD\_STAT, v\_Rcd\_Customer.MAKER\_ID, SYSDATE, NULL, NULL, v\_Rcd\_Customer.LAST\_PAY\_DT, v\_Rcd\_Customer.LAST\_ENTRY\_SEQ\_NO,v\_Rcd\_Customer.LAST\_AUDIT\_VER);

**Step 19:** Giống step 2

**Step 21:** Giống step 4

# **Upload**

1. **Upload file/Truy vấn/Tạo batch**



**Step 5:** Gọi service truy vấn hóa đơn khách hàng

* Gọi Service1.QueryBills(dt\_customer,Product\_ID)

→ service.BankRequest(MaKH, DIEM\_THU, OWNER\_BANK\_ID)

* Return:
  + dt[khachhang]
    - Makh, tenkh, diachikh, madl, masothuekh
  + dt[hoadon]
    - mahd, sotien, mota, tungay, denngay, giabieu, hoadonid, soho, magiaodich, DNTT, tiendien, tienthue, thuesuat, kyhieuhd, seryhd, HDDT, NCC

**Step 14:** Gọi Service Upload file

* Gọi Service1.UploadFile(MakerID, Upload\_ProductID, FileName)

**Step 16:** Upload file

* Gọi package pkg\_EVN\_Upload\_Payment.Fn\_Upload\_File  
  return: l\_Save\_Name\_Result

Parameter:

p\_Maker\_ID  
 p\_Product\_ID  
 p\_FileName  
 p\_Batch\_No  
Variable:

v\_Home\_Branch  
 v\_Rcd\_Err  
 v\_Rcd\_Upload\_Mater

1. Validate data input

v\_lc\_count number;

* Select count(\*) into v\_lc\_count
  + From DLTB\_BILL\_UPLOAD\_MASTER u
  + Where u.UPLOAD\_FILENAME = p\_FileName  
     and u.RECORD\_STAT = 'O';
* If v\_lc\_count > 0
  + v\_Rcd\_Err.ERROR\_CODE := '9001';
  + v\_Rcd\_Err.MESSAGE := 'Tên file đã tồn tại trong hệ thống. FileName is ' || p\_FileName || '.';
  + goto Final\_Process;
* select count(\*) into v\_lc\_count
  + From DLTM\_UPLOAD\_PRODUCT p
  + Where PRODUCT\_ID = p\_Product\_ID;
* If v\_lc\_count = 0
  + v\_Rcd\_Err.ERROR\_CODE := '9901';
  + v\_Rcd\_Err.MESSAGE := 'Sản phẩm không tồn tại. PRODUCT\_ID is ' || p\_Product\_ID || '.';
  + goto Final\_Process;

1. Get Settle Account

* select Home\_Branch into v\_Home\_Branch from ASTB\_USER u where u.USERNAME = p\_Maker\_ID and u.RECORD\_STATUS <> 'C' and rownum = 1;
* Select from DLTM\_UPLOAD\_SETTLE\_ACCOUNT p
  + Columns: AC\_NO, AC\_CCY, AC\_BRANCH, AC\_OWNER, decode(AC\_OWNER, null, 'G','A')
  + Into v\_Rcd\_Upload\_Mater.SETTLE\_AC\_NO, v\_Rcd\_Upload\_Mater.SETTLE\_AC\_CCY, v\_Rcd\_Upload\_Mater.SETTLE\_AC\_BRN, v\_Rcd\_Upload\_Mater.SETTLE\_AC\_OWNER, v\_Rcd\_Upload\_Mater.SETTLE\_AC\_CUST\_GL
  + Where: BRANCH\_CODE = v\_Home\_Branch;
* Exception
  + when no\_data\_found
    - Insert into DLTM\_UPLOAD\_SETTLE\_ACCOUNT
      * Columns: BRANCH\_CODE, AC\_NO, AC\_CCY , AC\_BRANCH, AC\_OWNER
      * Values: v\_Home\_Branch,'459900018','VND',v\_Home\_Branch, ''
  + when Too\_Many\_Rows
    - v\_Rcd\_Err.ERROR\_CODE := '9902'
    - v\_Rcd\_Err.MESSAGE := 'Có nhiều hơn 1 tài khoản ghi nợ cho mỗi chi nhánh có Branch\_Code is ' || v\_Home\_Branch || '.';
    - goto Final\_Process;

1. Process

* v\_Rcd\_Upload\_Mater.BATCH\_NO := generate\_Batch\_No;  
  v\_Rcd\_Upload\_Mater.UPLOAD\_DATE := TRUNC(SYSDATE);  
  v\_Rcd\_Upload\_Mater.BRANCH\_CODE := v\_Home\_Branch;  
  v\_Rcd\_Upload\_Mater.PRODUCT\_ID := p\_Product\_ID;  
  v\_Rcd\_Upload\_Mater.UPLOAD\_FILENAME := p\_FileName;  
  v\_Rcd\_Upload\_Mater.AUTH\_STAT := 'U';  
  v\_Rcd\_Upload\_Mater.RECORD\_STAT := 'O';  
  v\_Rcd\_Upload\_Mater.MAKER\_ID := p\_Maker\_ID;  
  v\_Rcd\_Upload\_Mater.MAKER\_DT := SYSDATE;  
  v\_Rcd\_Upload\_Mater.LAST\_BILL\_ESR := 0;
* Insert into DLTB\_BILL\_UPLOAD\_MASTER
  + Values: v\_Rcd\_Upload\_Mater
  + Returning BATCH\_NO into p\_Batch\_No;
  + v\_Rcd\_Err.ERROR\_CODE := '0000';
  + v\_Rcd\_Err.MESSAGE := 'Upload File thành công. Product\_ID, Branch\_Code is ' || p\_Product\_ID || ', ' || v\_Home\_Branch || '.';
* Exception:
  + v\_Rcd\_Err.ERROR\_CODE := '9999';
  + v\_Rcd\_Err.MESSAGE := 'Lỗi hệ thống: ' || sqlerrm|| chr(10) || 'Error At: ' || DBMS\_UTILITY.format\_error\_backtrace;
* Return: v\_Rcd\_Err

**Step 20 :** Gọi Service Add Bills

* Gọi Service1. AddBills(MakerID, BatchNo, dt\_bill)

**Step 22:** Add bills

Gọi pkg\_EVN\_Upload\_Payment.Fn\_Add\_Bills

Parameter:

p\_Maker\_ID  
p\_Batch\_No  
p\_Bill\_List

Variable:

v\_Home\_Branch  
 v\_Rcd\_Err  
 v\_Rcd\_Upload\_Mater

* Select M.\* into v\_Rcd\_Upload\_Mater
  + From DLTB\_BILL\_UPLOAD\_MASTER M
  + Where: BATCH\_NO = p\_Batch\_No  
     and MAKER\_ID = p\_Maker\_ID  
     and RECORD\_STAT = 'O'  
     and AUTH\_STAT = 'U'

v\_add\_Bills number(6) :=0;

* For r in xmltable(bill\_list)
  + v\_add\_Bills := v\_add\_Bills + 1;
  + if R.TOTAL\_BILL\_AMOUNT = 0 or R.NBR\_OF\_BILL = 0
    - p\_Rcd\_Err.ERROR\_CODE := '9003';
    - p\_Rcd\_Err.MESSAGE := 'Thêm hóa đơn bị lỗi. Do số tiền bằng 0 hoặc không có hoá đơn. (TOTAL\_BILL\_AMOUNT = 0 or NBR\_OF\_BILL = 0). BATCH\_NO, Bill\_ESR is ' || p\_Rcd\_Upload\_Mater.BATCH\_NO || ', ' || v\_add\_Bills || '.';
    - exit;
  + Update table DLTB\_BILL\_UPLOAD\_DETAILS d
    - Set: d.SOTIEN = R.SOTIEN  
       d.MADL = R.MADL  
       D.PAYMENT\_INFO\_NEW = R.PAYMENT\_INFO.getStringVal()  
       d.TOTAL\_BILL\_AMOUNT = R.TOTAL\_BILL\_AMOUNT  
       d.NBR\_OF\_BILL = R.NBR\_OF\_BILL;
    - Where d.BATCH\_NO = p\_Rcd\_Upload\_Mater.BATCH\_NO  
       d.MAKH = R.MAKH  
       d.RECORD\_STAT = 'O'
  + if sql%rowcount = 0
    - insert into DLTB\_BILL\_UPLOAD\_DETAILS
      * Columns: BATCH\_NO, BILL\_ENTRY\_SR, RECORD\_STAT, PROCESS\_STAT, MAKH, SOTIEN, MADL, PAYMENT\_INFO\_NEW, TOTAL\_BILL\_AMOUNT, NBR\_OF\_BILL, SETTLE\_AMOUNT
      * Values: p\_Rcd\_Upload\_Mater.BATCH\_NO, p\_Rcd\_Upload\_Mater.LAST\_BILL\_ESR + 1, 'O', 'I', R.MAKH, R.SOTIEN, R.MADL, R.PAYMENT\_INFO.getStringVal(), R.TOTAL\_BILL\_AMOUNT, R.NBR\_OF\_BILL, 0
      * Returning BILL\_ENTRY\_SR into   
         p\_Rcd\_Upload\_Mater.LAST\_BILL\_ESR;
* if v\_add\_Bills = 0
  + p\_Rcd\_Err.ERROR\_CODE := '9004';
  + p\_Rcd\_Err.MESSAGE := 'Không có hoá đơn nào được thêm vào. BATCH\_NO is ' || p\_Rcd\_Upload\_Mater.BATCH\_NO || '.';
* if v\_add\_Bills > 0
  + Update
    - Select LAST\_BILL\_ESR
      * From DLTB\_BILL\_UPLOAD\_MASTER m
      * Where BATCH\_NO = p\_Batch\_No  
         and MAKER\_ID = p\_Maker\_ID   
         and RECORD\_STAT = 'O'  
         and AUTH\_STAT = 'U'
    - Set LAST\_BILL\_ESR = v\_Rcd\_Upload\_Mater.LAST\_BILL\_ESR;
  + v\_Rcd\_Err.ERROR\_CODE := '0000';
  + v\_Rcd\_Err.MESSAGE := 'Thêm hoá đơn thành công. Batch\_No, Branch\_Code is ' || p\_Batch\_No || ', ' || v\_Home\_Branch || '.';
* Exception
  + when no\_data\_found
    - v\_Rcd\_Err.ERROR\_CODE := '9002';
    - v\_Rcd\_Err.MESSAGE := 'Hoá đơn không sẵn sàng để thêm vào. BATCH\_NO, Branch\_Code is ' || p\_Batch\_No || ', ' || v\_Home\_Branch || '.';
  + when others
    - rollback;
    - v\_Rcd\_Err.ERROR\_CODE := '9999';
    - v\_Rcd\_Err.MESSAGE := 'System Error: ' || sqlerrm|| chr(10) || 'Error At: ' || DBMS\_UTILITY.format\_error\_backtrace;

1. **Duyệt danh sách tiền điện**



**Step 2:** Gọi Service lấy list batch

* Gọi Service1. getBatch\_ByChecker(CheckerID, AuthStatus, Begin\_Dt, End\_Dt)

**Step 4:** Select list batch

select m.\*, to\_char(m.upload\_date, 'dd/MM/yyyy') supload\_date from DLTB\_BILL\_UPLOAD\_MASTER m,( select u.USERNAME, NVL(u.ACTIVE\_BRANCH, u.HOME\_BRANCH) ACTIVE\_BRANCH from ASTB\_USER u where u.USERNAME = :p\_UserName and u.RECORD\_STATUS = 'A') U where m.BRANCH\_CODE = u.ACTIVE\_BRANCH and m.RECORD\_STAT in ('O', 'P') and m.AUTH\_STAT = :p\_Auth\_Stat and trunc(m.upload\_date)>=to\_Date(:p\_Begin\_Dt, 'dd/MM/yyyy') and trunc(m.upload\_date)<=to\_Date(:p\_End\_Dt, 'dd/MM/yyyy')

**Step 10:** Gọi service lấy chi tiết batch

* Gọi Service1.getBatchEntries(BatchNo)

**Step 12:** Select batch entries

select d.\*, extractvalue(xmltype(nvl(d.payment\_info,'<PAYMENT\_INFO></PAYMENT\_INFO>')), '/PAYMENT\_INFO/BILLS/BILL[1]/OTHER\_INFO/MaGiaoDich') as MaGiaoDich from DLTB\_BILL\_UPLOAD\_DETAILS D where D.BATCH\_NO = :p\_Batch\_No and d.RECORD\_STAT = 'O'

**Step 17:** Gọi Service duyệt batch

* Gọi Service1. AuthBatch(CheckerID, BatchNo, Is\_SuspendPay)

**Step 19:** Duyệt batch

pkg\_EVN\_Upload\_Payment.Fn\_Authorize\_Batch  
Parameters:

p\_Checker\_ID  
 p\_Batch\_No

Variables:  
 v\_Home\_Branch  
 v\_Rcd\_Err  
 v\_Create\_Transaction  
 v\_Rcd\_Upload\_Mater

* Select M.\* into v\_Rcd\_Upload\_Mater
  + From DLTB\_BILL\_UPLOAD\_MASTER m
  + Where M.BATCH\_NO = p\_Batch\_No  
     and BRANCH\_CODE = v\_Home\_Branch  
     and M.RECORD\_STAT = 'O'  
     and M.AUTH\_STAT = 'U'
* v\_Rcd\_Upload\_Mater.CHECKER\_ID := p\_Checker\_ID;
* case v\_Rcd\_Upload\_Mater.PRODUCT\_ID when 'HCM\_UPLOAD'
  + v\_Create\_Transaction := pkg\_EVN\_HCM.Fn\_Import\_Transaction\_Upload(v\_Rcd\_Upload\_Mater, v\_Rcd\_Err);

pkg\_EVN\_HCM.Fn\_Import\_Transaction\_Upload(v\_Rcd\_Upload\_Mater, v\_Rcd\_Err)

Parameters:

p\_Rcd\_Batch\_Master DLTB\_BILL\_UPLOAD\_MASTER%ROWTYPE  
 p\_Rcd\_Err

Variable:

v\_Import\_Transaction\_Upload := false;

* p\_Rcd\_Batch\_Master.SETTLE\_TRN\_ID := pkg\_Global\_Dom\_Application.get\_Dom\_Transaction\_ID();
* for r in  
  ( select d.PAYMENT\_INFO\_NEW

from DLTB\_BILL\_UPLOAD\_DETAILS d  
where d.BATCH\_NO = p\_Rcd\_Batch\_Master.BATCH\_NO

and D.RECORD\_STAT = 'O'

)

Loop

v\_lc\_Customer\_Info xmltype;  
 v\_lc\_Bills\_Info xmltype;

* + select extract(PAYMENT\_INFO\_NEW, '/PAYMENT\_INFO/RECEIVE\_ACCOUNT'), extract(PAYMENT\_INFO\_NEW, '/PAYMENT\_INFO/BILLS')  
    into v\_lc\_Customer\_Info, v\_lc\_Bills\_Info  
    from (

select xmltype(r.PAYMENT\_INFO\_NEW) PAYMENT\_INFO\_NEW

from dual

);

* + for c in xmltable( '/RECEIVE\_ACCOUNT'  
     passing v\_lc\_Customer\_Info
    - Update DLTB\_CUSTOMER\_INFO
      * Set TENKH = C.TENKH,  
         DIACHIKH = C.DIACHIKH,  
         MADL = C.MADL,  
         MASOTHUEKH = C.MASOTHUE,  
         PHIEN = C.PHIEN,  
         LOTRINH = C.LOTRINH,  
         SOGHICSCMIS = C.SOGHICSCMIS,  
         DANHSO = C.DANHSO,  
         SOCONGTO = C.SOCONGTO,  
         NGANHNGHE = C.NGANHNGHE
      * where MAKH = C.MAKH
    - If sql%rowcount = 0
      * Insert into DLTB\_CUSTOMER\_INFO
      * Columns: MAKH, TENKH, DIACHIKH, MADL, MASOTHUEKH, PHIEN, LOTRINH, SOGHICSCMIS, DANHSO, SOCONGTO, NGANHNGHE
      * Values: trim(C.MAKH), C.TENKH, C.DIACHIKH, C.MADL, C.MASOTHUE, C.PHIEN, C.LOTRINH, C.SOGHICSCMIS, C.DANHSO, C.SOCONGTO, C.NGANHNGHE
  + Insert into DLTBD\_HOADON\_INFO
    - Columns: MAHD,MAKH,SOTIEN,MOTA,HOADONID,SOHO, BANKID, DAINHD, TRN\_CHANEL, TRN\_ID, MAGIAODICH, HOADON\_ID, TUNGAY, DENNGAY, GIABIEU, DNTT, TIENDIEN, TIENTHUE, THUESUAT, KYHIEUHD, SERYHD, HDDT
    - select b.MAHD, b.MAKH, b.SOTIEN, b.MOTA, b.HOADONID, b.SOHO, c\_Bank\_ID, decode(b.MAGIAODICH, null, '0', '1'), decode(b.MAGIAODICH, null, c\_Chanel\_AUTO, c\_Chanel\_TELLER) , p\_Rcd\_Batch\_Master.SETTLE\_TRN\_ID, nvl(b.MaGiaoDich, lpad(EBANK.SEQ\_PG\_030002.NEXTVAL,10, '0')), EBANK.SEQ\_PG\_030002.currval, b.TUNGAY, b.DENNGAY, b.GIABIEU, b.DNTT, b.TIENDIEN, b.TIENTHUE, b.THUESUAT, b.KYHIEUHD, b.SERYHD, b.HDDT
    - from xmltable( '/BILLS/BILL'  
       passing v\_lc\_Bills\_Info  
       columns
* insert into DATBD\_TRANSACTION
  + Columns: TRANSACTION\_ID, TRN\_BRN, VALUE\_DT, TRN\_CODE, MODULE, EVENT\_CODE, RECORD\_STATUS,MAKER\_ID, MAKER\_DT, CHECKER\_ID, APP\_TYPE, LAST\_EVENT\_SEQ\_NO
  + Values: p\_Rcd\_Batch\_Master.SETTLE\_TRN\_ID, p\_Rcd\_Batch\_Master.BRANCH\_CODE, trunc(sysdate), 'LIQUID', pkg\_EVN\_Upload\_Payment.Module\_Name, 'INIT', 'O', p\_Rcd\_Batch\_Master.MAKER\_ID, p\_Rcd\_Batch\_Master.MAKER\_DT, p\_Rcd\_Batch\_Master.CHECKER\_ID, 'DOM\_WEB', 1
* v\_Import\_Transaction\_Upload := true;
* if v\_Create\_Transaction

//v\_Create\_Transaction = v\_Import\_Transaction\_Upload

* + v\_Rcd\_Upload\_Mater.AUTH\_STAT := 'A';
  + v\_Rcd\_Upload\_Mater.CHECKER\_DT := sysdate;
  + v\_Rcd\_Upload\_Mater.SETTLE\_CHANEL := c\_Chanel\_TELLER;
  + v\_Rcd\_Upload\_Mater.CHECKER\_DT := sysdate;
  + update  
    (

select M.\*  
from DLTB\_BILL\_UPLOAD\_MASTER m  
where M.BATCH\_NO = p\_Batch\_No  
 and BRANCH\_CODE = v\_Home\_Branch  
 and M.RECORD\_STAT = 'O'  
 and M.AUTH\_STAT = 'U'

)

Set ROW = v\_Rcd\_Upload\_Mater;

* + v\_Rcd\_Err.ERROR\_CODE := '0000';
  + v\_Rcd\_Err.MESSAGE := 'Duyệt Batch thành công. Batch\_No, Branch\_Code is ' || p\_Batch\_No || ', ' || v\_Home\_Branch || '.';
* Exception
  + when no\_data\_found then
    - v\_Rcd\_Err.ERROR\_CODE := '9002'
    - v\_Rcd\_Err.MESSAGE := 'Batch không sẵn sàng để duyệt. BATCH\_NO, Branch\_Code is ' || p\_Batch\_No || ', ' || v\_Home\_Branch || '.';
  + when others then
    - v\_Rcd\_Err.ERROR\_CODE := '9999'
    - v\_Rcd\_Err.MESSAGE := 'System Error: ' || sqlerrm|| chr(10) || 'Error At: ' || DBMS\_UTILITY.format\_error\_backtrace;

1. **Kiểm tra giao dịch TimeOut**



1. Bước 1:

Chọn ngày hoặc nhập ngày cần check

1. Bước 2:

Check dữ liệu đầu vào ngày nhập

1. Bước 3:

Gọi Service để lấy danh sách các giao dịch nghi ngờ

1. Bước 4:

Gọi Service g\_Evn\_Payment.getBatch\_CheckTimeOut(l\_Staff\_ID, l\_Date) → Upload\_Payment.getBatch\_CheckTimeOut(checker,sdate)

1. Bước 5:

Sql:

select to\_char(rownum) STT, P.AMOUNT TRANS\_Amount, P.AC\_NO Account\_NO, t.XREF MSG\_ID, t.TRN\_DESC TRANS\_DES,u.BATCH\_NO BATCH\_NO,u.PRODUCT\_ID NCC,t.TRANSACTION\_ID TRANS\_ID,p.CORE\_TRN\_DT CORE\_DT, '' SRESULT

FROM

DLTBD\_TRANSACTION\_POST p,

DATBD\_TRANSACTION t,

DLTB\_BILL\_UPLOAD\_MASTER u

WHERE 1= 1

AND U.RECORD\_STAT = 'W'

AND P.TRN\_STATUS = 'W'

AND P.DRCR\_IND = 'D'

AND P.TRANSACTION\_ID = T.TRANSACTION\_ID

AND P.TRANSACTION\_ID = U.SETTLE\_TRN\_ID

And T.CHECKER\_ID = '" + checker\_id + "'

And to\_Char(T.MAKER\_DT,'dd/mm/yyyy') = '" + sdate + "'

1. Bước 6:

Return:

* Dt[TIMEOUT]:

STT, AMOUNT TRANS\_Amount, AC\_NO Account\_NO, XREF MSG\_ID, TRANS\_DES, BATCH\_NO, NCC, TRANS\_ID, CORE\_DT, SRESULT

1. Bước 7:

Truyền dữ liệu lên gridView và truyền vào ViewState

g\_Khach\_hang.DataSource = l\_DataTable;

g\_Khach\_hang.DataBind();

ViewState["CHECKTIMEOUT"] = l\_DataTable;

1. Bước 8:

Chọn Giao dịch nghi ngờ cần kiểm tra

1. Bước 9:

Gọi Service để thực hiện kiểm tra giao dịch

1. Bước 10:

Gọi Service g\_Evn\_Payment.checkTimeOut\_transaction(checker\_id, batch\_no, trans\_id, msg\_id) → Upload\_Payment. checkTimeOut\_transaction(checker\_id, batch\_no, trans\_id, msg\_id)

Gọi đến package pkg\_EVN\_Upload\_Payment.Fn\_App\_Settle\_CheckTimeOut

Parameter:

p\_Checker\_ID varchar2

p\_Batch\_No varchar2

trans\_id varchar2

msg\_id varchar2

Constant:

c\_Core\_Prod\_Code varchar2(5):= 'UPEBK';

c\_Core\_Txn\_Code varchar2(3):= 'EBK';

Variable:

v\_Home\_Branch varchar2(5);

v\_Rcd\_Err PKG\_DOM\_UTIL.RCD\_ERROR;

v\_Trn\_Brn DATBD\_TRANSACTION.TRN\_BRN%TYPE;

v\_Trn\_Desc DATBD\_TRANSACTION.TRN\_DESC%TYPE;

v\_Value\_Dt DATBD\_TRANSACTION.VALUE\_DT%TYPE;

v\_Maker\_ID DATBD\_TRANSACTION.MAKER\_ID%TYPE;

v\_Last\_Event\_Seq\_No DATBD\_TRANSACTION.LAST\_EVENT\_SEQ\_NO%TYPE;

v\_Upload\_Status boolean:= false;

v\_Core\_Check\_Status varchar2(1000) :='' ;

v\_Core\_Ref\_no varchar2(1000) :='' ;

v\_Core\_Err\_String varchar2(1000) :='' ;

v\_rowcount int :=0 ;

v\_rowcount\_trans int :=0 ;

v\_rowcount\_post int :=0 ;

Thực hiện Update

(

select M.\*

from DLTB\_BILL\_UPLOAD\_MASTER m

where M.BATCH\_NO = p\_Batch\_No

and BRANCH\_CODE = v\_Home\_Branch

and M.RECORD\_STAT = 'W'

and M.AUTH\_STAT = 'A'

and m.SETTLE\_DT is not null

and SETTLE\_TRN\_ID = trans\_id

)

set RECORD\_STAT = 'K'; *--old O*

v\_rowcount := sql%rowcount;

Nếu v\_rowcount khác 1 : Thông báo lỗi

Thực hiện insert biến v\_rowcount\_trans = select count(\*) from DATBD\_TRANSACTION t

where TRANSACTION\_ID = trans\_id

and t.RECORD\_STATUS='O'

AND t.XREF = msg\_id

Nếu v\_rowcount\_trans khác 1 : Thông báo lỗi

Thực hiện insert biến v\_rowcount\_post = select count(\*)from DLTBD\_TRANSACTION\_POST P

where TRANSACTION\_ID = trans\_id

and P.TRN\_STATUS='W'

and P.CORE\_REF\_NO is null;

Nếu v\_rowcount\_post khác 1 : Thông báo lỗi

1. Bước 11:

Thực hiện gọi sang Core hạch toán:

EBANK.PKG\_CORE\_SWITCH.CHECKTRANSACTION(msg\_id,v\_Core\_Check\_Status,v\_Core\_Ref\_no,v\_Core\_Err\_String);

1. Bước 12:

Nếu v\_Core\_Ref\_no có giá trị:

set biến v\_Upload\_Status := true

v\_Rcd\_Err.ERROR\_CODE := '0000'

v\_Rcd\_Err.ERROR\_TYPE := PKG\_DOM\_UTIL.C\_GL\_DEBUG\_MSG

v\_Rcd\_Err.MESSAGE := 'Check giao dich timeout thanh cong. Batch\_No, Branch\_Code, core\_ref l?: ' || p\_Batch\_No || ', ' || v\_Home\_Branch || ',' || v\_Core\_Ref\_no|| '.'

Update bảng DLTBD\_TRANSACTION\_POST p

Set P.CORE\_REF\_NO = v\_Core\_Ref\_no

P.CORE\_TRN\_DT = (select max(today) from sttm\_dates where branch\_code = p.AC\_BRANCH)

P.TRN\_STATUS = 'S'

Với điều kiện:

where P.TRANSACTION\_ID = trans\_id

and p.TRN\_STATUS = 'W'

Nếu v\_Core\_Ref\_no không có giá trị:

Kiểm tra biến v\_Core\_Err\_String substr(nvl(v\_Core\_Err\_String,'1#99'),3,2)

Nếu bằng ‘00’: v\_Upload\_Status := true;

v\_Rcd\_Err.ERROR\_CODE := '0000';

v\_Rcd\_Err.ERROR\_TYPE := PKG\_DOM\_UTIL.C\_GL\_DEBUG\_MSG;

v\_Rcd\_Err.MESSAGE := 'Check timeout transaction thanh cong. Batch\_No, Branch\_Code, core\_ref l?: ' || p\_Batch\_No || ', ' || v\_Home\_Branch || '.';

Nếu bằng ‘82’: v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0006';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Cust\_Data\_Err;

v\_Rcd\_Err.Message := 'T?i kho?n kh?ng d? ti?n thanh to?n.';

Nếu bằng ‘83’: v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0006';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Cust\_Data\_Err;

v\_Rcd\_Err.Message := 'T?i kho?n kh?ng d? ti?n thanh to?n.';

Nếu bằng ‘84’: v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0004';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'CCY c?a giao d?ch v? s?n ph?m kh?ng gi?ng nhau.';

Nếu bằng ‘80’: v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0080';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'Core exception.';

Nếu bằng ‘90’: v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0090';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'Core timeout.';

Nếu bằng ‘91’: v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0091';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'Exception when process ESB response.';

Nếu bằng ‘98’: v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0098';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'Core failed.';

Nếu bằng ‘99’: v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0099';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'Exception befor calling ESB.';

Giá trị khác: v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code :='9990';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'C?p nh?t v?o core kh?ng th?nh c?ng - ';

Update bảng DLTBD\_TRANSACTION\_POST

Set P.TRN\_STATUS = 'F'

P.POST\_DATE = sysdate

P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)

Với điều kiện

where P.TRANSACTION\_ID = trans\_id

and p.TRN\_STATUS = 'W'

Update

(

select M.\*

from DLTB\_BILL\_UPLOAD\_MASTER m

where M.BATCH\_NO = p\_Batch\_No

and BRANCH\_CODE = v\_Home\_Branch

and M.RECORD\_STAT = 'K'

)

Set

RECORD\_STAT = 'O';

Thực hiện Insert IBTSD\_MAP\_TRANS\_MSG

(ID, TRANS\_ID, MSG\_ID, VALUE\_DT, CHECKER\_ID, MODULE\_ID)

Values

(IBTSD\_MAP\_TRANS\_MSG\_SEQ.nextval, trans\_id, msg\_id, sysdate, p\_Checker\_ID, 'EVN\_UPLOAD')

1. Bước 13:

Trả lại kết quả

1. Bước 14:

Hiển thị danh sách kết quả sau khi kiểm tra

1. **EVN Upload Maker**



1. Bước 1:

Nhập giá trị từ ngày và đến ngày

1. Bước 2:

Thực hiện gọi service để hiện thị các file được upload

1. Bước 3:

Nếu Giá trị từ ngày và đến ngày null thì gọi service:

g\_Evn\_Payment.getBatch\_ByMaker\_ALL(l\_Staff\_ID);

sql:

select m.\*, to\_char(m.upload\_date, 'dd/MM/yyyy') supload\_date

from DLTB\_BILL\_UPLOAD\_MASTER m,(

select u.USERNAME, NVL(u.ACTIVE\_BRANCH, u.HOME\_BRANCH) ACTIVE\_BRANCH

from ASTB\_USER where u.USERNAME = :p\_UserName and u.RECORD\_STATUS = 'A'

) U

where m.MAKER\_ID = u.USERNAME and m.BRANCH\_CODE = u.ACTIVE\_BRANCH

and m.RECORD\_STAT in ('O', 'P')and m.AUTH\_STAT = 'U'

XML:

<SP\_Parameter>

<Parameter>

<Name>p\_UserName</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>;

Nếu Giá trị từ ngày và đến ngày không null thì gọi service:

g\_Evn\_Payment.getBatch\_ByMaker(l\_Staff\_ID, l\_Begin\_Date, l\_End\_Date);

sql:

select m.\*, to\_char(m.upload\_date, 'dd/MM/yyyy') supload\_date

from DLTB\_BILL\_UPLOAD\_MASTER m,(

select u.USERNAME, NVL(u.ACTIVE\_BRANCH, u.HOME\_BRANCH) ACTIVE\_BRANCH

from ASTB\_USER u where u.USERNAME = :p\_UserName and u.RECORD\_STATUS = 'A'

) U

where m.MAKER\_ID = u.USERNAME and m.BRANCH\_CODE = u.ACTIVE\_BRANCH

and m.RECORD\_STAT in ('O', 'P')and m.AUTH\_STAT = 'U'

and trunc(m.upload\_date)>=to\_Date(:p\_Begin\_Dt, 'dd/MM/yyyy')

and trunc(m.upload\_date)<=to\_Date(:p\_End\_Dt, 'dd/MM/yyyy')

XML:

<SP\_Parameter>

<Parameter>

<Name>p\_UserName</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Begin\_Dt</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_End\_Dt</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

1. Bước 4:

getBatch\_ByMaker\_ALL(l\_Staff\_ID);

sql:

select m.\*, to\_char(m.upload\_date, 'dd/MM/yyyy') supload\_date

from DLTB\_BILL\_UPLOAD\_MASTER m,(

select u.USERNAME, NVL(u.ACTIVE\_BRANCH, u.HOME\_BRANCH) ACTIVE\_BRANCH

from ASTB\_USER where u.USERNAME = :p\_UserName and u.RECORD\_STATUS = 'A'

) U

where m.MAKER\_ID = u.USERNAME and m.BRANCH\_CODE = u.ACTIVE\_BRANCH

and m.RECORD\_STAT in ('O', 'P')and m.AUTH\_STAT = 'U'

XML:

<SP\_Parameter>

<Parameter>

<Name>p\_UserName</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>;

getBatch\_ByMaker(l\_Staff\_ID, l\_Begin\_Date, l\_End\_Date);

sql:

select m.\*, to\_char(m.upload\_date, 'dd/MM/yyyy') supload\_date

from DLTB\_BILL\_UPLOAD\_MASTER m,(

select u.USERNAME, NVL(u.ACTIVE\_BRANCH, u.HOME\_BRANCH) ACTIVE\_BRANCH

from ASTB\_USER u where u.USERNAME = :p\_UserName and u.RECORD\_STATUS = 'A'

) U

where m.MAKER\_ID = u.USERNAME and m.BRANCH\_CODE = u.ACTIVE\_BRANCH

and m.RECORD\_STAT in ('O', 'P')and m.AUTH\_STAT = 'U'

and trunc(m.upload\_date)>=to\_Date(:p\_Begin\_Dt, 'dd/MM/yyyy')

and trunc(m.upload\_date)<=to\_Date(:p\_End\_Dt, 'dd/MM/yyyy')

XML:

<SP\_Parameter>

<Parameter>

<Name>p\_UserName</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Begin\_Dt</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_End\_Dt</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

1. Bước 5:

Trả lại danh sách các file đã được Upload

DataTable l\_DataTable = g\_Evn\_Payment.getBatch\_ByMaker(l\_Staff\_ID, l\_Begin\_Date, l\_End\_Date);

DataTable l\_DataTable = g\_Evn\_Payment.getBatch\_ByMaker\_ALL(l\_Staff\_ID);

1. Bước 6:

Hiển thị danh sách các file đã được Upload lên grid view

c\_GridView\_Master.DataSource = l\_DataTable;

1. Bước 7:

Chọn file trong danh sách file đã được upload

1. Bước 8:

Hiển thị thông tin file đã được chọn

string l\_Selected\_ID = e.CommandArgument.ToString();

DataTable l\_dataTable = (DataTable)Session[Lv\_Session.DT\_Upload\_Master];

l\_dataTable.CaseSensitive = true;

DataRow[] l\_dataRow = l\_dataTable.Select(Lv\_Define.BATCH\_NO + "='" + l\_Selected\_ID + "'");

DataTable l\_dataTable\_Selected =

CommonFuntion.Create\_Seclected\_Table(l\_dataRow[0], Lv\_Session.DT\_Upload\_Master\_Selected);

Session[Lv\_Session.DT\_Upload\_Master\_Selected] = l\_dataTable\_Selected;

foreach (DataRow dataRow in l\_dataRow)

{

string l\_Batch\_No = dataRow[Lv\_Define.BATCH\_NO].ToString();

string l\_Upload\_Date = dataRow[Lv\_Define.UPLOAD\_DATE].ToString();

string l\_Upload\_FileName = dataRow[Lv\_Define.UPLOAD\_FILENAME].ToString();

string l\_Maker\_ID = dataRow[Lv\_Define.MAKER\_ID].ToString();

lbl\_Selected\_ID.Text = l\_Batch\_No;

lbl\_File\_Name.Text = l\_Upload\_FileName;

Close\_All\_Panel();

c\_Panel\_Info.Visible = true;

}

1. Bước 9:

Khi click vào nút xem chi tiết

1. Bước 10:

Gọi service để lấy thông tin chi tiết của hóa đơn

1. Bước 11:

Gọi Service g\_Evn\_Payment.getBatchEntries(l\_Batch\_No) →Upload\_Payment.getBatchEntries(BatchNo)

1. Bước 12:

SQL

select d.\*, extractvalue(xmltype(nvl(d.payment\_info,'<PAYMENT\_INFO></PAYMENT\_INFO>')), '/PAYMENT\_INFO/BILLS/BILL[1]/OTHER\_INFO/MaGiaoDich') as MaGiaoDich //add MaGiaoDich cho HCM

from DLTB\_BILL\_UPLOAD\_DETAILS D

where D.BATCH\_NO = :p\_Batch\_No and d.RECORD\_STAT = 'O'

XML

<SP\_Parameter>

<Parameter>

<Name>p\_Batch\_No</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

1. Bước 13:

DataTable l\_DT\_Batch\_Detail = new DataTable(Lv\_Session.DT\_Upload\_Detail);

l\_DT\_Batch\_Detail = g\_Evn\_Payment.getBatchEntries(l\_Batch\_No);

Session[Lv\_Session.DT\_Upload\_Detail] = l\_DT\_Batch\_Detail;

DataTable l\_dataTable = (DataTable)(Session[Lv\_Session.DT\_Upload\_Master\_Selected]);

if (l\_dataTable.Rows.Count>0)

{

string l\_Batch\_No = l\_dataTable.Rows[0][Lv\_Define.BATCH\_NO].ToString();

Session[Lv\_Define.BATCH\_NO] = l\_Batch\_No;

Close\_All\_Panel();

c\_Panel\_Info.Visible = true;

c\_Panel\_Detail.Visible = true;

Load\_Detail\_List(l\_Batch\_No);

DataTable l\_DT\_Fee\_Detail = (DataTable)Session[Lv\_Session.DT\_Upload\_Detail];

CommonFuntion.GridView\_Show(c\_GridView\_Detail, l\_DT\_Fee\_Detail, Lv\_Define.BATCH\_NO, 0);

}

1. Bước 14:

Hiển thị thông tin chi tiết file

1. Bước 15:

Khi ấn nút hủy danh sách

1. Bước 16:

Hiển thị nút xác nhận hủy

DataTable l\_dataTable = (DataTable)(Session[Lv\_Session.DT\_Upload\_Master\_Selected]);

Lv\_Result\_Status l\_Result\_Status = new Lv\_Result\_Status();

if (l\_dataTable.Rows.Count > 0)

{

Close\_All\_Panel();

c\_Panel\_Info.Visible = true;

c\_Panel\_Cancel\_Confirm.Visible = true;

}

else

{

lbl\_Error.ForeColor = Definition\_Color.Status;

lbl\_Error.Text = "Không Có Dữ Liệu";

}

1. Bước 17:

Khi ấn nút xác nhận hủy

1. Bước 18:

Gọi service để thực hiện xác nhận hủy

1. Bước 19:

string l\_Batch\_No = l\_dataTable.Rows[0][Lv\_Define.BATCH\_NO].ToString();

Session[Lv\_Define.BATCH\_NO] = l\_Batch\_No;

string l\_Result = g\_Evn\_Payment.CancelBatch(l\_Staff\_ID, l\_Batch\_No) →Upload\_Payment.CancelBatch(MakerID, BatchNo)

1. Bước 20:

Thực hiện gọi store

pkg\_EVN\_Upload\_Payment.Fn\_Cancel\_Batch

Parameter

<SP\_Parameter>

<Parameter>

<Name></Name>

<Type>VARCHAR2</Type>

<Direction>Returnvalue</Direction>

</Parameter>

<Parameter>

<Name>p\_Maker\_ID</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Batch\_No</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

Thực hiện Update bảng

(

select RECORD\_STAT

from DLTB\_BILL\_UPLOAD\_MASTER

where BATCH\_NO = p\_Batch\_No

and MAKER\_ID = p\_Maker\_ID

and RECORD\_STAT = 'O'

and AUTH\_STAT = 'U'

)

set RECORD\_STAT = 'C';

Nếu không update được bản ghi nào : thông báo lỗi

Nếu updae được : thông báo thành công

1. Bước 21:

Trả lại kết quả

1. Bước 22:

Thông báo lên màn hình

1. Bước 23:

Khi ấn nút Back

1. Bước 24:

Gọi hàm Reload\_Branch\_Master()

1. Bước 25:

Thực hiện gọi service để hiện thị các file được upload

Nếu Giá trị từ ngày và đến ngày null thì gọi service:

g\_Evn\_Payment.getBatch\_ByMaker\_ALL(l\_Staff\_ID);

sql:

select m.\*, to\_char(m.upload\_date, 'dd/MM/yyyy') supload\_date

from DLTB\_BILL\_UPLOAD\_MASTER m,(

select u.USERNAME, NVL(u.ACTIVE\_BRANCH, u.HOME\_BRANCH) ACTIVE\_BRANCH

from ASTB\_USER where u.USERNAME = :p\_UserName and u.RECORD\_STATUS = 'A'

) U

where m.MAKER\_ID = u.USERNAME and m.BRANCH\_CODE = u.ACTIVE\_BRANCH

and m.RECORD\_STAT in ('O', 'P')and m.AUTH\_STAT = 'U'

XML:

<SP\_Parameter>

<Parameter>

<Name>p\_UserName</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>;

Nếu Giá trị từ ngày và đến ngày không null thì gọi service:

g\_Evn\_Payment.getBatch\_ByMaker(l\_Staff\_ID, l\_Begin\_Date, l\_End\_Date);

sql:

select m.\*, to\_char(m.upload\_date, 'dd/MM/yyyy') supload\_date

from DLTB\_BILL\_UPLOAD\_MASTER m,(

select u.USERNAME, NVL(u.ACTIVE\_BRANCH, u.HOME\_BRANCH) ACTIVE\_BRANCH

from ASTB\_USER u where u.USERNAME = :p\_UserName and u.RECORD\_STATUS = 'A'

) U

where m.MAKER\_ID = u.USERNAME and m.BRANCH\_CODE = u.ACTIVE\_BRANCH

and m.RECORD\_STAT in ('O', 'P')and m.AUTH\_STAT = 'U'

and trunc(m.upload\_date)>=to\_Date(:p\_Begin\_Dt, 'dd/MM/yyyy')

and trunc(m.upload\_date)<=to\_Date(:p\_End\_Dt, 'dd/MM/yyyy')

XML:

<SP\_Parameter>

<Parameter>

<Name>p\_UserName</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Begin\_Dt</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_End\_Dt</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

1. Bước 27:

Trả lại danh sách các file đã được Upload

DataTable l\_DataTable = g\_Evn\_Payment.getBatch\_ByMaker(l\_Staff\_ID, l\_Begin\_Date, l\_End\_Date);

DataTable l\_DataTable = g\_Evn\_Payment.getBatch\_ByMaker\_ALL(l\_Staff\_ID);

1. Bước 28:

Hiển thị danh sách các file được upload

## **EVN Upload View**



1. Bước 1:

Nhập giá trị từ ngày và đến ngày

1. Bước 2:

Thực hiện gọi service

1. Bước 3:

Thực hiện gọi Service l\_DataTable = g\_Evn\_Payment.getBatch\_ByBranch(l\_Staff\_ID, "-1", "-1", "-1", txt\_Begin\_Dt.Value, txt\_End\_Dt.Value) → Upload\_Payment.getBatch\_ByBranch(UserID, AuthStatus, RecordStatus, Settled, l\_Begin\_Dt, l\_End\_Dt);

1. Bước 4:

Thực hiện select dữ liệu

SQL

select

m.\*,

to\_char(m.upload\_date, 'dd/MM/yyyy') supload\_date ,

DECODE(m.AUTH\_STAT, 'A', DECODE(m.SETTLE\_DT, NULL, 'Đã duyệt file', 'Đã hạch toán'),

'U',

'Chưa duyệt file',

'Không xác định') FILE\_STAT

from DLTB\_BILL\_UPLOAD\_MASTER m,

(select u.USERNAME, NVL(u.ACTIVE\_BRANCH, u.HOME\_BRANCH) ACTIVE\_BRANCH

from ASTB\_USER u where u.USERNAME = :p\_UserName and u.RECORD\_STATUS = 'A') U

where m.BRANCH\_CODE = u.ACTIVE\_BRANCH"

Nếu biến l\_Begin\_Dt.Length > 6

Thêm điều kiện tại where: and trunc(m.upload\_date)>=to\_Date('"+l\_Begin\_Dt+"', 'dd/MM/yyyy')

Nếu biến l\_End\_Dt.Length > 6

Thêm điều kiện tại where: and trunc(m.upload\_date)<=to\_Date('"+l\_End\_Dt+"', 'dd/MM/yyyy')

and m.RECORD\_STAT in ('O', 'P','W','K')

and m.AUTH\_STAT = decode(:p\_Auth\_Stat, 'A', 'A', 'U', 'U', m.AUTH\_STAT)

and m.RECORD\_STAT = decode(:p\_Record\_Stat, 'O', 'O', 'P', 'P',m.RECORD\_STAT)

and :p\_Settled = decode(:p\_Settled, '-1', :p\_Settled,'0', decode(m.SETTLE\_DT, null, '0', '-1'),'1', decode(m.SETTLE\_DT, null, '-1', '1'), '2', decode(m.SETTLE\_DT, null, decode(m.RECORD\_STAT, 'O', '2', '-1'), decode(m.RECORD\_STAT, 'P', '2', '-1')))

Parameter:

<SP\_Parameter>

<Parameter>

<Name>p\_UserName</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Auth\_Stat</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Record\_Stat</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Settled</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

UserID, AuthStatus, RecordStatus, Settled

1. Bước 5:

Trả lại danh sách các file đã được upload vào hệ thống

DataTable dt\_batch = new DataTable("BATCH\_BRANCH");

db.SelectCommand(SQL\_CMD, C\_Parameter.DB\_XML\_PARAMETERS\_GET\_BATCH\_BY\_BRANCH, new Object[] { UserID, AuthStatus, RecordStatus, Settled }, dt\_batch)

1. Bước 6:

Hiển thị danh sách file lên gridview

DataTable l\_DataTable = new DataTable();

l\_DataTable = g\_Evn\_Payment.getBatch\_ByBranch(l\_Staff\_ID, "-1", "-1", "-1", txt\_Begin\_Dt.Value, txt\_End\_Dt.Value);

c\_GridView\_Master.DataSource = l\_DataTable;

c\_GridView\_Master.DataBind();

1. Bước 7:

Chọn file trong danh sách file hiển thị lên gridView

1. Bước 8:

Hiển thị thông tin File đã được chọn

string l\_Selected\_ID = e.CommandArgument.ToString();

DataTable l\_dataTable = (DataTable)Session[Lv\_Session.DT\_Upload\_Master];

l\_dataTable.CaseSensitive = true;

DataRow[] l\_dataRow = l\_dataTable.Select(Lv\_Define.BATCH\_NO + "='" + l\_Selected\_ID + "'");

DataTable l\_dataTable\_Selected =

CommonFuntion.Create\_Seclected\_Table(l\_dataRow[0], Lv\_Session.DT\_Upload\_Master\_Selected);

Session[Lv\_Session.DT\_Upload\_Master\_Selected] = l\_dataTable\_Selected;

foreach (DataRow dataRow in l\_dataRow)

{

string l\_Batch\_No = dataRow[Lv\_Define.BATCH\_NO].ToString();

string l\_Upload\_Date = dataRow[Lv\_Define.UPLOAD\_DATE].ToString();

string l\_Upload\_FileName = dataRow[Lv\_Define.UPLOAD\_FILENAME].ToString();

string l\_Maker\_ID = dataRow[Lv\_Define.MAKER\_ID].ToString();

string l\_Auth\_Status = dataRow[Lv\_Define.AUTH\_STAT].ToString();

string l\_Settle\_Trn\_ID = dataRow[Lv\_Define.SETTLE\_TRN\_ID].ToString();

string l\_Settle\_Ac\_No = dataRow[Lv\_Define.SETTLE\_AC\_NO].ToString();

string l\_Settle\_Ac\_Ccy = dataRow[Lv\_Define.SETTLE\_AC\_CCY].ToString();

string l\_Settle\_Ac\_Brn = dataRow[Lv\_Define.SETTLE\_AC\_BRN].ToString();

string l\_Settle\_Ac\_Owner = dataRow[Lv\_Define.SETTLE\_AC\_OWNER].ToString();

lbl\_Batch\_No.Text = l\_Batch\_No;

lbl\_File\_Name.Text = l\_Upload\_FileName;

lbl\_Auth\_Status.Text = l\_Auth\_Status;

lbl\_Settle\_Trn\_ID.Text = l\_Settle\_Trn\_ID;

lbl\_Settle\_Ac\_No.Text = l\_Settle\_Ac\_No;

lbl\_Settle\_Ac\_Ccy.Text = l\_Settle\_Ac\_Ccy;

lbl\_Settle\_Ac\_Brn.Text = l\_Settle\_Ac\_Brn;

lbl\_Settle\_Ac\_Owner.Text = l\_Settle\_Ac\_Owner;

Close\_All\_Panel();

c\_Panel\_Info.Visible = true;

}

1. Bước 9:

Khi ấn vào nút Xem chi tiết

1. Bước 10:

Thực hiển gọi service để hiển thị chi tiết thông tin của file đã được upload vào hệ thống

1. Bước 11:

Gọi Service g\_Evn\_Payment.load\_PaymentBills(l\_Batch\_No, l\_Product\_ID) → Upload\_Payment.load\_PaymentBills(BatchNo, Product\_ID)

1. Bước 12:

DataTable dt\_PaymentBills = new DataTable("PAYMENT\_BILLS");

switch (Product\_ID)

{

case "SPC\_UPLOAD": dt\_PaymentBills = (new SPC\_Interface(para)).load\_PaymentBills(Batch\_No, Product\_ID); break;

case "CPC\_UPLOAD": dt\_PaymentBills = (new CPC\_Interface(para)).load\_PaymentBills(Batch\_No, Product\_ID); break;

case "HCM\_UPLOAD": dt\_PaymentBills = (new HCM\_Interface(para)).load\_PaymentBills(Batch\_No, Product\_ID); break;

case "HNI\_UPLOAD": dt\_PaymentBills = (new HNI\_Interface(para)).load\_PaymentBills(Batch\_No, Product\_ID); break;

case "NPC\_UPLOAD": dt\_PaymentBills = (new NPC\_Interface(para)).load\_PaymentBills(Batch\_No, Product\_ID); break;

}

Nếu Product\_ID bằng "SPC\_UPLOAD"

SPC\_Interface(para)).load\_PaymentBills(Batch\_No, Product\_ID)

SQL

select BATCH\_NO,MAKH,(select f.TENKH from DLTB\_MN\_CUSTOMER\_INFO f where f.MAKH = d.MAKH and rownum=1) TENKH,(select f.MASOTHUEKH from DLTB\_MN\_CUSTOMER\_INFO f where f.MAKH = d.MAKH and rownum=1) MASOTHUEKH,

MAHD,SOTIEN,MAGIAODICH,TRN\_CHANEL,DAINHD,TRANGTHAIGD,TRANGTHAIHUYGD,

Case when trangthaigd = '0' and trangthaihuygd = '9' then 'THÀNH CÔNG'

when trangthaigd = '1' and trangthaihuygd = '9' then 'KHÔNG THÀNH CÔNG'

when trangthaigd = '2' or trangthaihuygd = '2' then 'UNKNOW'

else 'KHÁC'

end

PAYMENT\_STAT,SETTLE\_TRN\_ID,D.NGAYGIO

from DLTB\_BILL\_UPLOAD\_MASTER m,DLTBD\_MN\_HOADON\_INFO d

where m.BATCH\_NO = :p\_Batch\_No and m.RECORD\_STAT in ('O', 'P')

and :p\_Product\_ID = :p\_Product\_ID" //KHONG DUNG

and d.TRN\_CHANEL = 'TELLER'

and d.TRN\_ID = m.SETTLE\_TRN\_ID

Parameter

<SP\_Parameter>

<Parameter>

<Name>p\_Batch\_No</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Product\_ID</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

Batch\_No, Product\_ID

Nếu Product\_ID bằng "CPC\_UPLOAD"

CPC\_Interface(para)).load\_PaymentBills(Batch\_No, Product\_ID)

SQL

select BATCH\_NO,MAKH,

(select f.TENKH from DLTB\_MT\_CUSTOMER\_INFO f where f.MAKH = d.MAKH and rownum=1) TENKH,(select f.MASOTHUEKH from DLTB\_MT\_CUSTOMER\_INFO f where f.MAKH = d.MAKH and rownum=1) MASOTHUEKH,MAHD,SOTIEN,MAGIAODICH,TRN\_CHANEL,

DAINHD,TRANGTHAIGD,TRANGTHAIHUYGD,

Case when trangthaigd = '0' and trangthaihuygd = '9' then 'THÀNH CÔNG'

when trangthaigd = '1' and trangthaihuygd = '9' then 'KHÔNG THÀNH CÔNG'

when trangthaigd = '2' or trangthaihuygd = '2' then 'UNKNOW'

else 'KHÁC'

end PAYMENT\_STAT,SETTLE\_TRN\_ID,D.NGAYGIO

from DLTB\_BILL\_UPLOAD\_MASTER m,DLTBD\_MT\_HOADON\_INFO d

where m.BATCH\_NO = :p\_Batch\_No and m.RECORD\_STAT in ('O', 'P')

and :p\_Product\_ID = :p\_Product\_ID and d.TRN\_CHANEL = 'TELLER'

and d.TRN\_ID = m.SETTLE\_TRN\_ID

Parameter

<SP\_Parameter>

<Parameter>

<Name>p\_Batch\_No</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Product\_ID</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

Batch\_No, Product\_ID

Nếu Product\_ID bằng "HCM\_UPLOAD"

HCM\_Interface(para)).load\_PaymentBills(Batch\_No, Product\_ID)

SQL

select BATCH\_NO,MAKH,(select f.TENKH from DLTB\_CUSTOMER\_INFO f where f.MAKH = d.MAKH and rownum=1)TENKH,(select f.MASOTHUEKH from DLTB\_CUSTOMER\_INFO f where f.MAKH = d.MAKH and rownum=1) MASOTHUEKH,

MAHD,SOTIEN,MAGIAODICH,TRN\_CHANEL,DAINHD,TRANGTHAIGD,TRANGTHAIHUYGD,

Case when trangthaigd = '0' and trangthaihuygd = '9' then 'THÀNH CÔNG' when trangthaigd = '1' and trangthaihuygd = '9' then 'KHÔNG THÀNH CÔNG'

when trangthaigd = '2' or trangthaihuygd = '2' then 'UNKNOW'

else 'KHÁC'

end PAYMENT\_STAT,SETTLE\_TRN\_ID,D.NGAYGIO

from DLTB\_BILL\_UPLOAD\_MASTER m,DLTBD\_HOADON\_INFO d

where m.BATCH\_NO = :p\_Batch\_No

and m.RECORD\_STAT in ('O', 'P')

and :p\_Product\_ID = :p\_Product\_ID

and d.TRN\_CHANEL IN ('AUTO', 'TELLER')

and d.TRN\_ID = m.SETTLE\_TRN\_ID

Parameter

<SP\_Parameter>

<Parameter>

<Name>p\_Batch\_No</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Product\_ID</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

Batch\_No, Product\_ID

Nếu Product\_ID bằng "HNI\_UPLOAD"

HNI\_Interface(para)).load\_PaymentBills(Batch\_No, Product\_ID)

SQL

select BATCH\_NO,MAKH,(select f.TENKH from DLTB\_HN\_CUSTOMER\_INFO f where f.MAKH = d.MAKH and rownum=1) TENKH,(select f.MASOTHUEKH from DLTB\_HN\_CUSTOMER\_INFO f where f.MAKH = d.MAKH and rownum=1) MASOTHUEKH,

MAHD,SOTIEN + TIENTHUE SOTIEN,MAGIAODICH,TRN\_CHANEL,DAINHD,TRANGTHAIGD,

TRANGTHAIHUYGD,case when trangthaigd = '0' and trangthaihuygd = '9' then 'THÀNH CÔNG'

when trangthaigd = '1' and trangthaihuygd = '9' then 'KHÔNG THÀNH CÔNG'

when trangthaigd = '2' or trangthaihuygd = '2' then 'UNKNOW'

else 'KHÁC'

end PAYMENT\_STAT,SETTLE\_TRN\_ID,D.NGAYGIO

from DLTB\_BILL\_UPLOAD\_MASTER m,DLTBD\_HN\_HOADON\_INFO d

where m.BATCH\_NO = :p\_Batch\_No and m.RECORD\_STAT in ('O', 'P')

and :p\_Product\_ID = :p\_Product\_ID

and d.TRN\_CHANEL in ('AUTO', 'TELLER')

and d.TRN\_ID = m.SETTLE\_TRN\_ID

Parameter

<SP\_Parameter>

<Parameter>

<Name>p\_Batch\_No</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Product\_ID</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

Batch\_No, Product\_ID

Nếu Product\_ID bằng "NPC\_UPLOAD"

NPC\_Interface(para)).load\_PaymentBills(Batch\_No, Product\_ID)

DataTable getStatus = new DataTable("AUTH\_STAT");

string query = "select AUTH\_STAT from LVBSMS.DLTB\_BILL\_UPLOAD\_MASTER where BATCH\_NO = '" + Batch\_No + "'";

db.SelectCommand(query, getStatus);

String status = "A";

if (getStatus.Rows.Count > 0)

{

status = getStatus.Rows[0]["AUTH\_STAT"].ToString();

}

if ("A".Equals(status))

{ write\_App\_Log(db.SelectCommand(C\_Parameter.DB\_SQL\_LOAD\_NPC\_PAYMENT\_BILLS, C\_Parameter.DB\_XML\_PARAMETERS\_LOAD\_NPC\_PAYMENT\_BILLS, new Object[] { Batch\_No, Product\_ID }, dt\_PaymentBills));

}

SQL

select BATCH\_NO,MAKH, select f.TENKH from DLTB\_MB\_CUSTOMER\_INFO f where f.MAKH = d.MAKH and rownum=1) TENKH,(select f.MASOTHUEKH from DLTB\_MB\_CUSTOMER\_INFO f where f.MAKH = d.MAKH and rownum=1) MASOTHUEKH,MAHD,

SOTIEN,MAGIAODICH, TRN\_CHANEL,DAINHD,TRANGTHAIGD,TRANGTHAIHUYGD,

Case when trangthaigd = '0' and trangthaihuygd = '9' then 'THÀNH CÔNG'

when trangthaigd = '1' and trangthaihuygd = '9' then 'KHÔNG THÀNH CÔNG'

when trangthaigd = '2' or trangthaihuygd = '2' then 'UNKNOW'

else 'KHÁC'

end PAYMENT\_STAT,SETTLE\_TRN\_ID,D.NGAYGIO

from DLTB\_BILL\_UPLOAD\_MASTER m,DLTBD\_MB\_HOADON\_INFO d

where m.BATCH\_NO = :p\_Batch\_No and m.RECORD\_STAT in ('O', 'P')

and :p\_Product\_ID = :p\_Product\_ID //KHONG DUNG

and d.TRN\_CHANEL = 'TELLER'

and d.TRN\_ID = m.SETTLE\_TRN\_ID

Parameter

<SP\_Parameter>

<Parameter>

<Name>p\_Batch\_No</Name>

<Type>VARCHAR2</Type> <Direction>Input</Direction>

</Parameter>

<Parameter>

<Name>p\_Product\_ID</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

Batch\_No, Product\_ID

1. Bước 13:

Trả lại thông tin của file

DataTable l\_DT\_Batch\_Detail = new DataTable(Lv\_Session.DT\_Upload\_Detail);

l\_DT\_Batch\_Detail = g\_Evn\_Payment.load\_PaymentBills(l\_Batch\_No, l\_Product\_ID);

Session[Lv\_Session.DT\_Upload\_Detail] = l\_DT\_Batch\_Detail;

1. Bước 14:

Hiện thị thông tin chi tiết file

DataTable l\_DT\_Fee\_Detail = (DataTable)Session[Lv\_Session.DT\_Upload\_Detail];

CommonFuntion.GridView\_Show(c\_GridView\_Detail, l\_DT\_Fee\_Detail, Lv\_Define.BATCH\_NO, 0);

1. Bước 14.1.1:

Khi click vào nút xuất báo cáo

1. Bước 14.1.2:

Gọi service để xuất báo cáo

1. Bước 14.1.3:

Gọi Service g\_Evn\_Payment.load\_PaymentBills(l\_Batch\_No, l\_dataTable.Rows[0]["PRODUCT\_ID"].ToString())→Upload\_Payment. load\_PaymentBills(BatchNo, Product\_ID)

1. Bước 14.1.4:

Làm tương tự như bước 12

1. Bước 14.1.5

Trả lại thông tin file

DataTable l\_dt\_PayBill = new DataTable();

DataTable l\_dataTable = (DataTable)(Session[Lv\_Session.DT\_Upload\_Master\_Selected]);

if (l\_dataTable != null && l\_dataTable.Rows.Count > 0)

{

string l\_Batch\_No = l\_dataTable.Rows[0][Lv\_Define.BATCH\_NO].ToString();

Session[Lv\_Define.BATCH\_NO] = l\_Batch\_No;

l\_dt\_PayBill = g\_Evn\_Payment.load\_PaymentBills(l\_Batch\_No, l\_dataTable.Rows[0]["PRODUCT\_ID"].ToString());

}

1. Bước 14.1.6:

Thực hiện xuất báo cáo

l\_dt\_PayBill.TableName = "EVN\_DETAIL\_PAYBILL";

g\_Report.SetDataSource(l\_dt\_PayBill);

Response.Buffer = false;

Response.ClearContent();

Response.ClearHeaders();

if (ddlReportType.SelectedValue == "pdf")

g\_Report.ExportToHttpResponse(ExportFormatType.PortableDocFormat, Response, true, "rpt\_evn\_upload\_paybill\_detail");

if (ddlReportType.SelectedValue == "excel")

{

path = Server.MapPath("reports/rpt\_evn\_upload\_paybill\_detail\_excel.rpt");

g\_Report.Load(path);

l\_dt\_PayBill.TableName = "EVN\_DETAIL\_PAYBILL";

g\_Report.SetDataSource(l\_dt\_PayBill);

Response.Buffer = false;

Response.ClearContent();

Response.ClearHeaders();

g\_Report.ExportToHttpResponse(ExportFormatType.ExcelRecord, Response, true, "rpt\_evn\_upload\_paybill\_detail");

}

1. Bước 14.2.1:

Thực hiện ấn Xuất Báo Cáo Theo Nhóm

1. Bước 14.2.2:

Gọi Service để xuất báo cáo

1. Bước 14.2.3:

Gọi Service g\_Evn\_Payment.load\_PaymentBills(l\_Batch\_No, l\_dataTable.Rows[0]["PRODUCT\_ID"].ToString())→Upload\_Payment. load\_PaymentBills(BatchNo, Product\_ID)

1. Bước 14.2.4:

Làm tương tụ bước 12

1. Bước 14.2.5:

Trả lại thông tin của file

DataTable l\_dt\_PayBill = new DataTable();

DataTable l\_dataTable = (DataTable)(Session[Lv\_Session.DT\_Upload\_Master\_Selected]);

l\_dt\_PayBill = g\_Evn\_Payment.load\_PaymentBills(l\_Batch\_No, l\_dataTable.Rows[0]["PRODUCT\_ID"].ToString());

l\_dt\_PayBill.Columns.Add("MADL");

1. Bước 14.2.6:

Xuất báo cáo

string path = "840".Equals(l\_Staff\_Branch\_Code) ? Server.MapPath("reports/rpt\_evn\_upload\_paybill\_group\_with\_ngaygio.rpt") : Server.MapPath("reports/rpt\_evn\_upload\_paybill\_group.rpt");

g\_Report.Load(path);

l\_dt\_PayBill.TableName = "EVN\_DETAIL\_PAYBILL";

g\_Report.SetDataSource(l\_dt\_PayBill);

Response.Buffer = false;

Response.ClearContent();

Response.ClearHeaders();

if ("840".Equals(l\_Staff\_Branch\_Code))

{

if (ddlReportType.SelectedValue == "pdf")

g\_Report.ExportToHttpResponse(ExportFormatType.PortableDocFormat, Response, true,"rpt\_evn\_upload\_paybill\_group\_with\_ngaygio");

if (ddlReportType.SelectedValue == "excel")

g\_Report.ExportToHttpResponse(ExportFormatType.ExcelRecord, Response, true,

"rpt\_evn\_upload\_paybill\_group\_with\_ngaygio");

}

else

{

if (ddlReportType.SelectedValue == "pdf")

g\_Report.ExportToHttpResponse(ExportFormatType.PortableDocFormat, Response, true,"rpt\_evn\_upload\_paybill\_group");

if (ddlReportType.SelectedValue == "excel")

g\_Report.ExportToHttpResponse(ExportFormatType.ExcelRecord, Response, true, "rpt\_evn\_upload\_paybill\_group");

}

//end

g\_Report.Close();

g\_Report.Dispose();

Response.End();

1. Bước 15:

Khi ấn vào nút Xem chi tiết hóa đơn

1. Bước 16:

Thực hiện gọi Service để xem chi tiết hạch toán

1. Bước 17:

Gọi Service g\_Evn\_Payment.load\_SettlementTransaction(l\_Trans\_ID) → Upload\_Payment.load\_SettlementTransaction(String p\_Settlement\_Trn\_ID)

1. Bước 18:

DataSet ds\_trn = new DataSet();

ds\_trn.Tables.Add(load\_SettlementTransactionInfo(p\_Settlement\_Trn\_ID));

ds\_trn.Tables.Add(load\_SettlementPost(p\_Settlement\_Trn\_ID));

return ds\_trn;

Hàm load\_SettlementTransactionInfo(p\_Settlement\_Trn\_ID)

SQL

select t.TRANSACTION\_ID , t.TRN\_DESC, t.MAKER\_ID, t.CHECKER\_ID, t.TRN\_BRN, DECODE(t.AUTH\_STATUS,'A','Đã duyệt hạch toán','U','Chưa duyệt hạch toán','Không xác định') AUTH\_STATUS,TO\_CHAR(TRN\_DT, 'DD/MM/YYYY') TRN\_DT, TO\_CHAR(VALUE\_DT, 'DD/MM/YYYY') VALUE\_DT,b.BRANCH\_NAME, (b.BRANCH\_ADDR1 || ' ' || b.BRANCH\_ADDR2 || ' ' || b.BRANCH\_ADDR3) BRANCH\_ADDR,(select FIELD\_VAL\_2 from cstm\_function\_userdef\_fields where function\_id ='STDBRANC' AND REC\_KEY = t.TRN\_BRN || '~') MST from DATBD\_TRANSACTION t, sttm\_branch b

where T.TRANSACTION\_ID = :p\_Settlement\_Trn\_ID and b.BRANCH\_CODE = t.TRN\_BRN

Parameter

<SP\_Parameter>

<Parameter>

<Name>p\_Settlement\_Trn\_ID</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

p\_Settlement\_Trn\_ID

Hàm load\_SettlementPost(p\_Settlement\_Trn\_ID)

SQL

select P.AC\_NO,P.AC\_BRANCH,P.AC\_CCY,P.AMOUNT, p.CUST\_GL, P.DRCR\_IND, P.EVENT, p.EVENT\_SEQ\_NO,DECODE(p.TRN\_STATUS,'S','Hạch toán thành công','F','Hạch toán lỗi','W', 'Hạch toán timeout','C','Hủy tạo hạch toán',NULL, 'Chưa hạch toán','Không xác định') trn\_status,P.CORE\_REF\_NO, TO\_CHAR(CORE\_TRN\_DT, 'DD/MM/YYYY') CORE\_TRN\_DT,(select AC\_DESC from STTM\_CUST\_ACCOUNT WHERE CUST\_AC\_NO = P.AC\_NO) AC\_DESC

from DLTBD\_TRANSACTION\_POST p, DATBD\_TRANSACTION d

where d.TRANSACTION\_ID = :p\_Settlement\_Trn\_ID

and d.TRANSACTION\_ID = p.TRANSACTION\_ID

and d.LAST\_EVENT\_SEQ\_NO = p.EVENT\_SEQ\_NO

and nvl(p.TRN\_STATUS, 'I') in ('I', 'S','F')

Parameter

<SP\_Parameter>

<Parameter>

<Name>p\_Settlement\_Trn\_ID</Name>

<Type>VARCHAR2</Type>

<Direction>Input</Direction>

</Parameter>

</SP\_Parameter>

p\_Settlement\_Trn\_ID

1. Bước 19:

Trả lại kết quả

DataTable l\_DT\_Batch\_Post = new DataTable(Lv\_Session.DT\_Upload\_Detail);

DataSet l\_Ds\_Batch\_Post = new DataSet();

l\_Ds\_Batch\_Post = g\_Evn\_Payment.load\_SettlementTransaction(l\_Trans\_ID);

l\_DT\_Batch\_Post = l\_Ds\_Batch\_Post.Tables[1];

Session[Lv\_Session.DT\_Upload\_Detail] = l\_DT\_Batch\_Post;

1. Bước 20:

Hiển thị thông tin chi tiết hạch toán

DataTable l\_DT\_Fee\_Post = (DataTable)Session[Lv\_Session.DT\_Upload\_Detail];

CommonFuntion.GridView\_Show(c\_GridView\_Post, l\_DT\_Fee\_Post, Lv\_Define.BATCH\_NO, 0);

1. Bước 21:

Hiện thị danh sách POST DETAIL

DataTable l\_DT\_Fee\_Post = (DataTable)Session[Lv\_Session.DT\_Upload\_Detail];

CommonFuntion.GridView\_Show(c\_GridView\_Post, l\_DT\_Fee\_Post, Lv\_Define.BATCH\_NO, 0);

1. Bước 21.1:

Khi click vào nút xuất báo cáo

1. Bước 21.2:

Gọi service để xuất báo cáo

1. Bước 21.3:

Gọi Service g\_Evn\_Payment.load\_SettlementTransaction(l\_Settle\_Trn\_ID) →Upload\_Payment.load\_SettlementTransaction(p\_Settlement\_Trn\_ID)

1. Bước 21.4:

Làm tương tự bước 18

1. Bước 21.5:

Trả lại kết quả

DataSet l\_ds\_Settle\_Post = new DataSet();

DataTable l\_dt\_Settle\_Post = new DataTable();

DataTable l\_dataTable = (DataTable)(Session[Lv\_Session.DT\_Upload\_Master\_Selected]);

if (l\_dataTable != null && l\_dataTable.Rows.Count > 0)

{

string l\_Batch\_No = l\_dataTable.Rows[0][Lv\_Define.BATCH\_NO].ToString();

string l\_Settle\_Trn\_ID = l\_dataTable.Rows[0][Lv\_Define.SETTLE\_TRN\_ID].ToString();

Session[Lv\_Define.BATCH\_NO] = l\_Batch\_No;

l\_ds\_Settle\_Post = g\_Evn\_Payment.load\_SettlementTransaction(l\_Settle\_Trn\_ID);

l\_ds\_Settle\_Post.Tables[0].TableName = "TRANSACTION\_INFO";

l\_ds\_Settle\_Post.Tables[1].TableName = "EVN\_DETAIL\_POST";

l\_dt\_Settle\_Post = l\_ds\_Settle\_Post.Tables[1];

}

1. Bước 21.6:

Xuất báo cáo

string path = Server.MapPath("reports/rpt\_evn\_upload\_post\_detail.rpt");

g\_Report.Load(path);

g\_Report.SetDataSource(l\_ds\_Settle\_Post);

g\_Report.SetParameterValue("batchNo", Session[Lv\_Define.BATCH\_NO].ToString());

Response.Buffer = false;

Response.ClearContent();

Response.ClearHeaders();

if (ddlReportType.SelectedValue == "pdf")

g\_Report.ExportToHttpResponse(ExportFormatType.PortableDocFormat, Response, true, "rpt\_evn\_upload\_post\_detail");

if (ddlReportType.SelectedValue == "excel")

g\_Report.ExportToHttpResponse(ExportFormatType.ExcelRecord, Response, true, "rpt\_evn\_upload\_post\_detail");

g\_Report.Close();

g\_Report.Dispose();

Response.End();

1. Bước 22:

Thực hiện ấn nút Back

1. Bước 23:

Gọi hàm Reload\_Branch\_Master()

1. Bước 24:

Gọi lại hàm Search\_Branch\_Master\_List

l\_Begin\_Date = txt\_Begin\_Dt.Value;

l\_End\_Date = txt\_End\_Dt.Value;

lbl\_Error.Text = Search\_Branch\_Master\_List(l\_Begin\_Date, l\_End\_Date);

1. Bước 25:

Thực hiện tương tự bước 4

1. Bước 26:

Trả lại danh sách file đã được upload vào hệ thống

DataTable dt\_batch = new DataTable("BATCH\_BRANCH");

db.SelectCommand(SQL\_CMD, C\_Parameter.DB\_XML\_PARAMETERS\_GET\_BATCH\_BY\_BRANCH, new Object[] { UserID, AuthStatus, RecordStatus, Settled }, dt\_batch)

1. Bước 27:

Hiển thị lên gridView

c\_GridView\_Master.DataSource = l\_DataTable;

c\_GridView\_Master.DataBind();

if (l\_DataTable.Rows.Count <= 0)

{

Sreturn = "Không có dữ liệu!";

}

## **Duyệt gạch nợ**



**Step 3:** Gọi Service get Batch by Branch

* Gọi Service1. getBatch\_ByBranch(UserID, AuthStatus, RecordStatus, Settled, l\_Begin\_Dt, l\_End\_Dt)

**Step 5:** Select batch

* SQL\_CMD:

select m.\*, to\_char(m.upload\_date, 'dd/MM/yyyy') supload\_date ,DECODE(m.AUTH\_STAT, 'A', DECODE(m.SETTLE\_DT, NULL, 'Đã duyệt file', 'Đã hạch toán'), 'U', 'Chưa duyệt file', 'Không xác định') FILE\_STAT from DLTB\_BILL\_UPLOAD\_MASTER m,

(

select u.USERNAME, NVL(u.ACTIVE\_BRANCH, u.HOME\_BRANCH) ACTIVE\_BRANCH

from ASTB\_USER u

where u.USERNAME = :p\_UserName

and u.RECORD\_STATUS = 'A'

) U

where m.BRANCH\_CODE = u.ACTIVE\_BRANCH

* if (l\_Begin\_Dt.Length > 6)

SQL\_CMD += " and trunc(m.upload\_date)>=to\_Date('"+l\_Begin\_Dt+"', 'dd/MM/yyyy')"

* if (l\_End\_Dt.Length > 6)

SQL\_CMD += " and trunc(m.upload\_date)<=to\_Date('"+l\_End\_Dt+"', 'dd/MM/yyyy')" + "\n"

* SQL\_CMD += and m.RECORD\_STAT in ('O', 'P','W','K')

and m.AUTH\_STAT = decode(:p\_Auth\_Stat, 'A', 'A', 'U', 'U', m.AUTH\_STAT) and m.RECORD\_STAT = decode(:p\_Record\_Stat, 'O', 'O', 'P', 'P', m.RECORD\_STAT)

and :p\_Settled = decode(:p\_Settled, '-1', :p\_Settled, '0', decode(m.SETTLE\_DT, null, '0', '-1'), '1', decode(m.SETTLE\_DT, null, '-1', '1'), '2', decode(m.SETTLE\_DT, null, decode(m.RECORD\_STAT, 'O', '2', '-1'), decode(m.RECORD\_STAT, 'P', '2', '-1')))

* return dt\_batch

**Step 11:** Gọi Service batch detail

* Gọi Service1. load\_PaymentBills(BatchNo, Product\_ID)

**Step 13:** Select batch detail

SQL:

Select BATCH\_NO, MAKH,

(

select f.TENKH

from DLTB\_CUSTOMER\_INFO f   
where f.MAKH = d.MAKH

and rownum=1) TENKH,

(

select f.MASOTHUEKH

from DLTB\_CUSTOMER\_INFO f

where f.MAKH = d.MAKH and rownum=1

) MASOTHUEKH, MAHD, SOTIEN, MAGIAODICH, TRN\_CHANEL, DAINHD, TRANGTHAIGD, TRANGTHAIHUYGD,

case when trangthaigd = '0' and trangthaihuygd = '9' then 'THÀNH CÔNG'

when trangthaigd = '1' and trangthaihuygd = '9' then 'KHÔNG THÀNH CÔNG'

when trangthaigd = '2' or trangthaihuygd = '2' then 'UNKNOW'

else 'KHÁC'

end  
PAYMENT\_STAT, SETTLE\_TRN\_ID, D.NGAYGIO

from DLTB\_BILL\_UPLOAD\_MASTER m, DLTBD\_HOADON\_INFO d

where m.BATCH\_NO = :p\_Batch\_No

and m.RECORD\_STAT in ('O', 'P')

and :p\_Product\_ID = :p\_Product\_ID

and d.TRN\_CHANEL IN ('AUTO', 'TELLER')

and d.TRN\_ID = m.SETTLE\_TRN\_ID

* return dt\_PaymentBills

**Step 17:** Gọi service gạch nợ hóa đơn

* Gọi Service1. PayBills(CheckerID, BatchNo)

**Step 19:** Gạch nợ hóa đơn

pkg\_EVN\_Upload\_Payment.Fn\_Pay\_Bills

Parameters:

p\_Checker\_ID varchar2,  
 p\_Batch\_No varchar2,

p\_Branch\_Code out nocopy varchar2,

p\_Settle\_Trn\_ID out nocopy number,

p\_Product\_ID out nocopy varchar2

Variables:

v\_Home\_Branch varchar2(5)  
 v\_Rcd\_Err PKG\_DOM\_UTIL.RCD\_ERROR  
 v\_Count number;

Get Transaction để duyệt

* Update   
  (  
   select M.\*

from DLTB\_BILL\_UPLOAD\_MASTER m   
 where M.BATCH\_NO = p\_Batch\_No  
 and M.CHECKER\_ID = p\_Checker\_ID  
 and M.RECORD\_STAT = 'O'  
 and M.AUTH\_STAT = 'A'  
 and m.SETTLE\_DT is null  
) M  
set RECORD\_STAT = 'P'  
returning M.BRANCH\_CODE, M.PRODUCT\_ID, M.SETTLE\_TRN\_ID  
into p\_Branch\_Code, p\_Product\_ID, p\_Settle\_Trn\_ID;

* Update   
  (  
   select M.\* from DLTB\_BILL\_UPLOAD\_MASTER m   
   where M.BATCH\_NO = p\_Batch\_No  
   and M.CHECKER\_ID = p\_Checker\_ID  
   and M.RECORD\_STAT = 'P'  
   and M.AUTH\_STAT = 'A'  
   and m.SETTLE\_DT is null  
  ) M  
  set RECORD\_STAT = 'O’;

Kiểm tra hóa đơn

* case p\_Product\_ID when 'HCM\_UPLOAD' then  
   select count(\*) into v\_Count  
   from DLTBD\_HOADON\_INFO  
   where TRN\_ID = to\_char(p\_Settle\_Trn\_ID)  
   and NVL(TRANGTHAIGD, '1') not in ('0', '2')  
   and nvl(TRANGTHAIHUYGD, '9') = '9';  
   if v\_Count = 0 then  
   v\_Rcd\_Err.ERROR\_CODE := '9004';  
   v\_Rcd\_Err.MESSAGE := 'Không có hoá đơn nào được thanh toán.   
   BATCH\_NO is ' || p\_Batch\_No || '.';  
   goto Final\_Process;
* else   
   if sql%rowcount <> 1 then  
   v\_Rcd\_Err.ERROR\_CODE := '9002';  
   v\_Rcd\_Err.MESSAGE := 'Batch không sẵn sàng để thanh toán.

BATCH\_NO, Branch\_Code is '||p\_Batch\_No||', '||v\_Home\_Branch||'.';

end if  
 v\_Rcd\_Err.ERROR\_CODE := '0000';  
 v\_Rcd\_Err.MESSAGE := 'Init\_Pay\_Bills is executed successfully. Batch\_No, Branch\_Code is '||p\_Batch\_No||', '|| v\_Home\_Branch || '.';

exception  
 when others then  
 v\_Rcd\_Err.ERROR\_CODE := '9999';  
 v\_Rcd\_Err.MESSAGE := 'System Error: ' || sqlerrm|| chr(10) || 'Error At: ' || DBMS\_UTILITY.format\_error\_backtrace;

Gạch nợ hóa đơn

String TrangThai\_GD = "2";

int result = service.BankConfirm(dr\_payment\_bill["MAKH"].ToString(), dr\_payment\_bill["MAHD"].ToString(), double.Parse(dr\_payment\_bill["SOTIEN"].ToString()), NgayGiaoDich, MaGiaoDich, Branch\_Code, Trn\_Chanel, OWNER\_BANK\_ID, Print\_Bill\_New)

if (result == 0)

TrangThai\_GD = "0";  
else

TrangThai\_GD = "1";

**Step 21:** Update trạng thái hóa đơn

* Update

(

select NGAYGIO, TRANGTHAIGD, TRANGTHAIHUYGD, MAGIAODICH, HOADON\_ID, TRN\_CHANEL, DAINHD  
from DLTBD\_HOADON\_INFO  
where ROWID = :p\_Row\_ID

)

set NGAYGIO = to\_date(:p\_NgayGiaoDich, 'dd/mm/yyyy hh24:mi:ss')  
, TRANGTHAIGD = :p\_TrangThai\_GD  
, TRANGTHAIHUYGD = :p\_TrangThai\_HuyGD  
where NVL(TRANGTHAIGD, '1') not in ('0', '2')  
and nvl(TRANGTHAIHUYGD, '9') = '9'

## **Duyệt Báo Có EVN**



**Step 2:** Gọi Service lấy list batch

* Gọi Service1. getBatch\_ByBranch(UserID, AuthStatus, RecordStatus, Settled, l\_Begin\_Dt, l\_End\_Dt)

**Step 4:** Select list batch

* SQL\_CMD:

select m.\*, to\_char(m.upload\_date, 'dd/MM/yyyy') supload\_date ,DECODE(m.AUTH\_STAT, 'A', DECODE(m.SETTLE\_DT, NULL, 'Đã duyệt file', 'Đã hạch toán'), 'U', 'Chưa duyệt file', 'Không xác định') FILE\_STAT from DLTB\_BILL\_UPLOAD\_MASTER m,

(

select u.USERNAME, NVL(u.ACTIVE\_BRANCH, u.HOME\_BRANCH) ACTIVE\_BRANCH

from ASTB\_USER u

where u.USERNAME = :p\_UserName

and u.RECORD\_STATUS = 'A'

) U

where m.BRANCH\_CODE = u.ACTIVE\_BRANCH

* + if (l\_Begin\_Dt.Length > 6)

SQL\_CMD += " and trunc(m.upload\_date)>=to\_Date('"+l\_Begin\_Dt+"', 'dd/MM/yyyy')"

* + if (l\_End\_Dt.Length > 6)

SQL\_CMD += " and trunc(m.upload\_date)<=to\_Date('"+l\_End\_Dt+"', 'dd/MM/yyyy')" + "\n"

SQL\_CMD += and m.RECORD\_STAT in ('O', 'P','W','K') and m.AUTH\_STAT = decode(:p\_Auth\_Stat, 'A', 'A', 'U', 'U', m.AUTH\_STAT) and m.RECORD\_STAT = decode(:p\_Record\_Stat, 'O', 'O', 'P', 'P', m.RECORD\_STAT) and :p\_Settled = decode(:p\_Settled, '-1', :p\_Settled, '0', decode(m.SETTLE\_DT, null, '0', '-1'), '1', decode(m.SETTLE\_DT, null, '-1', '1'), '2', decode(m.SETTLE\_DT, null, decode(m.RECORD\_STAT, 'O', '2', '-1'), decode(m.RECORD\_STAT, 'P', '2', '-1')))

* return dt\_batch

**Step 8:** Gọi service tạo thông tin hạch toán

* Gọi Service1**.**Init\_Settlement(checkerID, BatchNo)

**Step 10:** Tạo thông tin hạch toán

pkg\_EVN\_Upload\_Payment.Fn\_Init\_Settlement

Parameters:

p\_Checker\_ID varchar2  
 p\_Batch\_No varchar2

Variables:

v\_Home\_Branch varchar2(5);  
 v\_Rcd\_Err PKG\_DOM\_UTIL.RCD\_ERROR;  
 v\_Rcd\_Trn DATBD\_TRANSACTION%ROWTYPE;  
 v\_Rcd\_Settle\_Post DLTBD\_TRANSACTION\_POST%ROWTYPE;  
 v\_Product\_ID DLTB\_BILL\_UPLOAD\_MASTER.PRODUCT\_ID%TYPE;  
 v\_PG\_Service\_ID DLTM\_UPLOAD\_PRODUCT.PG\_SERVICE\_ID%TYPE;  
 v\_PG\_Product\_Code DLTM\_UPLOAD\_PRODUCT.PG\_PRODUCT\_CODE%TYPE;

update  
(

select M.\*  
 from DLTB\_BILL\_UPLOAD\_MASTER m  
 where M.BATCH\_NO = p\_Batch\_No  
 and BRANCH\_CODE = v\_Home\_Branch  
 and M.RECORD\_STAT = 'O'  
 and M.AUTH\_STAT = 'A'  
 and m.SETTLE\_DT is null

)

set SETTLE\_DT = v\_Rcd\_Trn.TRN\_DT  
 RECORD\_STAT = 'P'

returning SETTLE\_TRN\_ID, PRODUCT\_ID, SETTLE\_AC\_NO, SETTLE\_AC\_CCY,   
 SETTLE\_AC\_BRN, SETTLE\_AC\_CUST\_GL

into v\_Rcd\_Trn.TRANSACTION\_ID, v\_Product\_ID,v\_Rcd\_Settle\_Post.AC\_NO,   
 v\_Rcd\_Settle\_Post.AC\_CCY, v\_Rcd\_Settle\_Post.AC\_BRANCH,   
 v\_Rcd\_Settle\_Post.CUST\_GL;   
if sql%rowcount <> 1 then

v\_Rcd\_Err.ERROR\_CODE := '9002';

v\_Rcd\_Err.MESSAGE := 'Batch không sẵn sàng để hạch toán nợ có. BATCH\_NO, Branch\_Code is ' || p\_Batch\_No || ', ' || v\_Home\_Branch || '.';

select PG\_SERVICE\_ID, PG\_PRODUCT\_CODE  
into v\_PG\_Service\_ID, v\_PG\_Product\_Code

from DLTM\_UPLOAD\_PRODUCT p  
where PRODUCT\_ID = v\_Product\_ID;

//kiem tra transaction chua duyet

select t.\* into v\_Rcd\_Trn  
from DATBD\_TRANSACTION t  
where TRANSACTION\_ID = v\_Rcd\_Trn.TRANSACTION\_ID  
 and t.RECORD\_STATUS='O'  
 and t.AUTH\_STATUS is null  
 AND t.MODULE = c\_Module\_Name

for update nowait;

exception

when no\_data\_found then

v\_Rcd\_Err.ERROR\_CODE := '9002';  
 v\_Rcd\_Err.MESSAGE := 'Batch không sẵn sàng để hạch toán nợ có.

BATCH\_NO, Transaction ID is ' || p\_Batch\_No || ', ' || v\_Rcd\_Trn.TRANSACTION\_ID || '.';

goto Final\_Process;

// check hoa dơn duyet bi exception

case v\_Product\_ID when 'HCM\_UPLOAD' then

declare

v\_lc\_Count number(6);   
 select count(\*) into v\_lc\_Count  
 from DLTBD\_HOADON\_INFO B  
 where b.TRN\_ID = to\_char(v\_Rcd\_Trn.TRANSACTION\_ID)  
 and b.TRN\_CHANEL in ( c\_Chanel\_AUTO, c\_Chanel\_TELLER)  
 and b.TRANGTHAIGD = '2';

if v\_lc\_Count > 0 then  
 v\_Rcd\_Err.ERROR\_CODE := '9002';  
 v\_Rcd\_Err.MESSAGE := 'Batch không sẵn sàng để tạo bút toán nợ có.

TRANGTHAIGD là UNKNOW. BATCH\_NO, TRANSACTION\_ID

is ' || p\_Batch\_No || ', ' ||

v\_Rcd\_Trn.TRANSACTION\_ID || '.';

goto Final\_Process;

else

v\_Rcd\_Err.ERROR\_CODE := '9901';  
 v\_Rcd\_Err.MESSAGE := 'Sản phẩm không tồn tại. PRODUCT\_ID is ' ||

v\_Product\_ID || '.';

goto Final\_Process;

if v\_Rcd\_Settle\_Post.AMOUNT <= 0 then

v\_Rcd\_Err.ERROR\_CODE := '9002';  
 v\_Rcd\_Err.MESSAGE := 'Batch không sẵn sàng để hạch toán nợ có (Số tiền thanh toán bằng 0). BATCH\_NO, Branch\_Code is ' || p\_Batch\_No || ', ' || v\_Home\_Branch || '.';

goto Final\_Process;

insert into DLTBD\_TRANSACTION\_POST

(

TRANSACTION\_ID,TRN\_CODE,EVENT,EVENT\_SEQ\_NO,  
 AC\_NO,AC\_CCY,AC\_BRANCH,CUST\_GL,  
 DRCR\_IND,AMOUNT,AMOUNT\_TAG

) values

(

v\_Rcd\_Trn.TRANSACTION\_ID, v\_Rcd\_Trn.TRN\_CODE, v\_Rcd\_Trn.EVENT\_CODE, v\_Rcd\_Trn.LAST\_EVENT\_SEQ\_NO, v\_Rcd\_Settle\_Post.AC\_NO, v\_Rcd\_Settle\_Post.AC\_CCY, v\_Rcd\_Settle\_Post.AC\_BRANCH, v\_Rcd\_Settle\_Post.CUST\_GL, 'D', v\_Rcd\_Settle\_Post.AMOUNT, 'BILL\_AMT'

)

v\_Rcd\_Err.ERROR\_CODE := '0000';  
v\_Rcd\_Err.MESSAGE := 'Settlement is executed successfully. Batch\_No, Branch\_Code is ' || p\_Batch\_No || ', ' || v\_Home\_Branch || '.';

exception

when others then  
 rollback;  
 v\_Rcd\_Err.ERROR\_CODE := '9999';  
 v\_Rcd\_Err.MESSAGE := 'System Error: ' || sqlerrm|| chr(10) || 'Error At: ' || DBMS\_UTILITY.format\_error\_backtrace;

goto Final\_Process;

**Step 14:** Gọi Service load hạch toán info

* **Gọi Service1.**load\_SettlementTransaction(p\_Settlement\_Trn\_ID)

**Step 16:** Select hạch toán info

* load\_SettlementTransactionInfo(p\_Settlement\_Trn\_ID)
  + select t.TRANSACTION\_ID , t.TRN\_DESC, t.MAKER\_ID, t.CHECKER\_ID, t.TRN\_BRN, DECODE(t.AUTH\_STATUS,'A','Đã duyệt hạch toán','U','Chưa duyệt hạch toán','Không xác định') AUTH\_STATUS, TO\_CHAR(TRN\_DT, 'DD/MM/YYYY') TRN\_DT, TO\_CHAR(VALUE\_DT, 'DD/MM/YYYY') VALUE\_DT, b.BRANCH\_NAME, (b.BRANCH\_ADDR1 || ' ' || b.BRANCH\_ADDR2 || ' ' || b.BRANCH\_ADDR3) BRANCH\_ADDR,

(

select FIELD\_VAL\_2

from cstm\_function\_userdef\_fields

where function\_id ='STDBRANC' AND REC\_KEY = t.TRN\_BRN || '~'

) MST

from DATBD\_TRANSACTION t, sttm\_branch b

where T.TRANSACTION\_ID = :p\_Settlement\_Trn\_ID

and b.BRANCH\_CODE = t.TRN\_BRN

* load\_SettlementPost(p\_Settlement\_Trn\_ID)
  + select P.AC\_NO,P.AC\_BRANCH,P.AC\_CCY,P.AMOUNT, p.CUST\_GL, P.DRCR\_IND, P.EVENT, p.EVENT\_SEQ\_NO, DECODE(p.TRN\_STATUS,'S','Hạch toán thành công','F','Hạch toán lỗi','W', 'Hạch toán timeout','C','Hủy tạo hạch toán',NULL, 'Chưa hạch toán','Không xác định') trn\_status, P.CORE\_REF\_NO, TO\_CHAR(CORE\_TRN\_DT, 'DD/MM/YYYY') CORE\_TRN\_DT,

(

select AC\_DESC

from STTM\_CUST\_ACCOUNT WHERE CUST\_AC\_NO = P.AC\_NO

) AC\_DESC

from DLTBD\_TRANSACTION\_POST p, DATBD\_TRANSACTION d

where d.TRANSACTION\_ID = :p\_Settlement\_Trn\_ID

and d.TRANSACTION\_ID = p.TRANSACTION\_ID

and d.LAST\_EVENT\_SEQ\_NO = p.EVENT\_SEQ\_NO

and nvl(p.TRN\_STATUS, 'I') in ('I', 'S','F')

* Return: ds\_trn;

**Step 19:** Gọi Service duyệt hạch toán

* Gọi Service1. Approve\_Settlement(CheckerID, BatchNo, Trn\_Desc)

**Step 21+23:** Duyệt hạch toán

pkg\_EVN\_Upload\_Payment.Fn\_Approve\_Settlement

Parameters:

p\_Checker\_ID varchar2

p\_Batch\_No varchar2  
 p\_Trn\_Desc varchar2

Constants:

c\_Core\_Prod\_Code varchar2(5):= 'UPEBK'

c\_Core\_Txn\_Code varchar2(3):= 'EBK';

Variables:

v\_Home\_Branch varchar2(5)

v\_Rcd\_Err PKG\_DOM\_UTIL.RCD\_ERROR  
 v\_Trn\_ID DATBD\_TRANSACTION.TRANSACTION\_ID%TYPE  
 v\_Trn\_Brn DATBD\_TRANSACTION.TRN\_BRN%TYPE  
 v\_Trn\_Desc DATBD\_TRANSACTION.TRN\_DESC%TYPE  
 v\_Value\_Dt DATBD\_TRANSACTION.VALUE\_DT%TYPE  
 v\_Maker\_ID DATBD\_TRANSACTION.MAKER\_ID%TYPE

v\_Last\_Event\_Seq\_No DATBD\_TRANSACTION.LAST\_EVENT\_SEQ\_NO%TYPE

v\_Upload\_Status boolean:= false  
 v\_MSG\_Id varchar2(20) := ''

1. update DLTB\_BILL\_UPLOAD\_MASTER status

update

(

select M.\*

from DLTB\_BILL\_UPLOAD\_MASTER m  
 where M.BATCH\_NO = p\_Batch\_No  
 and BRANCH\_CODE = v\_Home\_Branch  
 and M.RECORD\_STAT = 'P'  
 and M.AUTH\_STAT = 'A'  
 and m.SETTLE\_DT is not null

)

Set RECORD\_STAT = 'K'

returning SETTLE\_TRN\_ID

into v\_Trn\_ID;

if sql%rowcount <> 1 then

v\_Rcd\_Err.ERROR\_CODE := '9002';

v\_Rcd\_Err.MESSAGE := 'Batch không sẵn sàng để xác nhận hạch toán. BATCH\_NO, Branch\_Code :' || p\_Batch\_No || ', ' || v\_Home\_Branch || '.';

goto Final\_Process;

1. Start process batch

Update  
(

select t.\*

from DATBD\_TRANSACTION t  
 where TRANSACTION\_ID = v\_Trn\_ID

and t.RECORD\_STATUS='O'

and t.AUTH\_STATUS is null

AND t.module = c\_Module\_Name

)

Set VALUE\_DT = trunc(sysdate)

AUTH\_STATUS = 'A'

CHECKER\_DT = sysdate  
 CHECKER\_ID = p\_Checker\_ID  
 TRN\_DESC = p\_Trn\_Desc

returning TRN\_BRN, TRN\_DESC, VALUE\_DT, MAKER\_ID, LAST\_EVENT\_SEQ\_NO

into v\_Trn\_Brn, v\_Trn\_Desc, v\_Value\_Dt, v\_Maker\_ID, v\_Last\_Event\_Seq\_No;

if sql%rowcount <> 1 then  
 v\_Rcd\_Err.ERROR\_CODE := '9002';  
 v\_Rcd\_Err.MESSAGE := 'Batch không sẵn sàng để xác nhận hạch toán (Không tìm thấy giao dịch). BATCH\_NO, Transaction ID : ' || p\_Batch\_No || ', ' || v\_Trn\_ID || '.';

goto Final\_Process;

declare

v\_lc\_Sql\_Cmd varchar2(1000)  
v\_lc\_Core\_Ref\_No varchar2(30)  
v\_lc\_Trn\_Desc DATBD\_TRANSACTION.TRN\_DESC%TYPE  
v\_lc\_Err\_String varchar2(1000)

v\_lc\_Sql\_Cmd :=

select AC\_NO, AC\_BRANCH, AC\_CCY, null, DECODE(SIGN(AMOUNT),1,'C','D') DRCR\_IND, ABS(FCY\_AMOUNT) FCY\_AMOUNT, NULL, ABS(AMOUNT) AMOUNT

from

(

select p.AC\_NO, p.AC\_BRANCH, p.AC\_CCY, null AC\_OWNER,  
decode(P.DRCR\_IND, 'D', -1, 'C', 1, 0) \* p.AMOUNT AMOUNT,null FCY\_AMOUNT  
from pkg\_Global\_Dom\_Application.c\_gl\_Owner\_Schema. DLTBD\_TRANSACTION\_POST P  
where p.TRANSACTION\_ID = v\_Trn\_ID  
and p.EVENT\_SEQ\_NO = v\_Last\_Event\_Seq\_No  
and p.TRN\_STATUS is null

) where AMOUNT <> 0;

v\_lc\_Trn\_Desc := ebank.pkg\_util.Fn\_Reduce\_Char\_Sign(v\_Trn\_Desc);

ebank.PKG\_CORE\_SWITCH.PR\_UL\_TRANSFER\_V2 (v\_lc\_Sql\_Cmd, v\_Trn\_Brn, v\_lc\_Trn\_Desc, c\_Core\_Prod\_Code, c\_Core\_Txn\_Code, v\_Value\_Dt, C\_MAKER\_CORE,C\_CHECKER\_CORE, v\_lc\_Err\_String, v\_lc\_Core\_Ref\_No, v\_MSG\_Id,'N', 'EVN\_LVBIS');

* case substr(nvl(v\_lc\_Err\_String,'1#99'),3,2)
  + when '00' then

update DLTBD\_TRANSACTION\_POST p

set P.CORE\_REF\_NO = v\_lc\_Core\_Ref\_No,

P.CORE\_TRN\_DT = (select max(today) from sttm\_dates where branch\_code = p.AC\_BRANCH),

P.TRN\_STATUS = 'S',

where P.TRANSACTION\_ID = v\_Trn\_ID

and p.EVENT\_SEQ\_NO = v\_Last\_Event\_Seq\_No

and p.TRN\_STATUS is null;

v\_Upload\_Status := true;

v\_Rcd\_Err.ERROR\_CODE := '0000';

v\_Rcd\_Err.ERROR\_TYPE := PKG\_DOM\_UTIL.C\_GL\_DEBUG\_MSG;

v\_Rcd\_Err.MESSAGE := 'Approve\_Settlement is executed successfully. Batch\_No, Branch\_Code is ' || p\_Batch\_No || ', ' || v\_Home\_Branch || '.';

* + when '82' then

v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0006';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Cust\_Data\_Err;

v\_Rcd\_Err.Message := 'Tài khoản không đủ tiền thanh toán.';

* + when '83' then

v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0006';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Cust\_Data\_Err;

v\_Rcd\_Err.Message := 'Tài khoản không đủ tiền thanh toán.';

* + when '84' then

v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0004';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'CCY của giao dịch và sản phẩm không giống nhau.';

* + when '80' then

v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0080';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'Core exception.';

* + when '90' then

v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0090';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'Core timeout.';

* + when '91' then

v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0091';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'Exception when process ESB response.';

* + when '98' then

v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0098';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'Core failed.';

* + when '99' then

v\_Upload\_Status := false;

v\_Rcd\_Err.Error\_Code := '0099';

v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;

v\_Rcd\_Err.Message := 'Exception befor calling ESB.';

* + else

v\_Upload\_Status := false;  
v\_Rcd\_Err.Error\_Code :='9990';  
v\_Rcd\_Err.Error\_Type := pkg\_Dom\_Util.c\_gl\_Sys\_Data\_Err;  
v\_Rcd\_Err.Message := 'Cập nhật vào core không thành công - ' || v\_lc\_Err\_String;

* if(v\_Upload\_Status = true) then
  + update

(

select M.\*  
 from DLTB\_BILL\_UPLOAD\_MASTER m  
 where M.BATCH\_NO = p\_Batch\_No  
 and BRANCH\_CODE = v\_Home\_Branch  
 and M.RECORD\_STAT = 'K'  
 and M.AUTH\_STAT = 'A'  
 and m.SETTLE\_DT is not null

)

Set RECORD\_STAT = 'O';

* + update DATBD\_TRANSACTION t

set t.XREF = v\_MSG\_Id

where t.TRANSACTION\_ID = to\_char(v\_Trn\_ID);

* if (substr(v\_lc\_Err\_String,3,2) = '80' or substr(v\_lc\_Err\_String,3,2) = '90' or substr(v\_lc\_Err\_String,3,2) = '91') then
  + update DLTBD\_TRANSACTION\_POST p

set P.TRN\_STATUS = 'W'  
,P.POST\_DATE = sysdate  
,P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)

where P.TRANSACTION\_ID = v\_Trn\_ID  
and P.EVENT\_SEQ\_NO = v\_Last\_Event\_Seq\_No;

* + update DATBD\_TRANSACTION t

set t.XREF = v\_MSG\_Id  
where t.TRANSACTION\_ID = to\_char(v\_Trn\_ID);

* + update

(

select M.\*  
 from DLTB\_BILL\_UPLOAD\_MASTER m  
 where M.BATCH\_NO = p\_Batch\_No  
 and BRANCH\_CODE = v\_Home\_Branch  
 and M.RECORD\_STAT = 'K'  
 and M.AUTH\_STAT = 'A'  
 and m.SETTLE\_DT is not null

)

set RECORD\_STAT = 'W';

* else
  + update DLTBD\_TRANSACTION\_POST p  
    set P.TRN\_STATUS = 'F',

P.POST\_DATE = sysdate,

P.CORE\_TRN\_DT = (select today from sttm\_dates where branch\_code = v\_Home\_Branch)

where P.TRANSACTION\_ID = v\_Trn\_ID

and P.EVENT\_SEQ\_NO = v\_Last\_Event\_Seq\_No;

* + update DATBD\_TRANSACTION t

set t.XREF = v\_MSG\_Id

where t.TRANSACTION\_ID = to\_char(v\_Trn\_ID);

* + update

(

select M.\*  
 from DLTB\_BILL\_UPLOAD\_MASTER m  
 where M.BATCH\_NO = p\_Batch\_No

and BRANCH\_CODE = v\_Home\_Branch

and M.RECORD\_STAT = 'K'

and M.AUTH\_STAT = 'A'  
 and m.SETTLE\_DT is not null

)

set RECORD\_STAT = 'O';

* insert into IBTSD\_MAP\_TRANS\_MSG(ID, TRANS\_ID, MSG\_ID, VALUE\_DT, CHECKER\_ID, MODULE\_ID)

values(IBTSD\_MAP\_TRANS\_MSG\_SEQ.nextval, v\_Trn\_ID, v\_MSG\_Id, sysdate, p\_Checker\_ID, 'EVN\_UPLOAD');

* exception
  + when others then

v\_Rcd\_Err.ERROR\_CODE := '9999';  
v\_Rcd\_Err.ERROR\_TYPE := PKG\_DOM\_UTIL.C\_GL\_SYS\_ERR;  
v\_Rcd\_Err.MESSAGE := 'System Error: ' || sqlerrm|| chr(10) || 'Error At: ' || DBMS\_UTILITY.format\_error\_backtrace;

**Step 26:** Gọi Service hủy hạch toán

* **Gọi Service1.**Cancel\_Settlement(l\_Staff\_ID, l\_Batch\_No)

**Step 28:** Hủy hạch toán

pkg\_EVN\_Upload\_Payment.Fn\_Cancel\_Settlement

Parameters:

p\_Checker\_ID varchar2   
 p\_Batch\_No varchar2

Variables:

v\_Home\_Branch varchar2(5);

v\_Rcd\_Err PKG\_DOM\_UTIL.RCD\_ERROR  
 v\_Rcd\_Trn DATBD\_TRANSACTION%ROWTYPE;

* Update

(

select M.SETTLE\_DT, M.RECORD\_STAT, M.SETTLE\_TRN\_ID  
 from DLTB\_BILL\_UPLOAD\_MASTER m  
 where M.BATCH\_NO = p\_Batch\_No  
 and BRANCH\_CODE = v\_Home\_Branch  
 and M.RECORD\_STAT = 'P'  
 and M.AUTH\_STAT = 'A'  
 and m.SETTLE\_DT is not null

)

set SETTLE\_DT = null,  
 RECORD\_STAT = 'O'

returning SETTLE\_TRN\_ID

into v\_Rcd\_Trn.TRANSACTION\_ID;

if sql%rowcount <> 1 then

v\_Rcd\_Err.ERROR\_CODE := '9002';  
v\_Rcd\_Err.ERROR\_TYPE := PKG\_DOM\_UTIL.C\_GL\_USER\_DATA\_ERR;  
v\_Rcd\_Err.MESSAGE := 'Batch kh?ng s?n s?ng d? hu? h?ch to?n.

BATCH\_NO, Branch\_Code is ' || p\_Batch\_No || ', ' ||   
 v\_Home\_Branch || '.';

* select t.\*

into v\_Rcd\_Trn

from DATBD\_TRANSACTION t

where TRANSACTION\_ID = v\_Rcd\_Trn.TRANSACTION\_ID

and t.RECORD\_STATUS='O'   
and t.AUTH\_STATUS is null  
AND t.MODULE = c\_Module\_Name

for update nowait;

exception

when no\_data\_found then

v\_Rcd\_Err.ERROR\_CODE := '9002';  
v\_Rcd\_Err.ERROR\_TYPE := PKG\_DOM\_UTIL.C\_GL\_USER\_DATA\_ERR;  
v\_Rcd\_Err.MESSAGE := 'Batch kh?ng s?n s?ng d? hu? h?ch to?n. BATCH\_NO, Transaction ID is ' || p\_Batch\_No || ', ' || v\_Rcd\_Trn.TRANSACTION\_ID || '.';

goto Final\_Process;

* update

(

select TRN\_STATUS

from DLTBD\_TRANSACTION\_POST p

where p.TRANSACTION\_ID = v\_Rcd\_Trn.TRANSACTION\_ID  
 and p.EVENT\_SEQ\_NO = v\_Rcd\_Trn.LAST\_EVENT\_SEQ\_NO  
 and p.TRN\_STATUS is null

)

set TRN\_STATUS = 'C';

v\_Rcd\_Err.ERROR\_CODE := '0000';

v\_Rcd\_Err.ERROR\_TYPE := PKG\_DOM\_UTIL.C\_GL\_DEBUG\_MSG;

v\_Rcd\_Err.MESSAGE := 'Cancel\_Settlement is executed successfully. Batch\_No, Branch\_Code is '|| p\_Batch\_No || ', ' ||v\_Home\_Branch || '.';

* exception

when others then

rollback;

v\_Rcd\_Err.ERROR\_CODE := '9999';

v\_Rcd\_Err.ERROR\_TYPE := PKG\_DOM\_UTIL.C\_GL\_SYS\_ERR;  
v\_Rcd\_Err.MESSAGE := 'System Error: ' || sqlerrm|| chr(10) || 'Error At: ' || DBMS\_UTILITY.format\_error\_backtrace;