



Coding GuideLine FrameWork

Người tạo: Nguyễn Quang Hoàng

Coding GuideLine FrameWork

I)	Giới thiệu	6
1)	Tổng quan	6
2)	Môi trường phát triển	6
3)	Mô hình FrameWork.....	7
4)	Cáu trúc Solution	10
II)	Menu	13
III)	Phân quyền.....	15
IV)	Sử dụng các control chuẩn có sẵn.....	15
1)	TextBox.....	15
2)	CheckBox.....	17
3)	ComboBox	18
a)	Tạo một combobox nếu chưa tồn tại.....	18
b)	Tạo các field(cột) được hiển thị trong combobox(Sử dụng bảng sysField).....	20
c)	Tạo Field sử dụng ComboBox trên màn hình.....	20
d)	Khai báo ngôn ngữ cho column trên combobox	23
4)	ComboCheckList	23
5)	DatePicker.....	26
6)	SpecialControl.....	29
a)	Cáu trúc.....	29
b)	Khởi tạo.....	29
c)	Ví dụ.....	29
7)	DateTimePicker	30
a)	Cáu trúc: Gồm 2 phần là DatePicker và TimePicker	30
b)	Cách tạo	31
8)	Upload.....	34
a)	Mô tả: Control Upload có dạng như sau:.....	34
b)	Cách tạo	34
9)	RadioButton	36
a)	Mô tả: Control RadioButton có dạng như thế này	36
b)	Cách tạo	36
10)	TimePicker.....	40

a)	Mô tả: Control TimePicker có dạng như thế này	40
b)	Cách tạo	40
11)	SpinEditNumber	42
a)	Mô tả	42
b)	Cách tạo	43
12)	TextArea	45
a)	Mô tả	45
b)	Cách tạo	45
13)	Item Star Level.....	47
a)	Mô tả	47
b)	Cách tạo	48
14)	MultiSelect.....	50
a)	Mô tả	50
b)	Cách tạo	50
15)	Select Object	52
a.	Mô tả	52
b.	Cách tạo	53
V)	Các bước tạo màn hình chuẩn	56
1)	ContentMaster(Màn hình danh mục/tìm kiếm).....	56
a)	Các bước bắt buộc.....	56
b)	Customize	58
2)	Màn hình nhập liệu không lưới detail(Thêm/Sửa).....	75
a)	Các bước bắt buộc.....	75
b)	Customize	77
c)	Validation trên màn hình nhập liệu.....	86
d)	Hàm mở rộng onAfterInsertSuccess	87
3)	Màn hình nhập liệu có lưới detail	88
a)	Các bước bắt buộc.....	88
b)	Customize	91
c)	EditorTempalte trên lưới.....	105
4)	Màn hình xem chi tiết không có lưới detail (ViewNoDetail)	108
a)	Các bước bắt buộc.....	108
b)	Customize	110

5)	Màn hình xem chi tiết có lưới detail 1 (ViewMasterDetail)	117
a)	Các bước bắt buộc.....	117
b)	Customize	119
6)	Màn hình xem chi tiết có lưới detail 2 (ViewMasterDetail2)	128
a)	Các bước bắt buộc.....	128
b)	Customize	131
c)	Action (Hành động chỉ dùng ở màn hình xem chi tiết có lưới detail 2).....	146
7)	Report(Báo cáo)	148
a)	Cấu trúc.....	148
b)	Tạo các Group Report cho từng module.....	148
c)	Tạo các Report	149
d)	Tạo các mã màn hình tương ứng với các report(để phân quyền).....	150
e)	Tạo ReportFilter để quy định các Field trên màn hình lọc	150
f)	Thêm Template cho các báo cáo có sử dụng nhiều template.....	150
g)	Custom Report (Layout lại report)	151
8)	Màn hình chọn (PopupSelectData)	153
a)	Tạo bảng cho màn hình.....	154
b)	Tạo màn hình	154
c)	Gọi màn hình từ javascript.....	155
d)	Customer	155
VI)	Các tính năng dùng chung.....	159
1)	Ghi chú.....	159
2)	Lịch sử	160
3)	Gửi mail	161
a)	Group gửi mail	161
b)	Màn hình gửi mail (CMNF9005).....	162
4)	Đính kèm.....	163
a)	Group đính kèm	163
b)	Màn hình đính kèm	165
VII)	Khai báo ngôn ngữ.....	166
1)	Khai báo ngôn ngữ trong DBAdmin	166
2)	Tạo dữ liệu trong DataBase Khách hàng	166
a)	Thêm ngôn ngữ mới cho các [control chung]	166

b)	Thêm ngôn ngữ mới cho một màn hình.....	167
c)	Thêm message mới	167
d)	Thêm ngôn ngữ hay message dùng Storeprocedure	168
e)	Qui định đặt các file message và language trên SVN	168
3)	Khai báo trong Project	168
a)	Message JavaScript.....	168
b)	Hiển thị ngôn ngữ cho control chung, menu (trong .cshtml)	168
c)	Hiển thị ngôn ngữ khi cho một màn hình (trong ViewModel và .cshtml)	168
VIII)	Sử dụng Custom JavaScript và Custom CSS	168
1)	Javascript custom	168
a)	Bind và Unbind sự kiện các button chuẩn	169
2)	CSS custom.....	169

I) Giới thiệu

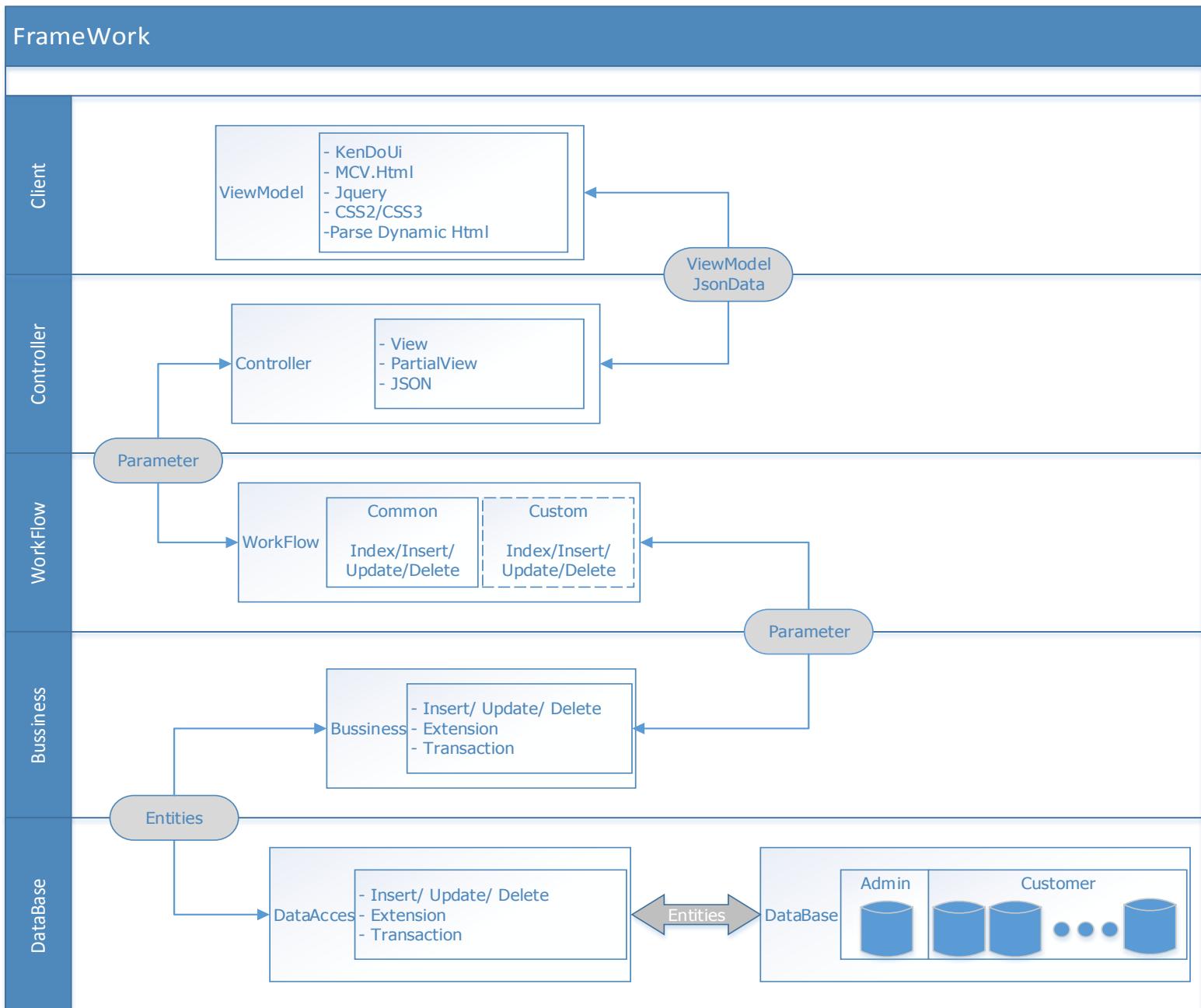
1) Tổng quan

- ❖ Tài liệu cho người phát triển có cái nhìn chung về Framework 9.0
- ❖ Cấu trúc, ý nghĩa thư mục
- ❖ Tài liệu sẽ được cập nhật liên tục để đảm bảo tiến độ công việc của dự án

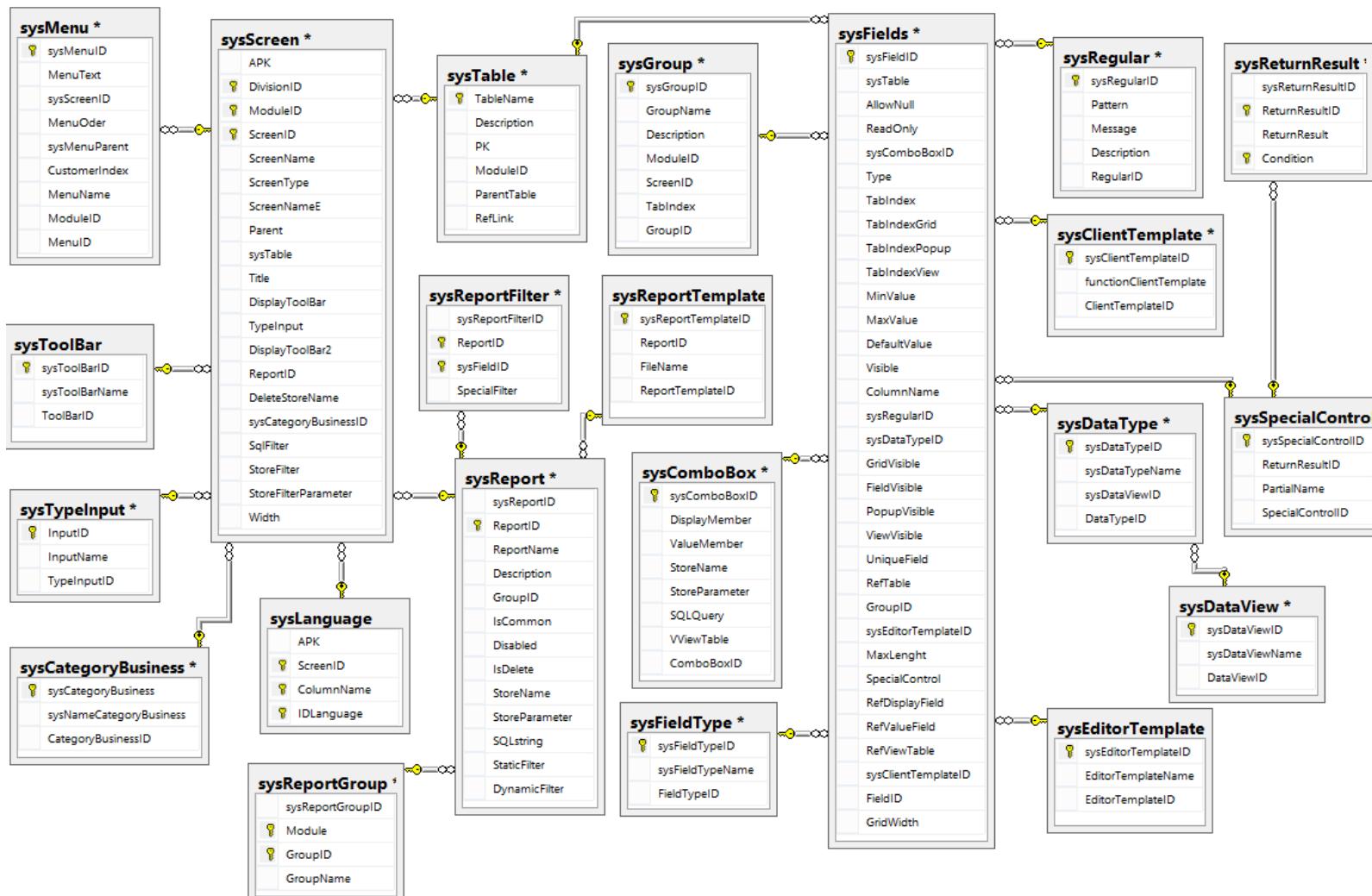
2) Môi trường phát triển

Phát triển trên các trình duyệt web	
Ngôn ngữ sử dụng	
Công cụ sử dụng để thiết kế	
Thư viện hỗ trợ	+ Kendo UI Complete for ASP.NET MVC, v2013.3.1127, JQuery-1.8.3
Database	SQL Server 2012
Mô hình sử dụng	MVC4

3) Mô hình FrameWork



Cấu trúc FrameWork



Cấu trúc DataBase Admin

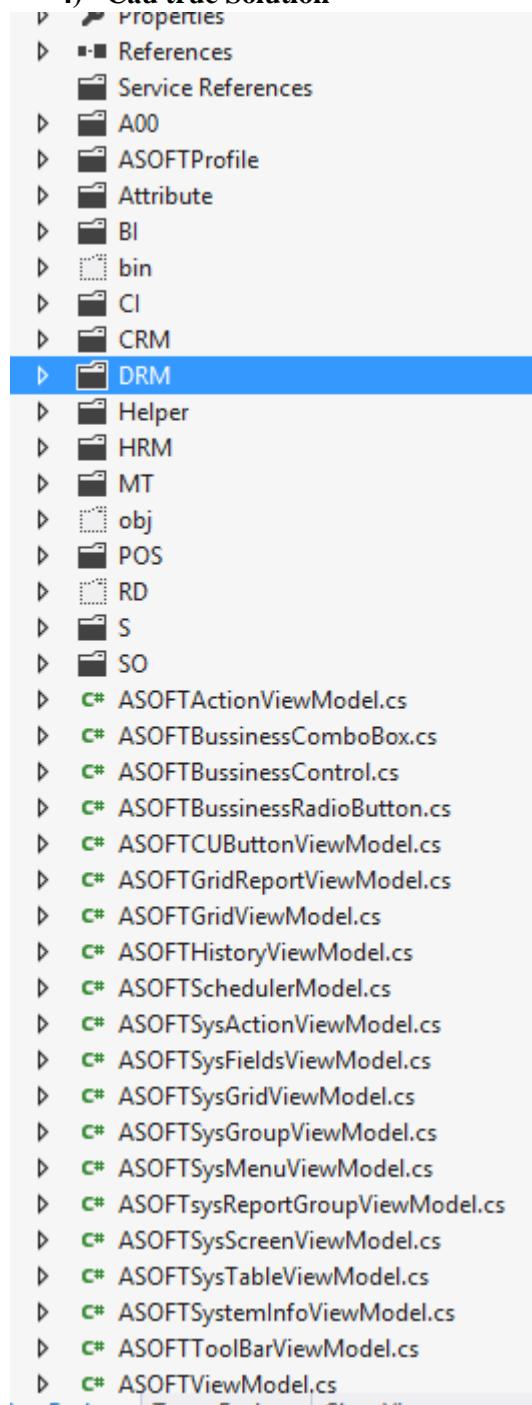
Chức năng của từng bảng trong Database Admin

sysComboBox	Các combobox trong ERP, quy định cách lấy dữ liệu cho combobox
sysDataType	Các kiểu dữ liệu của ERP
sysDataView	Các kiểu dữ liệu hiển thị trên màn hình
sysFields	Các Field sử dụng trong ERP Gồm cả field của combobox và báo cáo
sysFieldType	Các loại control trên màn hình
sysGroup	Gom nhóm các control của master trong màn hình view
sysMenu	Lưu trữ menu của chương trình

sysRegular	Lưu trữ pattern và message để validation theo regex
sysReport	Các report sử dụng trong ERP
sysReportFilter	Quy định các control sẽ được sử dụng làm điều kiện lọc cho report
sysReportGroup	Định nghĩa các nhóm báo cáo sẽ xuất hiện trên màn hình view report
sysReportTemplate	Quy định các mẫu báo cáo mà report sử dụng
sysScreen	Lưu trữ màn hình trong ERP
sysTable	Các bảng có trong DB khách hàng
sysToolBar	Lưu trữ các loại toolbar sử dụng trên màn hình
sysTypeInput	Các dạng màn hình nhập liệu
sysSpecialControl	Định nghĩa các control đặc biệt cho field, ví dụ control từ ngày đến ngày, từ kì đến kì.
sysReturnResult	Định nghĩa câu truy vấn theo điều kiện của các control đặc biệt khi sử dụng lọc chuẩn.
sysClientTemplate	Định nghĩa ClientTemplate cho Field khi sử dụng trên lưới (ví dụ: link trên lưới, button trên lưới, checkbox trên lưới)
sysEditorTemplate	Định nghĩa EditorTemplapte cho Field khi sử dụng trên lưới (Ví dụ: tìm kiếm trên lưới(Autocomplete), Combobox trên lưới(ComboboxGrid))
sysRadioButton	Định nghĩa các radiobutton có trên màn hình
sysAction	Định nghĩa action ở màn hình ViewmasterDetail2

NOTE: Khi tạo dữ liệu cho các bảng trong DatabaseAdmin, phải để giá trị **null cho các trường không sử dụng, không được để giá trị **rỗng(khoảng trắng)****

4) Cấu trúc Solution



ASOFTSysFieldsViewModel: Class ViewModel tham chiếu cho các control trên màn hình (Dùng chung cho tất cả màn hình)

ASOFTSysScreenViewModel: Class ViewModel chung tham chiếu header, title trên màn hình

ASOFTMenuViewModel: Class ViewModel chung tham chiếu menu trên màn hình

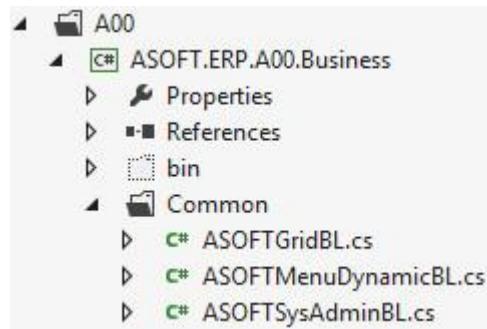
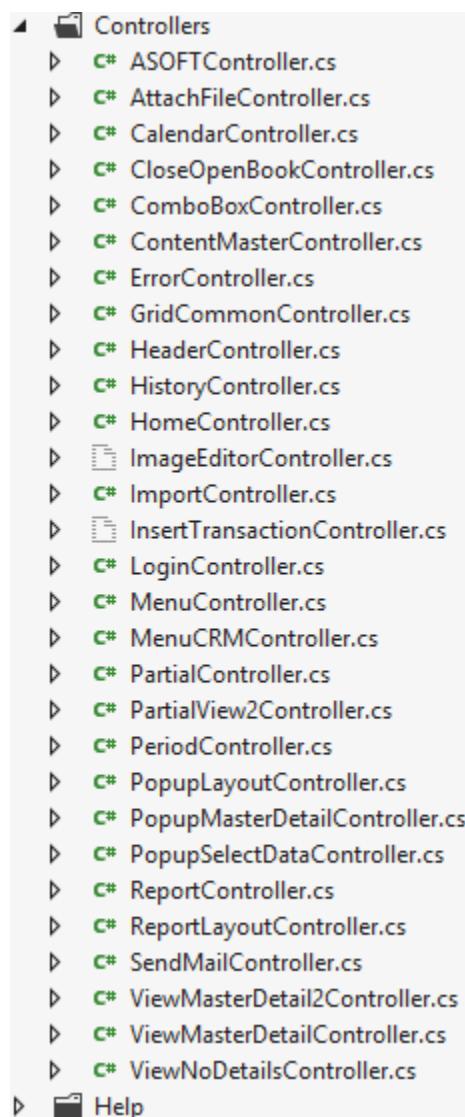
ASOFTSysReportGroupViewModel: Class ViewModel chung để tham chiếu ReportGroup trên màn hình

ASOFTSysGroupViewModel: class ViewModel chung dùng để tham chiếu các group của master trên màn hình

ASOFTToolBarrViewModel: class ViewModel chung dùng để tham chiếu các toolbar trên màn hình

ASOFTSysTableViewModel: Class ViewModel chung để tham chiếu sysTable trên màn hình

ASOFTSysActionViewModel: Class ViewModel chung để tham chiếu sysAction trên màn hình.



ContentMasterController: Controller dùng chung cho loại màn hình danh mục

PopupLayoutController: Controller dùng chung cho màn hình nhập liệu không detail.

PopupMasterDetailController: Controller dùng chung cho màn hình nhập liệu có detail.

ViewNoDetailsController: Controller dùng chung cho màn hình chi tiết không có detail.

ViewMasterDetailController: Controller dùng chung cho màn hình chi tiết có detail.

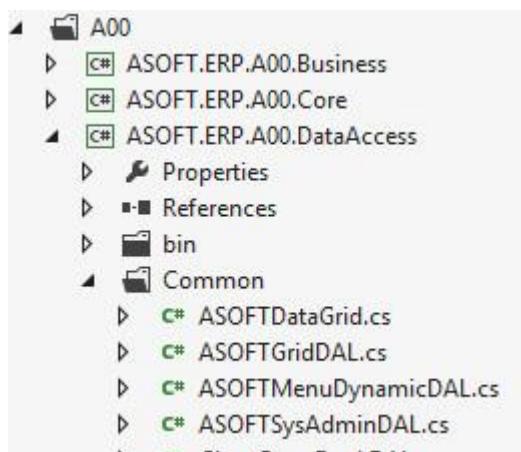
ViewMasterDetail2Controller: Controller dùng chung cho màn hình chi tiết có detail 2.

ReportLayoutController: Controller dùng chung cho report

ASOFTSysAdminBL: các hàm thêm/xóa/sửa db Admin (Fields,Screen,Menu)

ASOFTGridBL: chứa các hàm xử lý trên lưới, thêm xóa sửa db ERP

ASOFTMenuDynamicBL: chứa hàm parse động menu

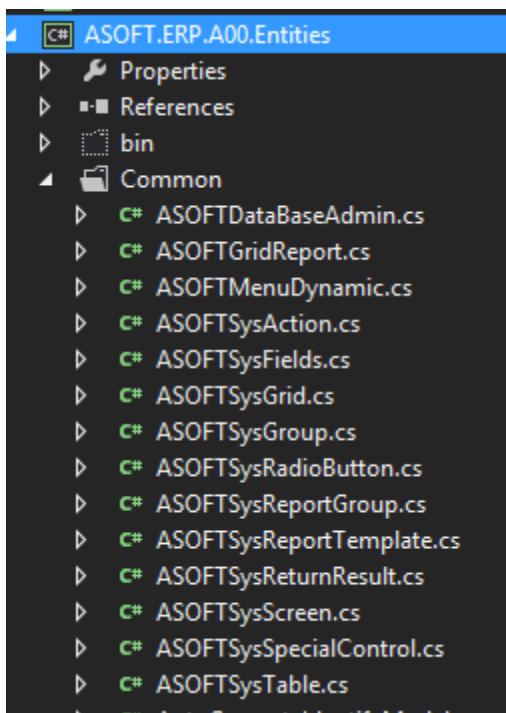


ASOFTAdminDAL: các hàm thêm/xóa/sửa db Admin (Fields,Screen,Menu)

ASOFTGridDAL: chứa các hàm xử lý trên lưới thêm xóa sửa db ERP

ASOTMenuDynamicDAL: chứa hàm parse động menu

ASOFTDataGrid: chứa hàm load dữ liệu vào lưới



ASOTSysFields: class tham chiếu đến các column trong bảng sysFields của database Admin.

ASOTSysGrid: class Model chứa dữ liệu được hiển thị trên lưới.(lưới sử dụng dữ liệu Entities)

ASOTMenuDynamic: class tham chiếu đến các column trong bảng sysMenu của database Admin.

ASOTSysScreen: class tham chiếu đến các column trong bảng sysScreen của database Admin.

ASOTSysTable: class tham chiếu đến các column trong bảng sysTable của database Admin.

ASOTSysReportGroup: class tham chiếu đến các column trong bảng sysReportGroup của database Admin.

ASOTSysAction: class tham chiếu đến các column trong bảng sysAction của database Admin.

ASOTSysRadioButton: class tham chiếu đến các column trong bảng sysRadioButton của database Admin.

ASOTSysReportTemplate: class tham chiếu đến các column trong bảng sysReportTemplate của database Admin.

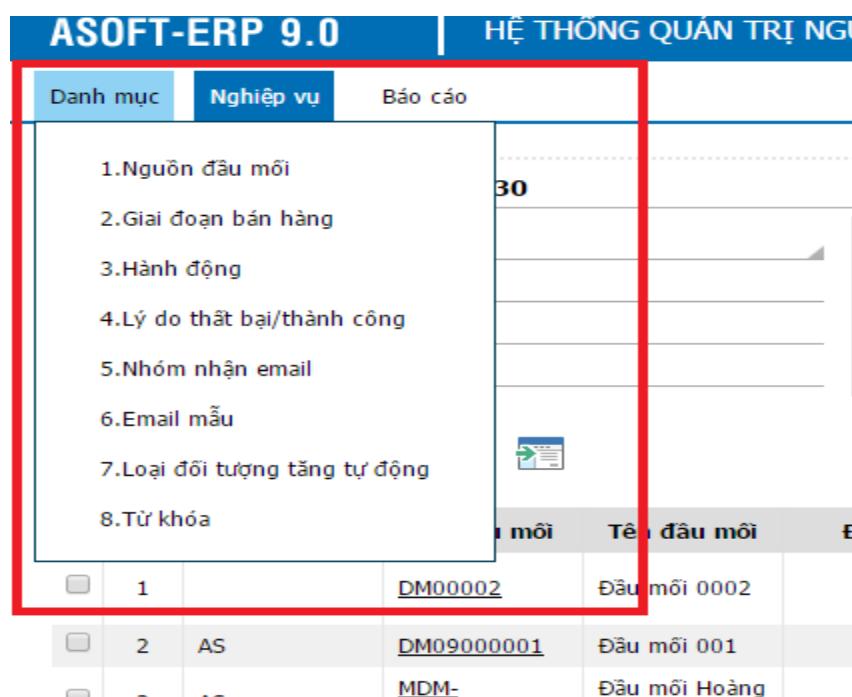
ASOTSysReturnResult: class tham chiếu đến các column trong bảng sysReturnResult của database Admin.

ASOFTSysSpecialControl: class tham chiếu đến các column trong bảng sysSpecialControl của database Admin.

Bổ sung thêm mục hướng dẫn chi tiết lưu script DB Admin

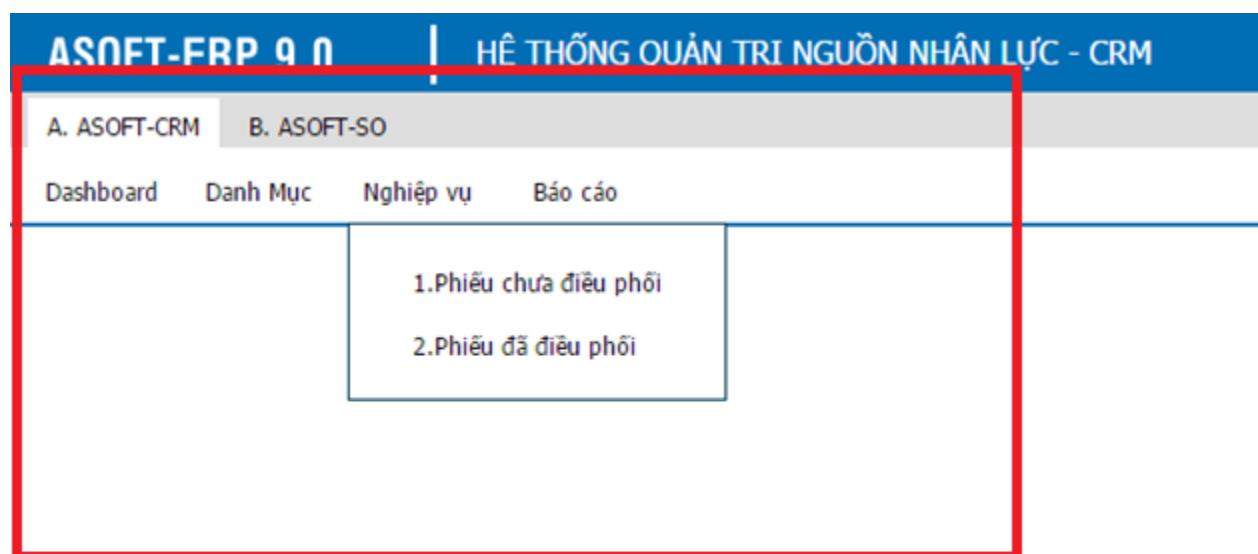
II) Menu

- Menu mới: không phân theo module



ID	Tên đầu mối	Địa chỉ
1	Đầu mối 0002	Đầu mối 0002
2	Đầu mối 001	Đầu mối 001
3	Đầu mối Hoàng	Đầu mối Hoàng

- Menu cũ: Phân theo module



The screenshot shows a software interface titled "ASOFT-FRP 9.0" and "HỆ THỐNG QUẢN TRỊ NGUỒN NHÂN LỰC - CRM". A red box highlights a section of the interface. Inside this red box, there is a sub-menu with two items: "1.Phiếu chưa điều phối" and "2.Phiếu đã điều phối". The main menu bar includes "A. ASOFT-CRM" and "B. ASOFT-SO", with sub-options "Dashboard", "Danh Mục", "Nghệp vụ", and "Báo cáo".

ASOFT®-CRM

- Để thêm 1 menu vào màn hình ta chỉ cần thêm một dòng vào bảng sysMenu

sysMenu	
sysMenuID	Id của Menu (tự sinh động)
MenuText	Text Menu (VD: Danh Mục Sản Phẩm)
sysScreenID	Tên màn hình của menu đó
MenuOder	Thứ tự của menu
sysMenuParent	Menu cha
CustomerIndex	Khách hàng được nhìn thấy menu
ModuleName	Tên menu
ModuleID	Module mà menu thuộc
MenuLevel	Cấp của menu

- Sử dụng script

```
declare @MenuText nvarchar(100)
declare @sysScreenID nvarchar(50)
declare @MenuOder int
declare @sysMenuParent int
declare @CustomerIndex int
declare @ModuleName nvarchar(50)
declare @ModuleID nvarchar(50)
declare @MenuID nvarchar(50)
declare @MenuLevel nvarchar(50)
```

```

set @sysMenuParent=(select top 1 sysMenuID from sysMenu where MenuID = 'ASOFTCRM')
set @MenuText=N'CRM_NghiepVu'
set @sysScreenID=null
set @MenuOder=1
set @CustomerIndex=-1
set @ModuleName=N'ASOFTCRM_NghiepVu'
set @ModuleID=N'ASOFTCRM'
set @MenuID=N'ASOFTCRM_NghiepVu'
set @MenuLevel=1
If not exists(select top 1 1 from [dbo].[sysMenu] where [MenuID] = N'ASOFTCRM_NghiepVu')Begin
insert into
sysMenu(MenuText,sysScreenID,MenuOder,sysMenuParent,CustomerIndex,ModuleName,ModuleID,MenuID)values(@MenuText,@
End

```

- Lưu ý: Menu cấp cuối cùng mới có mã màn hình. Không phải menu cuối thì ScreenID = null.
Cách đặt MenuName: {MenuName parent}_{Name}_{Mã màn hình Danh mục}_{Mã màn hình cập nhật}
- Menu cũ: level menu bắt đầu từ 0, Menu mới: level menu bắt đầu từ 1.
- Để xét level cho menu bằng cách thêm cấp độ menu vào column MenuLevel trong bảng sysMenu

III) Phân quyền

- Phân quyền vẫn giống với FrameWork cũ. Thêm vào bảng AT1403
- Tham khảo tài liệu trên SVN:
SVN\10_DOCUMENT\15_REGULATION_DOCUMENT\ASOFT_ERP9_ScreenPermission GuideLine.docx.

IV) Sử dụng các control chuẩn có sẵn

1) TextBox

Mã Sản Phẩm	<input type="text"/>
Tên Sản Phẩm	<input type="text"/>

Để tạo 1 textbox ta vào bảng sysfield thêm 1 dòng dữ liệu vào:

sysFields	
sysFieldID	Id của Textbox (tự sinh động)
sysTable	Textbox thuộc bảng nào
AllowNull	Textbox có phải nhập hay không (Requir, 1: không, 0: có)
sysComboBoxID	Trường này không phải nhập (dùng cho combobox)
Type	1 (Textbox)
TabIndex	Thứ tự xuất hiện
MinValue	Giá trị nhỏ nhất phải nhập (thường dùng cho textbox số liệu) (có thể không nhập)
.MaxValue	Giá trị lớn nhất phải nhập (thường dùng cho textbox số liệu) (có thể không nhập)

DefaultValue	Giá trị mặc định của textbox
Visible	Ân hiện của textbox (1: Hiện, 0: Ân)
ColumnName	Tham chiếu cột trong bảng db Khách Hàng
sysRegularID	Tham chiếu tới bảng regular (Dùng khi textbox có sử dụng ragular)
sysTypeID	Tham chiếu tới bảng sysType (lấy kiểu dữ liệu của textbox)
GridVisible	Hiện trên lưới (1: có, 0: không)
FieldVisible	Hiện trên Màn hình (1: có, 0: không)
UniqueField	Khóa chính (1: là khóa chính, 0: không là khóa chính)
RefTable	Bảng mà fields(textbox) tham chiếu tới trong db khách hàng(có thể không nhập)
GroupID	(Giải thích ở màn hình viewmasterdetail)

- Sử dụng script

```

declare @sysTable nvarchar(50)
declare @AllowNull int
declare @sysComboBoxID int
declare @Type int
declare @TabIndex int
declare @MinValue int
declare @MaxValue int
declare @DefaultValue nvarchar(100)
declare @Visible int
declare @ColumnName varchar(50)
declare @sysRegularID int
declare @sysTypeID int
declare @GridVisible int
declare @FieldVisible int
declare @UniqueField int
declare @RefTable nvarchar(50)
declare @GroupID int
declare @sysEditorTemplateID int
declare @sysClientTemplateID int
declare @ReadOnly int
declare @TabIndexGrid int
declare @TabIndexPopup int
declare @TabIndexView int
declare @PopupVisible int
declare @ViewVisible int
declare @MaxLength int
declare @SpecialControl int
declare @RefDisplayField varchar(50)
declare @RefValueField varchar(50)
declare @RefViewTable varchar(50)
declare @FieldID varchar(50)
declare @GridWidth int

set @sysComboBoxID=(select top 1 sysComboBoxID from
sysComboBox where ComboBoxID = 'CB_0004')
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @GroupID=(select(select top
convert(varchar(50),sysGroupID) from sysGroup where
GroupID='SOF2002.ThongTinDonHang'))
```

```

set @sysTable=N'OT2001'
set @AllowNull=N'0'
set @ReadOnly=N'0'
set @Type=N'4'
set @TabIndex=N'2'
set @TabIndexGrid=N'1'
set @TabIndexPopup=N'1'
set @TabIndexView=N'1'
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'DivisionID'
set @sysDataTypeID=N'7'
set @GridVisible=N'1'
set @FieldVisible=N'1'
set @PopupVisible=N'1'
set @ViewVisible=N'1'
set @UniqueField=N'0'
set @RefTable=null
set @MaxLenght=null
set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'Field0013'
set @GridWidth=N'100'
If not exists(select top 1 1 from [dbo].[sysFields] where
[FieldID] = N'Field0013')Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,Ta
bIndex,TabIndexGrid,TabIndexPopup,TabIndexView,MinValue,Max
Value,DefaultValue,Visible,ColumnName,sysRegularID,sysDataT
ypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,Uni
queField,RefTable,GroupID,sysEditorTemplateID,MaxLenght,Spe
cialControl,RefDisplayField,RefValueField,RefViewTable,sysC
lientTemplateID,FieldID,GridWidth)values(@sysTable,@AllowNu
ll,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@
TabIndexPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultVal
ue,@Visible,@ColumnName,@sysRegularID,@sysDataTypeID,@GridV
isible,@FieldVisible,@PopupVisible,@ViewVisible,@UniqueFiel
d,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLenght,@Speci
alControl,@RefDisplayField,@RefValueField,@RefViewTable,@sy
sClientTemplateID,@FieldID,@GridWidth)
End

```

2) CheckBox

THÊM/SỬA DANH MỤC SẢN PHẨM

MaSanPham

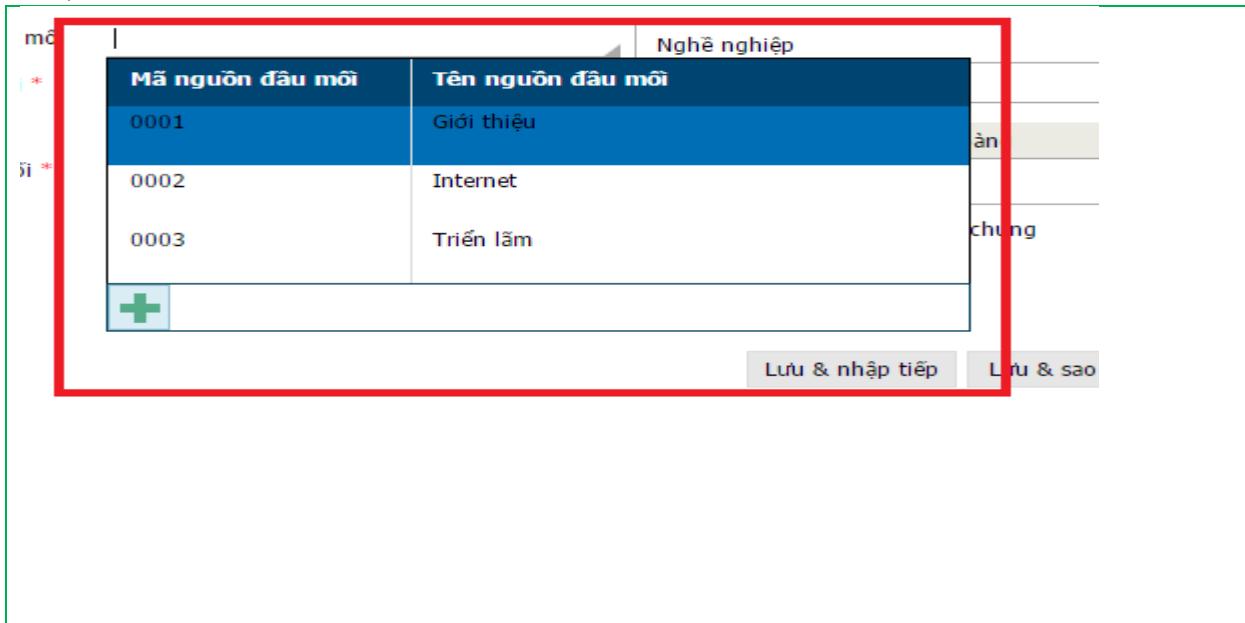
DM001.Disable

Để tạo 1 checkbox ta vào bảng sysfield thêm 1 dòng dữ liệu vào:

sysFields	
sysFieldID	Id của Textbox (tự sinh động)
sysTable	Chextbox thuộc bảng nào
AllowNull	Không cần nhập
sysComboBoxID	Nhập mã combobox, mặc định checkbox sẽ chuyển thành combobox trong màn hình contentmaster.
Type	2 (Checkbox)
TabIndex	Thứ tự xuất hiện
MinValue	Không cần nhập
MaxValue	Không cần nhập
DefaultValue	Không cần nhập
Visible	Ẩn hiện của checkbox (1: Hiện, 0: Ẩn)
ColumnName	Tham chiếu cột trong bảng db Khách Hàng
sysRegularID	Tham chiếu tới bảng regular (Dùng khi textbox có sử dụng regular)
sysDataTypeID	Tham chiếu tới bảng sysDataType (lấy kiểu dữ liệu của checkbox)
GridVisible	Hiện trên lưới (1: có, 0: không)
FieldVisible	Hiện trên Màn hình (1: có, 0: không)
UniqueField	Khóa chính (1: là khóa chính, 0: không là khóa chính)
RefTable	Bảng mà fields(checkbox) tham chiếu tới trong db khách hàng (có thể không nhập)
GroupID	(Giải thích ở màn hình viewmasterdetail)

- Sử dụng script tương tự textbox.
- Ở màn hình danh mục content master checkbox sẽ tự chuyển thành combobox yes/no

3) ComboBox



Mã nguồn đầu môi	Tên nguồn đầu môi
0001	Giới thiệu
0002	Internet
0003	Triển lãm

a) Tạo một combobox nếu chưa tồn tại.

sysComboBox	
sysComboBoxID	Khóa chính tự tăng
DisplayMember	Tên cột hiển thị
ValueMember	Tên cột giá trị
StoreName	Tên Store để lấy dữ liệu
StoreParameter	Các Parameter của Store
SQLQuery	Câu truy vấn SQL(Lưu ý không đưa điều kiện lọc)
VViewTable	Các Field được sử dụng trong combobox
sysTableID	Bảng mà dữ liệu lấy ra. VD: AT1103: có chức năng thêm nhanh Null : không có chức năng thêm nhanh. Dữ liệu lấy từ 2 bảng trở lên sẽ không hỗ trợ được thêm nhanh.

- StoreParameter có dạng: @Para1=@Column → sử dụng ColumnName của Field trên màn hình filter

@Para2=@DivisionID → sử dụng biến môi trường

Các parameter cách nhau bằng dấu phẩy

Ví dụ: @Para1=@Column, @Para2=@DivisionID

- SQLQuery có dạng: select * from table key1 =@Colum1 and key2=@DivisionID or key3=@UserName
Trong đó: @Column1 → Biến trên control
@DivisionID → Biến môi trường
- Sử dụng script

```

declare @Displaymember varchar(50)
declare @Valuemember varchar(50)
declare @Storename varchar(50)
declare @StoreParameter varchar(50)
declare @SQLQuery varchar(50)
declare @VViewTable varchar(50)
declare @ComboBoxID varchar(50)

set @DisplayMember=N'EmployeeName'
set @ValueMember= N'EmployeeID'
set @StoreName=null
set @StoreParameter=null
set @SQLQuery= N'Select EmployeeID, FullName as EmployeeName from AT1103'
set @VViewTable=N'VViewEmployeeID'
set @ComboBoxID=N'CB_0001'
set @sysTableID =N'AT1103'

If not exists(select top 1 1 from [dbo].[sysComboBox] where [ComboBoxID] =
N'CB_0001')Begin
insert into
sysComboBox(DisplayMember,ValueMember,StoreName,StoreParameter,SQLQuery,VViewTab
le,ComboBoxID)values(@DisplayMember,@ValueMember,@StoreName,@StoreParameter,@SQL
Query,@VViewTable,@ComboBoxID)
End

```

MEMBERS OF GREENSUN GROUP







19

b) Tạo các field(cột) được hiển thị trong combobox(Sử dụng bảng sysField)

sysFields	
sysFieldID	Khóa chính tự tăng
sysTable	Tham chiếu đến ViewTable của sysComboBox để xác định Field này sử dụng trong ComboBox chứ không sử dụng trên màn hình
Type	Sử dụng loại 1: TextBox
TabIndex	Thứ tự cột
ColumnName	Tên cột tham chiếu trong câu truy vấn hoặc store
Visible	1
FieldID	Mã của Field

- Sử dụng script

```

declare @sysTable varchar(50)
declare @type int
declare @TabIndex int
declare @ColumnName varchar(50)
declare @AllowNull int
declare @Visible int
declare @FieldID varchar(50)

set @Visible = 1
set @AllowNull = 1
set @sysTable = ''
set @type = 1
set @TabIndex = 0
set @ColumnName = ''
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] =
N'Field0001')Begin
insert into sysFields(sysTable, Type, AllowNull, Visible, TabIndex, ColumnName,
FieldID)
values (@sysTable, @type, @AllowNull, @Visible, @TabIndex, @ColumnName,
@FieldID)
End

```

c) Tạo Field sử dụng ComboBox trên màn hình

sysFields	
sysFieldID	Khóa chính tự tăng
sysComboBoxID	Mã tham chiếu tới sysComboBox
sysTable	Bảng được sử dụng trong DB khách hàng
Type	Loại 3: ComboBox
TabIndex	Thứ tự cột
ColumnName	Tên cột tham chiếu DB khách hàng
Visible	1
FieldID	Mã của Field

- Sử dụng script tương tự textbox

```

declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)

```

```

declare @ReadOnly nvarchar(MAX)
declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysTypeID nvarchar(MAX)
declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)
declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
declare @MaxLenght nvarchar(MAX)
declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)
declare @RefTableComboboxID nvarchar(MAX)

```

```

set @sysComboBoxID=(select top 1 sysComboBoxID from
sysComboBox where ComboBoxID = 'CB_0027')
set @sysRegularID=null

```

```

set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @sysRadioButtonID=null
set @GroupID=(select(select top 1 convert(varchar(50),sysGroupID)
from sysGroup where GroupID='SOF2002.ThongTinDonHang'))
set @sysTable=N'OT2001'
set @AllowNull=N'1'
set @ReadOnly=N'0'
set @Type=N'3'
set @TabIndex=null
set @TabIndexGrid=null
set @TabIndexPopup=N'20'
set @TabIndexView=N'25'
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'PaymentTermID'
set @sysTypeID=N'7'
set @GridVisible=N'0'
set @FieldVisible=N'0'
set @PopupVisible=N'1'
set @ViewVisible=N'1'
set @UniqueField=N'0'
set @RefTable=null
set @MaxLenght=null
set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'Field0208'
set @GridWidth=N'100'
set @LanguageID=null
set @IsImport=null
set @GroupOnGrid=null
set @SumOnGrid=null
set @AverageOnGrid=null
set @MaxOnGrid=null
set @MinOnGrid=null
set @RefTableComboboxID=null
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] =
N'Field0208')Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex

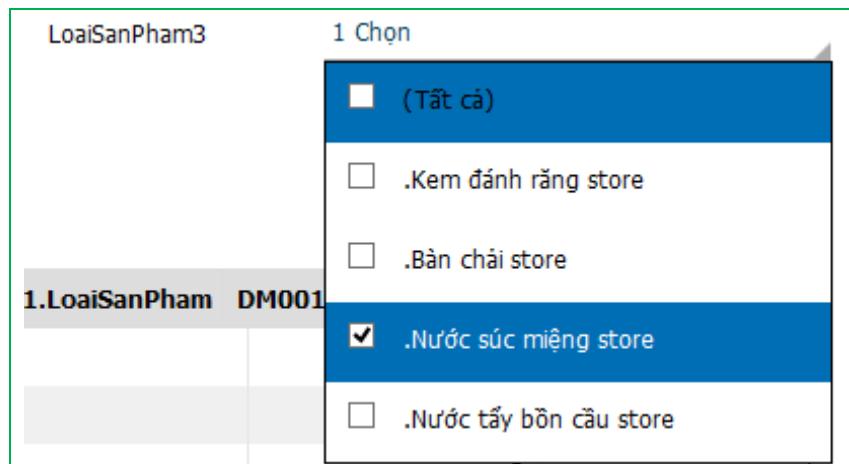
```

```
,TabIndexGrid,TabIndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visible,ColumnName,sysRegularID,sysTypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,UniqueField,RefTable,GroupID,sysEditorTemplateID,MaxLength,SpecialControl,RefDisplayField,RefValueField,RefViewTable,sysClientTemplateID,FieldID,GridWidth,sysRadioButtonID,LanguageID,IsImport,GroupOnGrid,SumOnGrid,AverageOnGrid,MaxOnGrid,MinOnGrid,RefTableComboboxID)values(@sysTable,@AllowNull,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabIndexPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@ColumnName,@sysRegularID,@sysTypeID,@GridVisible,@FieldVisible,@PopupVisible,@ViewVisible,@UniqueField,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLength,@SpecialControl,@RefDisplayField,@RefValueField,@RefViewTable,@sysClientTemplateID,@FieldID,@GridWidth,@sysRadioButtonID,@LanguageID,@IsImport,@GroupOnGrid,@SumOnGrid,@AverageOnGrid,@MaxOnGrid,@MinOnGrid,@RefTableComboboxID)
End
```

d) Khai báo ngôn ngữ cho column trên combobox

- Key ngôn ngữ mặc định hiển thị trên combobox có dạng: {ScreenID}.{Column}.CB
 - ScreenID: Mã màn hình đang thao tác
 - Column: tên cột của Field
 - CB: khóa cố định (Luôn luôn phải có)

4) ComboCheckList



- Tạo Field trên màn hình và sử dụng như ComboBox với Type là 4, sysTypeID là 7

- CheckList chỉ hiển thị DisplayMember và xử lý ValueMember đối với Store hoặc SQL trả về kết quả có nhiều cột
- Đối với checklist tạo mới không cần có ViewTable như ComboBox

Sử dụng script tương tự checkbox

```

declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)
declare @ReadOnly nvarchar(MAX)
declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysDataTypeID nvarchar(MAX)
declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)
declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
declare @MaxLenght nvarchar(MAX)
declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)
declare @RefTableComboboxID nvarchar(MAX)

```

```

set @sysComboBoxID=(select top 1 sysComboBoxID from sysComboBox where
ComboBoxID = 'CB_0004')
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @sysRadioButtonID=null
set @GroupID=(select(select top 1 convert(varchar(50),sysGroupID) from sysGroup where
GroupID='SOF2002.ThongTinDonHang'))
set @sysTable=N'OT2001'
set @AllowNull=N'0'
set @ReadOnly=N'0'
set @Type=N'4'
set @TabIndex=N'2'
set @TabIndexGrid=N'1'
set @TabIndexPopup=N'1'
set @TabIndexView=N'1'
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'DivisionID'
set @sysTypeID=N'7'
set @GridVisible=N'1'
set @FieldVisible=N'1'
set @PopupVisible=N'1'
set @ViewVisible=N'1'
set @UniqueField=N'1'
set @RefTable=null
set @MaxLenght=null
set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'Field0013'
set @GridWidth=N'100'
set @LanguageID=null
set @IsImport=null
set @GroupOnGrid=null
set @SumOnGrid=null
set @AverageOnGrid=null
set @MaxOnGrid=null
set @MinOnGrid=null
set @RefTableComboboxID=null
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] = N'Field0013')Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex,TabIndexGrid,TabIndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visible,ColumnName,sysRegul
arID,sysTypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,UniqueField,RefTab
le,GroupID,sysEditorTemplateID,MaxLenght,SpecialControl,RefDisplayField,RefValueField,

```

```
RefViewTable,sysClientTemplateID,FieldID,GridWidth,sysRadioButtonID,LanguageID,IsImport,GroupOnGrid,SumOnGrid,AverageOnGrid,MaxOnGrid,MinOnGrid,RefTableComboboxID)values(@sysTable,@AllowNull,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabIndexPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@ColumnName,@sysRegularID,@sysTypeID,@GridVisible,@FieldVisible,@PopuVisible,@ViewVisible,@UniqueField,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLength,@SpecialControl,@RefDisplayField,@RefValueField,@RefViewTable,@sysClientTemplateID,@FieldID,@GridWidth,@sysRadioButtonID,@LanguageID,@IsImport,@GroupOnGrid,@SumOnGrid,@AverageOnGrid,@MaxOnGrid,@MinOnGrid,@RefTableComboboxID)
End
```

5) DatePicker

mục	Nghiệp vụ	Báo cáo	Trợ giúp																																																								
HÀNG - POSF0027																																																											
Đến ngày	30/11/2015		Mã cửa hàng																																																								
Đến kỳ	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="7" style="text-align: center;">November 2015</th> </tr> <tr> <th>Su</th> <th>Mo</th> <th>Tu</th> <th>We</th> <th>Th</th> <th>Fr</th> <th>Sa</th> </tr> </thead> <tbody> <tr> <td>25</td> <td>26</td> <td>27</td> <td>28</td> <td>29</td> <td>30</td> <td>31</td> </tr> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td>8</td> <td>9</td> <td>10</td> <td>11</td> <td>12</td> <td>13</td> <td>14</td> </tr> <tr> <td>15</td> <td>16</td> <td>17</td> <td>18</td> <td>19</td> <td>20</td> <td>21</td> </tr> <tr> <td>22</td> <td>23</td> <td>24</td> <td>25</td> <td>26</td> <td>27</td> <td>28</td> </tr> <tr> <td>29</td> <td>30</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </tbody> </table>			November 2015							Su	Mo	Tu	We	Th	Fr	Sa	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5
November 2015																																																											
Su	Mo	Tu	We	Th	Fr	Sa																																																					
25	26	27	28	29	30	31																																																					
1	2	3	4	5	6	7																																																					
8	9	10	11	12	13	14																																																					
15	16	17	18	19	20	21																																																					
22	23	24	25	26	27	28																																																					
29	30	1	2	3	4	5																																																					
Mã kho xuất	Monday, November 30, 2015																																																										

- Tạo một Fields có Type là 5
- Các giá trị còn lại tạo tương tự textbox

```
declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)
```

```

declare @ReadOnly nvarchar(MAX)
declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysDataTypeID nvarchar(MAX)
declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)
declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
declare @MaxLenght nvarchar(MAX)
declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)

set @sysComboBoxID=null
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @sysRadioButtonID=null
set @GroupID=null
set @sysTable=N'DM001'
set @AllowNull=N'1'
set @ReadOnly=null

```

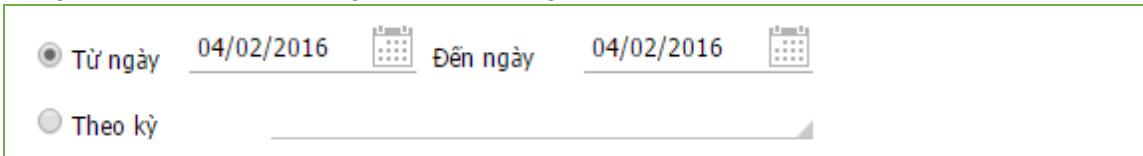
```

set @Type=N'5'
set @TabIndex=N'12'
set @TabIndexGrid=null
set @TabIndexPopup=null
set @TabIndexView=null
set @MinValue=null
set @MaxValue=null
set @DefaultValue=N'1'
set @Visible=N'1'
set @ColumnName=N'VoucherDate'
set @sysTypeID=N'13'
set @GridVisible=N'1'
set @FieldVisible=N'1'
set @PopupVisible=N'1'
set @ViewVisible=N'1'
set @UniqueField=N'0'
set @RefTable=null
set @MaxLenght=null
set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'Field0005'
set @GridWidth=N'100'
set @LanguageID=null
set @IsImport=null
set @GroupOnGrid=null
set @SumOnGrid=null
set @AverageOnGrid=null
set @MaxOnGrid=null
set @MinOnGrid=null
set @RefTableComboboxID=null
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] = N'Field0005')Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex,TabIndexGrid,TabIndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visible,ColumnName,sysRegularID,sysTypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,UniqueField,RefTable,GroupID,sysEditorTemplateID,MaxLenght,SpecialControl,RefDisplayField,RefValueField,RefViewTable,sysClientTemplateID,FieldID,GridWidth,sysRadioButtonID,LanguageID,IsImport,GroupOnGrid,SumOnGrid,AverageOnGrid,MaxOnGrid,MinOnGrid,RefTableComboboxID)values(@sysTable,@AllowNull,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabIndexPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@ColumnName,@sysRegularID,@sysTypeID,@GridVisible,@FieldVisible,@PopupVisible,@ViewVisible,@UniqueField,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLenght,@SpecialControl,@RefDisplayField,@RefValueField,@RefViewTable,@sysClientTemplateID,@FieldID,@GridWidth,@sysRadioButtonID,@LanguageID,@IsImport,@GroupOnGrid,@SumOnGrid,@AverageOnGrid,@MaxOnGrid,@MinOnGrid,@RefTableComboboxID)
End

```

6) SpecialControl

- Dùng để tạo các control không theo bất kỳ dạng chuẩn nào



a) Cấu trúc

- Khi định nghĩa một Field có sử dụng SpecialControl, hệ thống sẽ không tạo control chuẩn mà sẽ parse giao diện từ PartialName lưu trong View/Share/Control
- Khi SpecialControl sử dụng truy vấn để lọc dữ liệu thì sẽ lưu câu truy vấn trong bảng ReturnResult

b) Khởi tạo

- Định nghĩa SpecialControl trong bảng sysSpecialControl

sysSpecialControl	
ReturnResultID	Tham chiếu tới bảng truy vấn theo điều kiện của SpecialControl
PartialName	Tên PartialName sẽ được sử dụng
SpecialControlID	Mã SpecialControl(tự đặt)

- Định nghĩa các câu truy vấn trả về trong bảng ReturnResult

sysReturnResult	
ReturnResultID	Mã của câu truy vấn trả về
ReturnResult	Câu truy vấn trả về
Condition	Điều kiện để sử dụng câu truy vấn trả về

c) Ví dụ

- Trong bảng SpecialControl ta định nghĩa

	sysSpecialControlID	ReturnResultID	PartialName	SpecialControlID
	1	1	PeriodChecklist	PeriodChecklist
/*	NULL	NULL	NULL	NULL

- Trong bảng ReturnResult ta định nghĩa

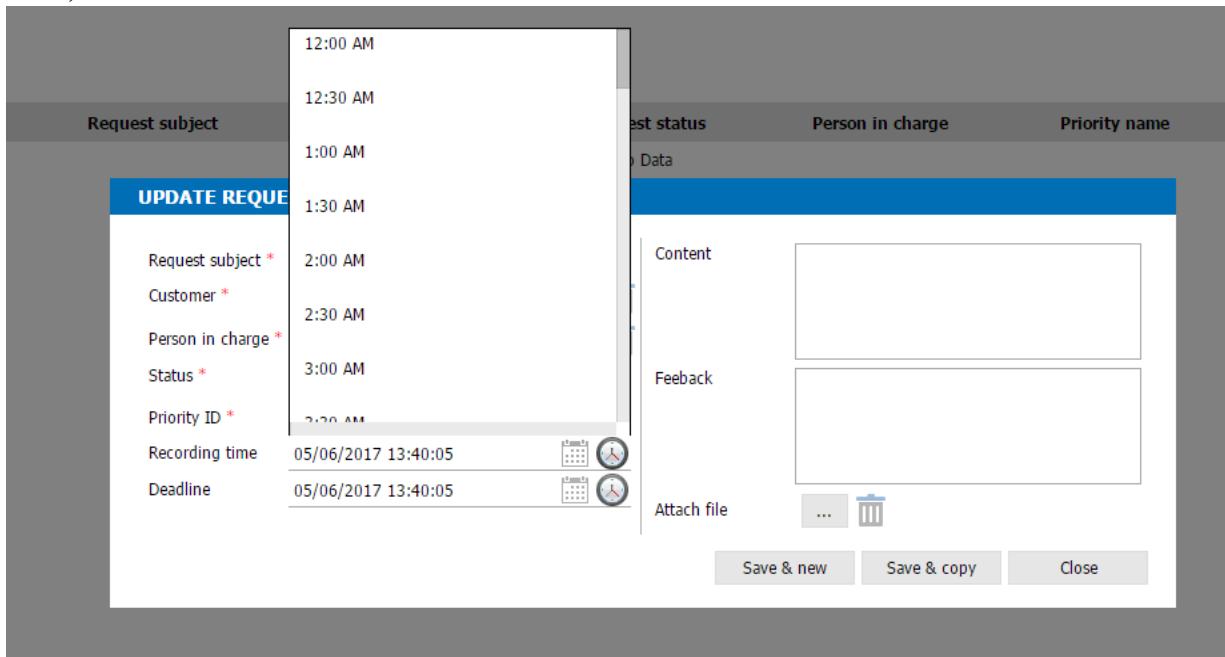
sysReturnResultID	ReturnResultID	ReturnResult	Condition
2	1	(CONVERT(VARCHAR(10),OT2001.OrderDate,112) BETWEEN CONVERT(VARCHAR(10),@FromDatePeriodControl ,112)... @IsPeriod=0	
3	1	((CASE WHEN OT2001.TranMonth <10 THEN '0'+rtrim(ltrim(str(OT2001.TranMonth)))+'/'+ltrim(Rtrim(str(OT2001.Tra... @IsPeriod=1	
NULL	NULL	NULL	NULL

Lúc này khi Field sử dụng SpecialControl là PeriodChecklist thì khi lọc câu truy vấn trả về sẽ phụ thuộc vào biến IsPeriod

- Nếu biến IsPeriod=0 sẽ trả về câu truy vấn thứ nhất có sysReturnResult là 2

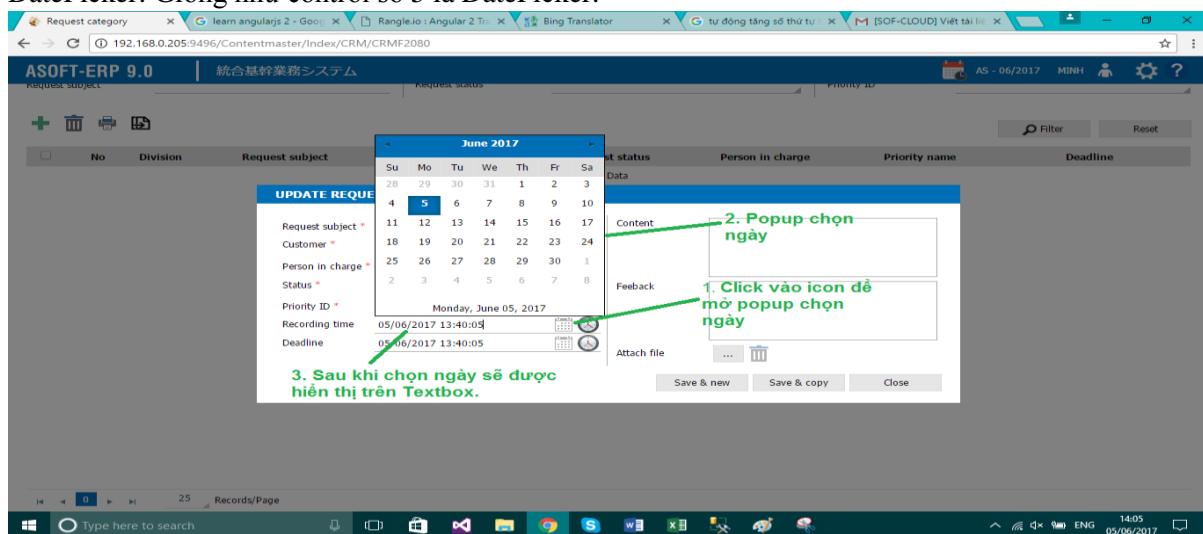
- Nếu biến IsPeriod=1 sẽ trả về câu truy vấn thứ hai có sysReturnResult là 3
- Câu truy vấn trên sẽ được thêm vào điều kiện where khi lọc dữ liệu
- Câu truy vấn sử dụng giống câu truy vấn của combobox
- Trong đó @ColumnName → sử dụng biến trên màn hình
 @@DivisionID → sử dụng biến môi trường.

7) DateTimePicker

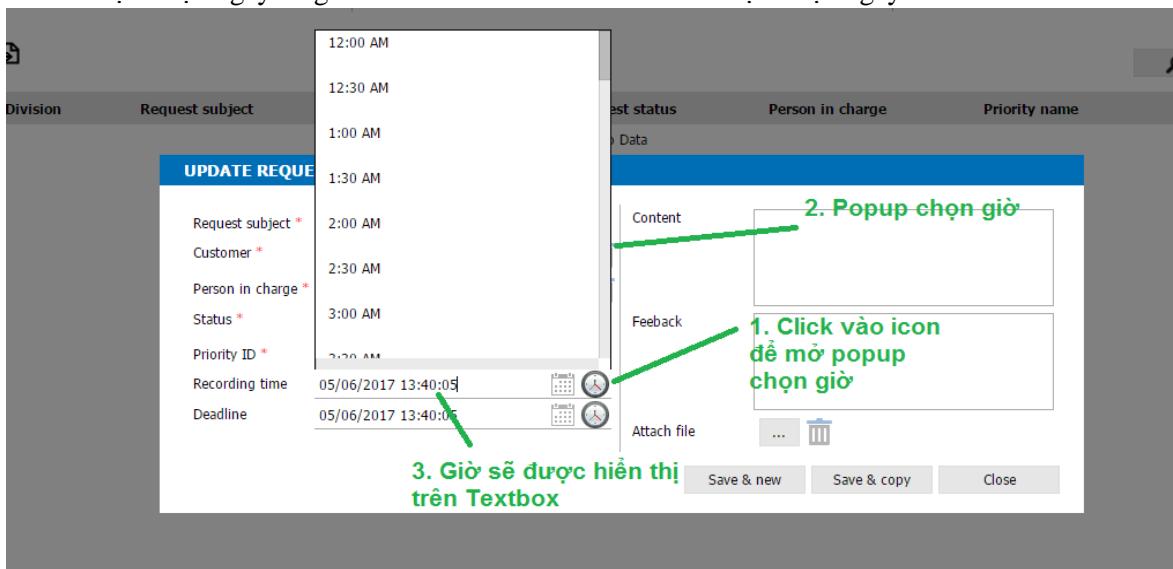


a) Cấu trúc: Gồm 2 phần là DatePicker và TimePicker

- DatePicker: Giống như control số 5 là DatePicker:



- TimePicker: Dùng để chọn giờ, đây là điểm khác với control số 5 là DatePicker, DateTimePicker có thể chọn được ngày và giờ chi tiết. DatePicker chỉ có thể chọn được ngày:



b) Cách tạo

- Tạo một Fields có Type là 7, sysTypeID là 13;
- Các giá trị còn lại tạo tương tự textbox

```

declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)
declare @ReadOnly nvarchar(MAX)
declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysTypeID nvarchar(MAX)
declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)

```

```

declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
declare @MaxLenght nvarchar(MAX)
declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)

set @sysComboBoxID=null
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @sysRadioButtonID=null
set @GroupID=(select(select top 1 convert(varchar(50),sysGroupID) from sysGroup where
GroupID='SOF2002.ThongTinDonHang'))
set @sysTable=N'OT2001'
set @AllowNull=N'1'
set @ReadOnly=N'0'
set @Type=N'7'
set @TabIndex=null
set @TabIndexGrid=N'14'
set @TabIndexPopup=N'12'
set @TabIndexView=N'12'
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'ShipDate'
set @sysTypeID=N'13'
set @GridVisible=N'1'
set @FieldVisible=N'0'
set @PopupVisible=N'1'
set @ViewVisible=N'1'
set @UniqueField=N'0'
set @RefTable=null
set @MaxLenght=null
set @SpecialControl=null
set @RefDisplayField=null

```

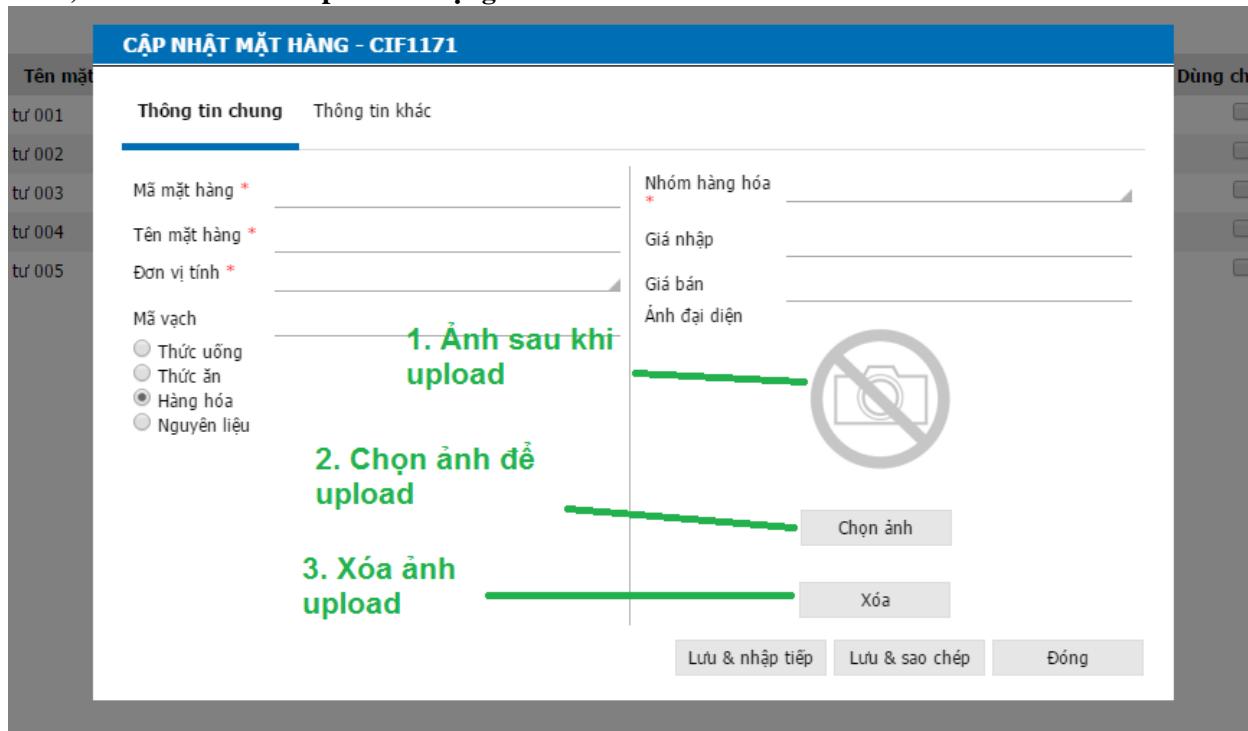
```

set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'Field0019'
set @GridWidth=N'100'
set @LanguageID=null
set @IsImport=null
set @GroupOnGrid=null
set @SumOnGrid=null
set @AverageOnGrid=null
set @MaxOnGrid=null
set @MinOnGrid=null
set @RefTableComboboxID=null
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] = N'Field0019')Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex,TabIndexGrid,TabIndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visible,ColumnName,sysRegularID,sysTypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,UniqueField,RefTable,GroupID,sysEditorTemplateID,MaxLength,SpecialControl,RefDisplayField,RefValueField,RefViewTable,sysClientTemplateID,FieldID,GridWidth,sysRadioButtonID,LanguageID,IsImport,GroupOnGrid,SumOnGrid,AverageOnGrid,MaxOnGrid,MinOnGrid,RefTableComboboxID)values(@sysTable,@AllowNull,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabIndexPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@ColumnName,@sysRegularID,@sysTypeID,@GridVisible,@FieldVisible,@PopupVisible,@ViewVisible,@UniqueField,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLength,@SpecialControl,@RefDisplayField,@RefValueField,@RefViewTable,@sysClientTemplateID,@FieldID,@GridWidth,@sysRadioButtonID,@LanguageID,@IsImport,@GroupOnGrid,@SumOnGrid,@AverageOnGrid,@MaxOnGrid,@MinOnGrid,@RefTableComboboxID)
End

```

8) Upload

a) Mô tả: Control Upload có dạng như sau:



b) Cách tạo

- Tạo một Fields có Type là 11, sysTypeID là 14;
- Các giá trị còn lại tạo tương tự textbox

```
declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)
declare @ReadOnly nvarchar(MAX)
declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysTypeID nvarchar(MAX)
declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
```

```

declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)
declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
declare @MaxLenght nvarchar(MAX)
declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)

set @sysComboBoxID=null
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @GroupID=(select(select top 1 convert(varchar(50),sysGroupID) from sysGroup where
GroupID='CIF1121.ThongTinDonVi')+','+(select top 1 convert(varchar(50),sysGroupID) from
sysGroup where GroupID='CIF1122.ThongTinDonVi'))
set @sysTable=N'AT1101'
set @AllowNull=N'1'
set @ReadOnly=N'0'
set @Type=N'11'
set @TabIndex=null
set @TabIndexGrid=null
set @TabIndexPopup=N'13'
set @TabIndexView=null
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'Logo'
set @sysTypeID=N'14'
set @GridVisible=N'0'
set @FieldVisible=N'0'
set @PopupVisible=N'1'
set @ViewVisible=N'0'
set @UniqueField=N'0'
set @RefTable=null
set @MaxLenght=N'100'

```

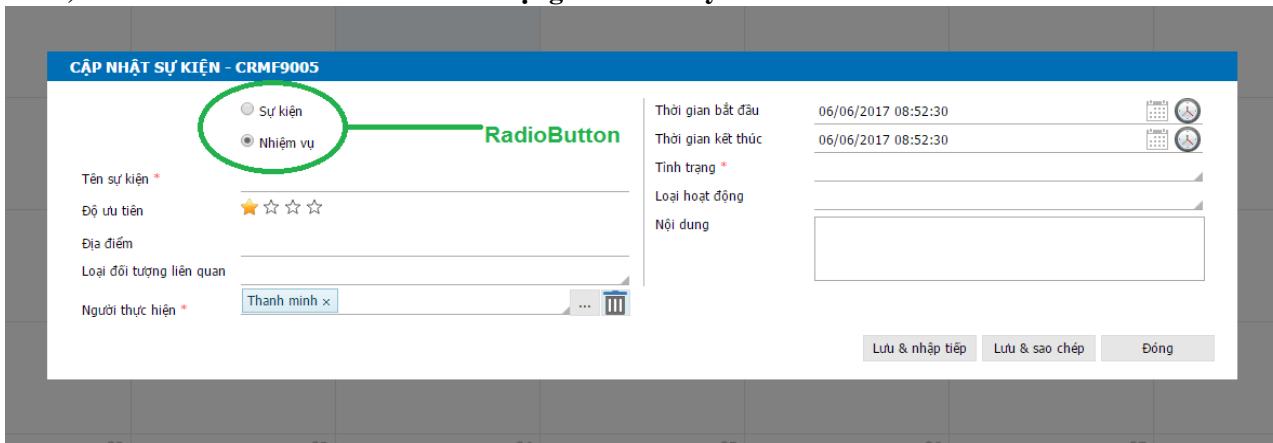
```

set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'FieldLogoAT1101'
set @GridWidth=N'100'
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] =
N'FieldLogoAT1101')Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex,TabIndexGrid,TabIndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visible,ColumnName,sysRegularID,sysDataTypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,UniqueField,RefTable,GroupID,sysEditorTemplateID,MaxLength,SpecialControl,RefDisplayField,RefValueField,RefViewTable,sysClientTemplateID,FieldID,GridWidth)values(@sysTable,@AllowNull,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabIndexPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@ColumnName,@sysRegularID,@sysDataTypeID,@GridVisible,@FieldVisible,@PopupVisible,@ViewVisible,@UniqueField,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLength,@SpecialControl,@RefDisplayField,@RefValueField,@RefViewTable,@sysClientTemplateID,@FieldID,@GridWidth)
End

```

9) RadioButton

a) Mô tả: Control RadioButton có dạng như thế này



b) Cách tạo

- Radiobutton là một danh sách những lựa chọn mà chúng ta chỉ được chọn duy nhất một lựa chọn trong danh sách đó. Chú ý danh sách là số nhiều nên radiobutton chỉ được tạo khi có ít nhất 2 lựa chọn.
- **VD:** Nhìn lên hình trên, chúng ta có 2 lựa chọn trong radiobutton là “**Sự kiện**” và “**Nhiệm vụ**”. Ta chỉ được chọn duy nhất 1 lựa chọn trong số chúng.
- Và để tạo được những danh sách này chúng ta làm như sau:

Bước 1: Vào Database Asoft Admin => Table [dbo].[RadioButton]. Tại đây chúng ta sẽ tạo mới một danh sách những lựa chọn cho radiobutton.

	sysRadioButtonID	StoreName	StoreParameter	SQLQuery	RadioButtonID
	1	NULL	NULL	Select 'CIF1171.MethodID1' as IDLang...	Rdo_MethodID
	2	NULL	NULL	Select 'CIF1175.SelectALLMethod' as ...	Rdo_ALLMethodID
	1002	NULL	NULL	Select N'DMY' as Text, '0' as Value un...	Rdo_OrderFormat
	1003	NULL	NULL	SELECT InventoryTypeID as Value, Inv...	Rdo_InventoryTypeID
▶	1004	NULL	NULL	Select N'CRMF9005.Type1' as IDLangua...	Rdo_TypeIDCRMF9005
*	NULL	NULL	NULL	NULL	NULL

Nhìn hình trên ta thấy [dbo].[RadioButton] có 5 cột.

- **Column[“sysRadioButtonID”]**: có dạng Int identity. Cột này không cần phải khai báo, cứ để trống.
- **Column[“StoreName”]**: có dạng varchar. Dùng để khai báo tên của StoreProcedure. StoreProcedure này dùng để load lên danh sách của radiobutton.
- **Column[“StoreParameter”]**: là tham số của StoreProcedure nếu có. Mỗi tham số cách nhau bằng dấu phẩy.
- **Column[“SQLQuery”]**: Đây cũng là câu load lên danh sách của radiobutton. Đến đây ta sẽ thấy **Column[“StoreName”]** và **Column[“StoreParameter”]** cũng dùng để load lên danh sách của radiobutton. Về cơ bản **Column[“StoreName”]** và **Column[“SQLQuery”]** đều có chức năng giống nhau đều là load danh sách của radiobutton. Nhưng vì để có nhiều sự lựa chọn nên đã tạo ra 2 cột này. Lưu ý là nếu dùng **Column[“SQLQuery”]** thì **Column[“StoreName”]** và **Column[“StoreParameter”]** ta để NULL giống hình trên và ngược lại.
- **Column[“RadioButtonID”]**: là cột để nhập id của dòng chúng ta khai báo. Dùng để cho những table khác select ra.

Bước 2: Vào [dbo].[sysFields] khi thêm một dòng mới chú ý set:

- Tạo một Fields có Type là 9, sysDataTypeID tùy vào kiểu dữ liệu của field
- Các giá trị còn lại tạo tương tự textbox

```

declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)
declare @ReadOnly nvarchar(MAX)
declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysTypeID nvarchar(MAX)

```

```

declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)
declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
declare @MaxLenght nvarchar(MAX)
declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)

set @sysComboBoxID=null
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @sysRadioButtonID=(select top 1 sysRadioButtonID from sysRadioButton where
RadioButtonID = 'Rdo_OrderFormat')
set @GroupID=(select(select top 1 convert(varchar(50),sysGroupID) from sysGroup where
GroupID='CIF0040.KieuSoNgay'))
set @sysTable=N'AT14051'
set @AllowNull=N'1'
set @ReadOnly=N'0'
set @Type=N'9'
set @TabIndex=null
set @TabIndexGrid=null
set @TabIndexPopup=N'52'
set @TabIndexView=null
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'OrderFormat'
set @sysTypeID=N'7'
set @GridVisible=N'0'
set @FieldVisible=N'0'

```

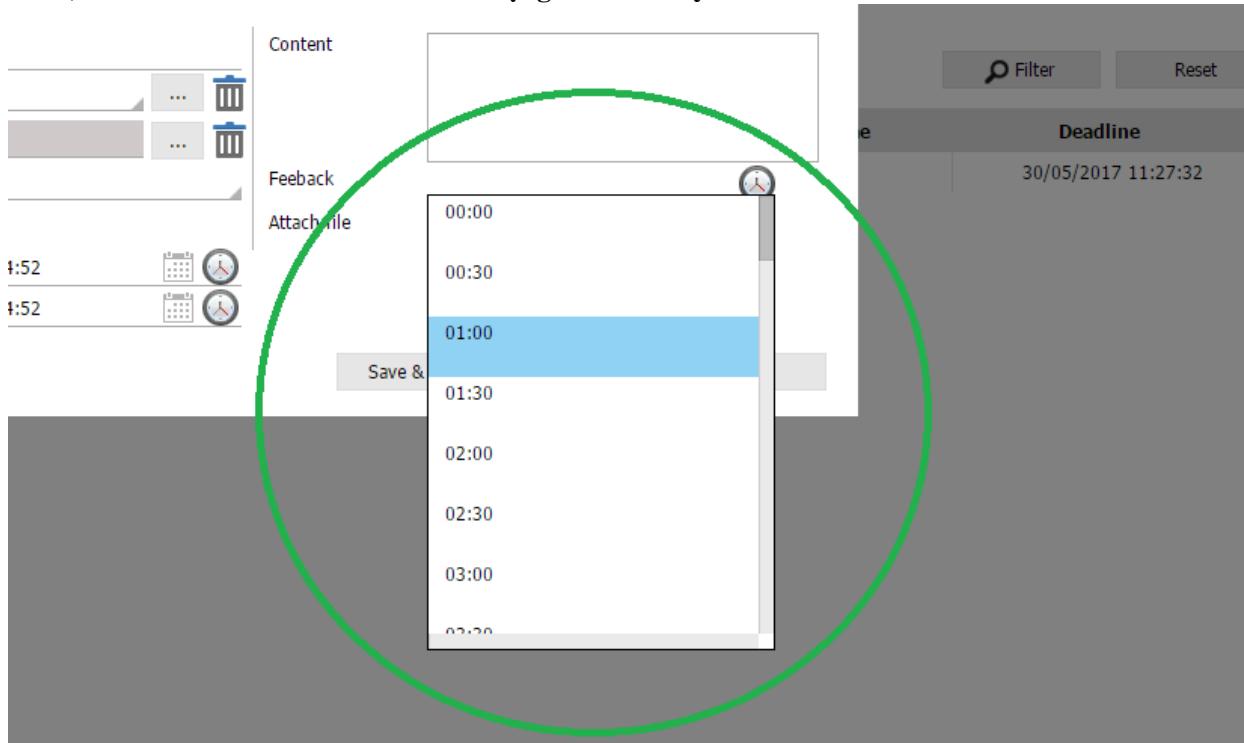
```

set @PopupVisible=N'1'
set @ViewVisible=N'0'
set @UniqueField=N'0'
set @RefTable=null
set @MaxLenght=null
set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'FieldOrderFormatAT14051'
set @GridWidth=N'100'
set @LanguageID=null
set @IsImport=null
set @GroupOnGrid=null
set @SumOnGrid=null
set @AverageOnGrid=null
set @MaxOnGrid=null
set @MinOnGrid=null
set @RefTableComboboxID=null
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] =
N'FieldOrderFormatAT14051')Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex,TabIndexGrid,TabIndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visible,ColumnName,sysRegularID,sysTypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,UniqueField,RefTable,GroupID,sysEditorTemplateID,MaxLenght,SpecialControl,RefDisplayField,RefValueField,RefViewTable,sysClientTemplateID,FieldID,GridWidth,sysRadioButtonID,LanguageID,IsImport,GroupOnGrid,SumOnGrid,AverageOnGrid,MaxOnGrid,MinOnGrid,RefTableComboboxID)values(@sysTable,@AllowNull,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabIndexPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@ColumnName,@sysRegularID,@sysTypeID,@GridVisible,@FieldVisible,@PopupVisible,@ViewVisible,@UniqueField,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLenght,@SpecialControl,@RefDisplayField,@RefValueField,@RefViewTable,@sysClientTemplateID,@FieldID,@GridWidth,@sysRadioButtonID,@LanguageID,@IsImport,@GroupOnGrid,@SumOnGrid,@AverageOnGrid,@MaxOnGrid,@MinOnGrid,@RefTableComboboxID)
End

```

10) TimePicker

a) Mô tả: Control TimePicker có dạng như thế này



- TimePicker là control cho người dùng chọn giờ. Trong DateTimePicker đã có nhắc tới.

b) Cách tạo

- Tạo một Fields có Type là 11, sysTypeID là 16
- Các giá trị còn lại tạo tương tự textbox

```
declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)
declare @ReadOnly nvarchar(MAX)
declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysTypeID nvarchar(MAX)
```

```

declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)
declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
declare @MaxLenght nvarchar(MAX)
declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)

set @sysComboBoxID=null
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @sysRadioButtonID=(select top 1 sysRadioButtonID from sysRadioButton where
RadioButtonID = 'Rdo_OrderFormat')
set @GroupID=(select(select top 1 convert(varchar(50),sysGroupID) from sysGroup where
GroupID='CIF0040.KieuSoNgay'))
set @sysTable=N'AT14051'
set @AllowNull=N'1'
set @ReadOnly=N'0'
set @Type=N'11'
set @TabIndex=null
set @TabIndexGrid=null
set @TabIndexPopup=N'52'
set @TabIndexView=null
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'OrderFormat'
set @sysTypeID=N'16'
set @GridVisible=N'0'
set @FieldVisible=N'0'

```

```

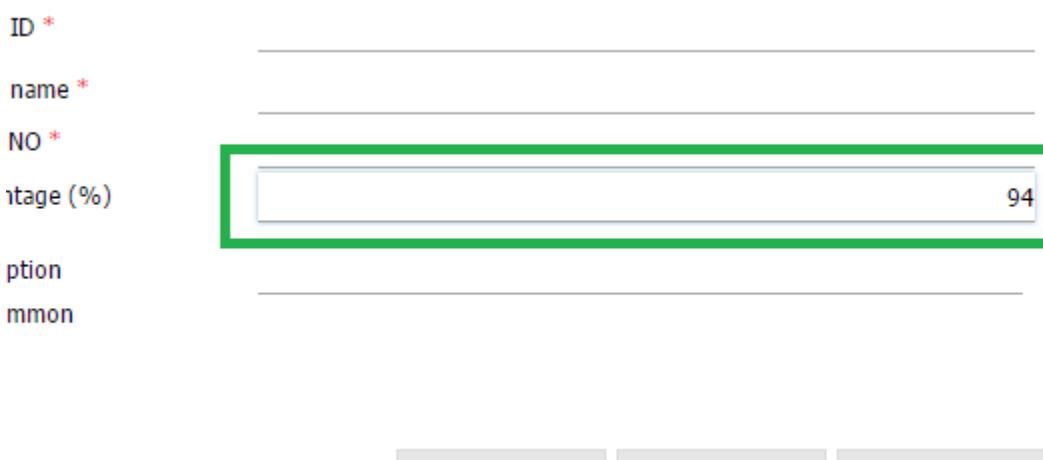
set @PopupVisible=N'1'
set @ViewVisible=N'0'
set @UniqueField=N'0'
set @RefTable=null
set @MaxLenght=null
set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'FieldOrderFormatAT14051'
set @GridWidth=N'100'
set @LanguageID=null
set @IsImport=null
set @GroupOnGrid=null
set @SumOnGrid=null
set @AverageOnGrid=null
set @MaxOnGrid=null
set @MinOnGrid=null
set @RefTableComboboxID=null
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] =
N'FieldOrderFormatAT14051')Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex,TabIndexGrid,TabIndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visible,ColumnName,sysRegularID,sysTypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,UniqueField,RefTable,GroupID,sysEditorTemplateID,MaxLenght,SpecialControl,RefDisplayField,RefValueField,RefViewTable,sysClientTemplateID,FieldID,GridWidth,sysRadioButtonID,LanguageID,IsImport,GroupOnGrid,SumOnGrid,AverageOnGrid,MaxOnGrid,MinOnGrid,RefTableComboboxID)values(@sysTable,@AllowNull,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabIndexPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@ColumnName,@sysRegularID,@sysTypeID,@GridVisible,@FieldVisible,@PopupVisible,@ViewVisible,@UniqueField,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLenght,@SpecialControl,@RefDisplayField,@RefValueField,@RefViewTable,@sysClientTemplateID,@FieldID,@GridWidth,@sysRadioButtonID,@LanguageID,@IsImport,@GroupOnGrid,@SumOnGrid,@AverageOnGrid,@MaxOnGrid,@MinOnGrid,@RefTableComboboxID)
End

```

11) SpinEditNumber

a) Mô tả

- SpinEditNumber control là một textbox và chỉ được nhập số. Số thì có nhiều loại như số nguyên, số thập phân...



b) Cách tạo

- Tạo một Fields có Type là 12, sysTypeID tùy vào kiểu dữ liệu của Field
- Các giá trị còn lại tạo tương tự textbox

```
declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)
declare @ReadOnly nvarchar(MAX)
declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysTypeID nvarchar(MAX)
declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)
declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
declare @MaxLenght nvarchar(MAX)
declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
```

```

declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)

set @sysComboBoxID=null
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @sysRadioButtonID=null
set @GroupID=(select(select top 1 convert(varchar(50),sysGroupID) from sysGroup where GroupID='CIF1321.ThongTinMaTangTuDong')+','+(select top 1 convert(varchar(50),sysGroupID) from sysGroup where GroupID='CIF1322.ThongTinMaTangTuDong')) 
set @sysTable=N'AT1007'
set @AllowNull=N'1'
set @ReadOnly=N'0'
set @Type=N'12'
set @TabIndex=null
set @TabIndexGrid=null
set @TabIndexPopup=N'25'
set @TabIndexView=N'24'
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'OutputLength'
set @sysTypeID=N'5'
set @GridVisible=N'0'
set @FieldVisible=N'0'
set @PopupVisible=N'1'
set @ViewVisible=N'1'
set @UniqueField=N'0'
set @RefTable=null
set @MaxLength=null
set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'FieldOutputLengthAT1007'
set @GridWidth=N'100'

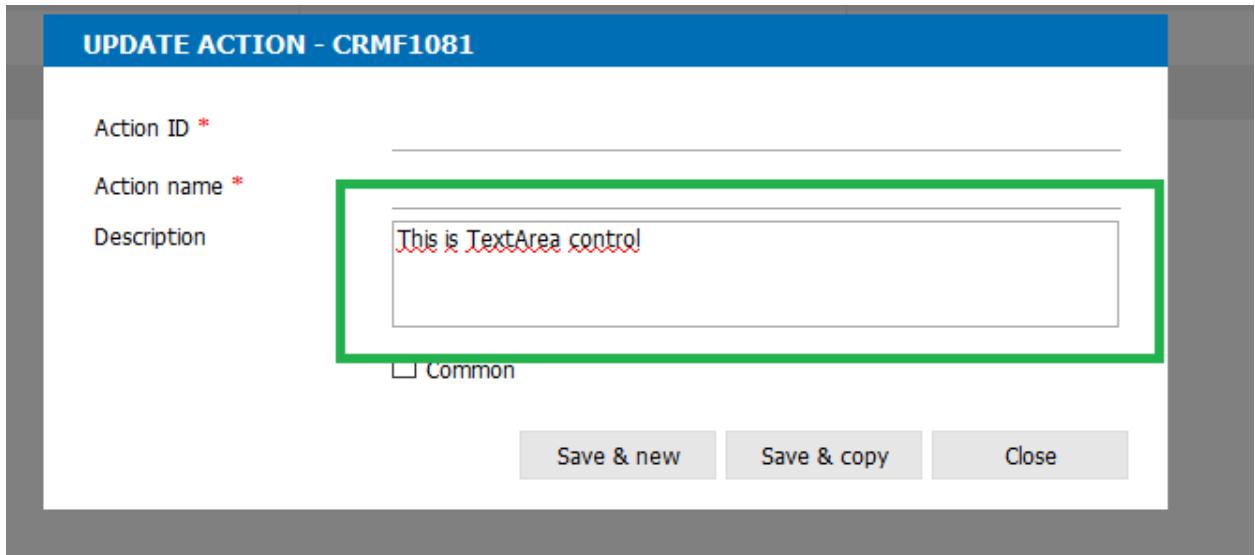
```

```
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] = N'FieldOutputLengthAT1007)Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex,TabIn
dexGrid,TabIndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visi
ble,ColumnName,sysRegularID,sysTypeID,GridVisible,FieldVisible,Popup
Visible,ViewVisible,UniqueField,RefTable,GroupID,sysEditorTemplateID,MaxL
enght,SpecialControl,RefDisplayField,RefValueField,RefViewTable,sysClientTe
mplateID,FieldID,GridWidth,sysRadioButtonID)values(@sysTable,@AllowNull,
@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabIndexP
opup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@Colu
mnName,@sysRegularID,@sysTypeID,@GridVisible,@FieldVisible,@Popup
Visible,@ViewVisible,@UniqueField,@RefTable,@GroupID,@sysEditorTemplate
ID,@MaxLenght,@SpecialControl,@RefDisplayField,@RefValueField,@RefViewT
able,@sysClientTemplateID,@FieldID,@GridWidth,@sysRadioButtonID)
End
```

12) TextArea

a) Mô tả

- TextArea là control dùng để nhập văn bản, hay các trích dẫn chứa nhiều ký tự.



b) Cách tạo

- Tạo một Fields có Type là 13, sysTypeID là 7
- Các giá trị còn lại tạo tương tự textbox

```

declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)
declare @ReadOnly nvarchar(MAX)
declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysDataTypeID nvarchar(MAX)
declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)
declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
declare @MaxLenght nvarchar(MAX)
declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)

set @sysComboBoxID=null
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @GroupID=null
set @sysTable=N'CRMT90031'
set @AllowNull=N'1'
set @ReadOnly=N'0'

```

```

set @Type=N'13'
set @TabIndex=N'2'
set @TabIndexGrid=N'3'
set @TabIndexPopup=N'2'
set @TabIndexView=N'2'
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'Description'
set @sysTypeID=N'7'
set @GridVisible=N'1'
set @FieldVisible=N'1'
set @PopupVisible=N'1'
set @ViewVisible=N'1'
set @UniqueField=null
set @RefTable=null
set @MaxLenght=null
set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'FilterDescriptionCRMT90031'
set @GridWidth=null
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] =
N'FilterDescriptionCRMT90031')Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex,TabIndexGrid,TabIndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visible,ColumnName,sysRegularID,sysTypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,UniqueField,RefTable,GroupID,sysEditorTemplateID,MaxLenght,SpecialControl,RefDisplayField,RefValueField,RefViewTable,sysClientTemplateID,FieldID,GridWidth)values(@sysTable,@AllowNull,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabIndexPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@ColumnName,@sysRegularID,@sysTypeID,@GridVisible,@FieldVisible,@PopupVisible,@ViewVisible,@UniqueField,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLenght,@SpecialControl,@RefDisplayField,@RefValueField,@RefViewTable,@sysClientTemplateID,@FieldID,@GridWidth)
End

```

13) Item Star Level

a) Mô tả

- Item star level có dạng như sau:

UPDATE REQUEST - CRMF2081

Request subject *			Content
Customer *			
Person in charge *	Quản trị hệ thống		
Status *	Chưa thực hiện		Feedback
Priority ID *			
Recording time	08/06/2017 14:55:46		
Deadline	08/06/2017 14:55:46		
			Attach file  
<input type="button" value="Save & new"/> <input type="button" value="Save & copy"/> <input type="button" value="Close"/>			

- Item star level biểu thị mức độ quan trọng, hay cấp độ cần thiết. Sử dụng để đánh giá hay thể hiện mức độ quan trọng...

b) Cách tạo

- Tạo một Fields có Type là 14, sysDataType tùy vào kiểu dữ liệu của Field
- Các giá trị còn lại tạo tương tự textbox

```
declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)
declare @ReadOnly nvarchar(MAX)
declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysTypeID nvarchar(MAX)
declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)
declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
```

```

declare @MaxLenght nvarchar(MAX)
declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)

set @sysComboBoxID=null
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @sysRadioButtonID=null
set @GroupID=null
set @sysTable=N'CRMT90041'
set @AllowNull=N'1'
set @ReadOnly=N'0'
set @Type=N'14'
set @TabIndex=N'3'
set @TabIndexGrid=N'6'
set @TabIndexPopup=N'2'
set @TabIndexView=N'3'
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'PriorityID'
set @sysTypeID=N'6'
set @GridVisible=N'1'
set @FieldVisible=N'1'
set @PopupVisible=N'1'
set @ViewVisible=N'1'
set @UniqueField=null
set @RefTable=null
set @MaxLenght=null
set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'FilterPriorityCRMT90041'

```

```

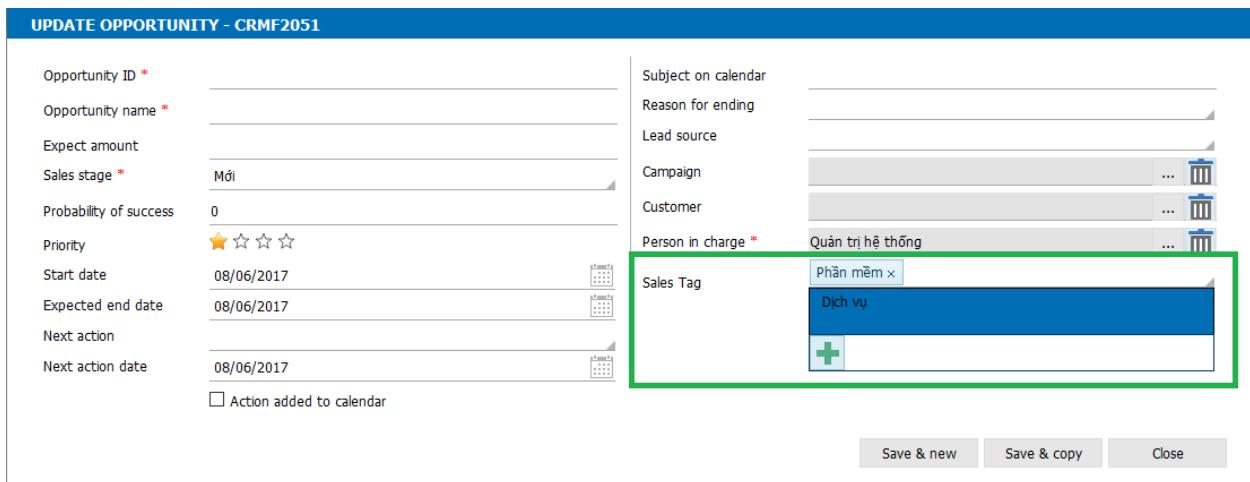
set @GridWidth=null
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] =
N'FilterPriorityCRMT90041')Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex,TabIndexGrid,TabIndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visible,ColumnName,sysRegularID,sysTypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,UniqueField,RefTable,GroupID,sysEditorTemplateID,MaxLength,SpecialControl,RefDisplayField,RefValueField,RefViewTable,sysClientTemplateID,FieldID,GridWidth,sysRadioButtonID)values(@sysTable
,@AllowNull,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabIndexPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@ColumnName,@sysRegularID,@sysTypeID,@GridVisible,@FieldVisible,@PopupVisible,@ViewVisible,@UniqueField,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLength,@SpecialControl,@RefDisplayField,@RefValueField,@RefViewTable,@sysClientTemplateID,@FieldID,
@GridWidth,@sysRadioButtonID)
End

```

14) MultiSelect

a) Mô tả

- MultiSelect control có dạng như sau:



The screenshot shows a form titled "UPDATE OPPORTUNITY - CRMF2051". The "Sales Tag" field is highlighted with a green border. A dropdown menu is open over the field, displaying three items: "Phần mềm x", "Dịch vụ", and a blue button with a white plus sign. The rest of the form contains various input fields like Opportunity ID, Opportunity name, and Start date.

- MultiSelect control cho phép chọn nhiều dòng. Sử dụng cho việc gửi email, hay gán nhiều đối tượng.

b) Cách tạo

Giống với **ComboCheckList** nhưng Field có Type là 15, tham khảo mục IV.4

```

declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)
declare @ReadOnly nvarchar(MAX)

```

```

declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysDataTypeID nvarchar(MAX)
declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)
declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
declare @MaxLenght nvarchar(MAX)
declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)

set @sysComboBoxID=null
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @sysRadioButtonID=null
set @GroupID=(select(select top 1 convert(varchar(50),sysGroupID) from sysGroup where
GroupID='CRMF2042.ThongTinChiTietChienDich'))
set @sysTable=N'CRMT20401'
set @AllowNull=N'0'
set @ReadOnly=N'0'
set @Type=N'15'

```

```

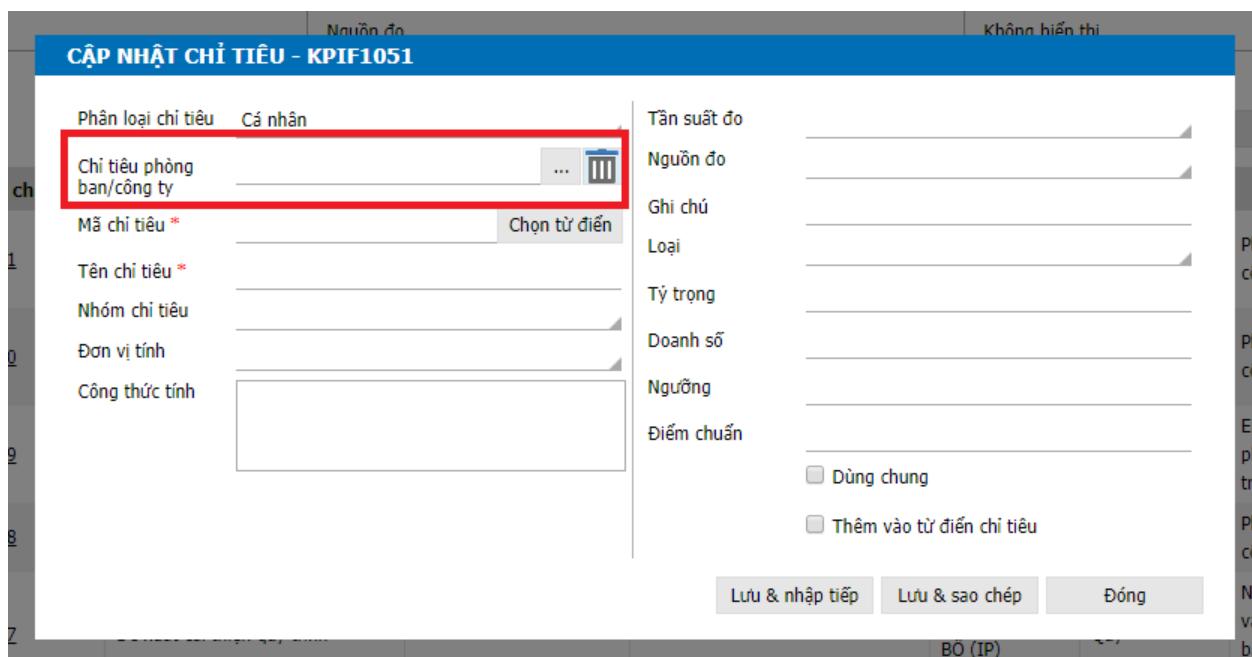
set @TabIndex=N'4'
set @TabIndexGrid=N'8'
set @TabIndexPopup=N'4'
set @TabIndexView=N'5'
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'AssignedToUserID'
set @sysTypeID=N'7'
set @GridVisible=N'1'
set @FieldVisible=N'1'
set @PopupVisible=N'1'
set @ViewVisible=N'1'
set @UniqueField=null
set @RefTable=null
set @MaxLenght=null
set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'FieldAssignedToUserIDCRMT20401'
set @GridWidth=N'150'
set @LanguageID=null
set @IsImport=N'1'
set @RefTableComboboxID=null
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] =
N'FieldAssignedToUserIDCRMT20401')Begin
insert into
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex,TabIndexGrid,TabIndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visible,ColumnName,sysRegularID,sysTypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,UniqueField,RefTable,GroupID,sysEditorTemplateID,MaxLenght,SpecialControl,RefDisplayField,RefValueField,RefViewTable,sysClientTemplateID,FieldID,GridWidth,sysRadioButtonID,LanguageID,IsImport,RefTableComboboxID)values(@sysTable,@AllowNull,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabIndexPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@ColumnName,@sysRegularID,@sysTypeID,@GridVisible,@FieldVisible,@PopupVisible,@ViewVisible,@UniqueField,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLenght,@SpecialControl,@RefDisplayField,@RefValueField,@RefViewTable,@sysClientTemplateID,@FieldID,@GridWidth,@sysRadioButtonID,@LanguageID,@IsImport,@RefTableComboboxID)
End

```

15) Select Object

a. Mô tả

- Select Object control có dạng như sau:



b. Cách tạo

- Tạo một Fields có Type là 16, sysDataType tùy vào kiểu dữ liệu của Field
- Các giá trị còn lại tạo tương tự textbox

```

declare @sysTable nvarchar(MAX)
declare @AllowNull nvarchar(MAX)
declare @ReadOnly nvarchar(MAX)
declare @sysComboBoxID nvarchar(MAX)
declare @Type nvarchar(MAX)
declare @TabIndex nvarchar(MAX)
declare @TabIndexGrid nvarchar(MAX)
declare @TabIndexPopup nvarchar(MAX)
declare @TabIndexView nvarchar(MAX)
declare @MinValue nvarchar(MAX)
declare @MaxValue nvarchar(MAX)
declare @DefaultValue nvarchar(MAX)
declare @Visible nvarchar(MAX)
declare @ColumnName nvarchar(MAX)
declare @sysRegularID nvarchar(MAX)
declare @sysTypeID nvarchar(MAX)
declare @GridVisible nvarchar(MAX)
declare @FieldVisible nvarchar(MAX)
declare @PopupVisible nvarchar(MAX)
declare @ViewVisible nvarchar(MAX)
declare @UniqueField nvarchar(MAX)
declare @RefTable nvarchar(MAX)
declare @GroupID nvarchar(MAX)
declare @sysEditorTemplateID nvarchar(MAX)
declare @MaxLength nvarchar(MAX)

```

```

declare @SpecialControl nvarchar(MAX)
declare @RefDisplayField nvarchar(MAX)
declare @RefValueField nvarchar(MAX)
declare @RefViewTable nvarchar(MAX)
declare @sysClientTemplateID nvarchar(MAX)
declare @FieldID nvarchar(MAX)
declare @GridWidth nvarchar(MAX)
declare @sysRadioButtonID nvarchar(MAX)
declare @LanguageID nvarchar(MAX)
declare @IsImport nvarchar(MAX)
declare @GroupOnGrid nvarchar(MAX)
declare @SumOnGrid nvarchar(MAX)
declare @AverageOnGrid nvarchar(MAX)
declare @MaxOnGrid nvarchar(MAX)
declare @MinOnGrid nvarchar(MAX)

set @sysComboBoxID=null
set @sysRegularID=null
set @sysEditorTemplateID=null
set @sysClientTemplateID=null
set @sysRadioButtonID=null
set @GroupID=null
set @sysTable=N'CRMT90041'
set @AllowNull=N'1'
set @ReadOnly=N'0'
set @Type=N'16'
set @TabIndex=N'3'
set @TabIndexGrid=N'6'
set @TabIndexPopup=N'2'
set @TabIndexView=N'3'
set @MinValue=null
set @MaxValue=null
set @DefaultValue=null
set @Visible=N'1'
set @ColumnName=N'PriorityID'
set @sysDataTypeID=N'6'
set @GridVisible=N'1'
set @FieldVisible=N'1'
set @PopupVisible=N'1'
set @ViewVisible=N'1'
set @UniqueField=null
set @RefTable=null
set @MaxLenght=null
set @SpecialControl=null
set @RefDisplayField=null
set @RefValueField=null
set @RefViewTable=null
set @FieldID=N'FilterPriorityCRMT90041'
set @GridWidth=null

```

```
If not exists(select top 1 1 from [dbo].[sysFields] where [FieldID] =  
N'FilterPriorityCRMT90041')Begin  
insert into  
sysFields(sysTable,AllowNull,ReadOnly,sysComboBoxID,Type,TabIndex,TabIndexGrid,TabI  
ndexPopup,TabIndexView,MinValue,MaxValue,DefaultValue,Visible,ColumnName,sysRegul  
arID,sysDataTypeID,GridVisible,FieldVisible,PopupVisible,ViewVisible,UniqueField,RefTab  
le,GroupID,sysEditorTemplateID,MaxLenght,SpecialControl,RefDisplayField,RefValueField,  
RefViewTable,sysClientTemplateID,FieldID,GridWidth,sysRadioButtonID)values(@sysTable  
,@AllowNull,@ReadOnly,@sysComboBoxID,@Type,@TabIndex,@TabIndexGrid,@TabInd  
exPopup,@TabIndexView,@MinValue,@MaxValue,@DefaultValue,@Visible,@ColumnName  
,@sysRegularID,@sysDataTypeID,@GridVisible,@FieldVisible,@PopupVisible,@ViewVisi  
ble,@UniqueField,@RefTable,@GroupID,@sysEditorTemplateID,@MaxLenght,@SpecialCo  
ntrol,@RefDisplayField,@RefValueField,@RefViewTable,@sysClientTemplateID,@FieldID,  
@GridWidth,@sysRadioButtonID)  
End
```

V) Các bước tạo màn hình chuẩn

- 1) ContentMaster(Màn hình danh mục/tìm kiếm)
- a) Các bước bắt buộc

Danh mục Nghiệp vụ Báo cáo

Nghiệp vụ / Đầu mối

DANH MỤC ĐẦU MỐI - CRMF2030

Đơn vị	Địa chỉ	Tình trạng đầu mối
Mã đầu mối	Điện thoại	Dùng chung
Tên đầu mối	Email	Không hiển thị
Người phụ trách	Nguồn đầu mối	

	STT	Đơn vị	Mã đầu mối	Tên đầu mối	Địa chỉ	Điện thoại	Email	Nguồn đầu mối	Tình trạng đầu mối	Người phụ trách	Dùng chung
<input type="checkbox"/>	1		DM00002	Đầu mối 0002		016951417000 0		Giới thiệu	Mới	Quang hoàng	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2	AS	DM09000001	Đầu mối 001					Mới	Quang hoàng	<input type="checkbox"/>
<input type="checkbox"/>	3	AS	MDM- 0000000001	Đầu mối Hoàng test				Giới thiệu	Mới	Quang hoàng	<input type="checkbox"/>

«
1
»
25
Kết quả/trang

Bước 1: Tạo bảng cho màn hình

Nếu bảng có rồi thì không cần tạo lại.

Nếu chưa có, thực hiện thêm 1 dòng vào bảng sysTable:

sysTable	
TableName	Tên bảng (VD: CRMT20301)
Description	Chú thích về bảng
PK	Khóa chính của bảng (VD: APK)
ModuleID	Module của bảng (VD: AsoftS)
ParentTable	Bảng cha (VD: CRMT20501; có thể có nhiều bảng cha VD: CRMT20501, CRMT10101; không có để NULL)

RefLink	Trường của bảng để click link trên lưới (Dùng trên lưới, không có để NULL)
RefScreenMainID	Mã màn hình nhập liệu của bảng.

Các trường còn lại để NULL

- Sử dụng script

```

declare @TableName nvarchar(50)
declare @Description nvarchar(100)
declare @PK varchar(50)
declare @ModuleID nvarchar(50)
declare @ParentTable nvarchar(50)
declare @RefLink nvarchar(50)

set @TableName = ''
set @Description = ''
set @PK = ''
set @ModuleID = ''
set @ParentTable = ''
set @RefLink = ''
set @RefScreenMainID = ''

insert into sysTable(TableName, Description, PK, ModuleID, ParentTable, RefLink,
RefScreenMainID)
values(@TableName, @Description, @PK, @ModuleID, @ParentTable, @RefLink, @
RefScreenMainID)

```

- Sau khi thêm bảng, tạo bảng tương ứng tại DataBase khách hàng.

Bước 2: Tạo màn hình

Thêm 1 dòng vào bảng sysScreen

sysScreen	
APK	Tự sinh động
DivisionID	Đơn vị (VD: AS)
ModuleID	Module của bảng (VD: AsoftCRM)
ScreenID	ID Màn hình (VD: CRMF2030)
ScreenName	Tên màn hình (VD: Danh mục đầu mối)
ScreenType	Loại màn hình (VD: 2: Màn hình danh mục/tìm kiếm)
ScreenNameE	Tên tiếng anh của màn hình
Parent	Màn hình cha (VD: NULL)
sysTable	Tham chiếu tới bảng sysTable, bảng mà màn hình thuộc vào (VD: CRMT20301)
Title	Title của màn hình (Mã ngôn ngữ để lấy ra tên màn hình VD: CRMF2030.Title)
DisplayToolBar	ToolBar1 của màn hình (VD: 1,2 (Delete, Add)) thám khảo bảng sysToolBar

Các trường còn lại NULL

- Sử dụng script

```

declare @DivisionID nvarchar(50)
declare @ModuleID nvarchar(50)
declare @ScreenID nvarchar(50)
declare @ScreenName nvarchar(250)
declare @ScreenType tinyint
declare @ScreenNameE nvarchar(250)
declare @Parent nvarchar(50)
declare @sysTable nvarchar(50)
declare @Title nvarchar(50)
declare @DisplayToolBar varchar(100)

set @DivisionID = ''
set @ModuleID = ''
set @ScreenID = ''
set @ScreenName = ''
set @ScreenType = 2
set @ScreenNameE = ''
set @Parent = ''
set @sysTable = ''
set @Title = ''
set @DisplayToolBar = ''

insert into sysScreen(DivisionID, ModuleID, ScreenID, ScreenName, ScreenType,
ScreenNameE, Parent,
        sysTable, Title, DisplayToolBar)
values(@DivisionID,@ModuleID,@ScreenID,@ScreenName,@ScreenType,@ScreenNameE,@Par
ent,
        @sysTable,@Title,@DisplayToolBar)

```

Bước 3: Tạo các control cho màn hình

- Thêm dòng vào bảng sysField, số lượng field thêm vào bảng số lượng control trên màn hình, cách thêm giống như thêm textbox, checkbox, combobox.....

b) Customize

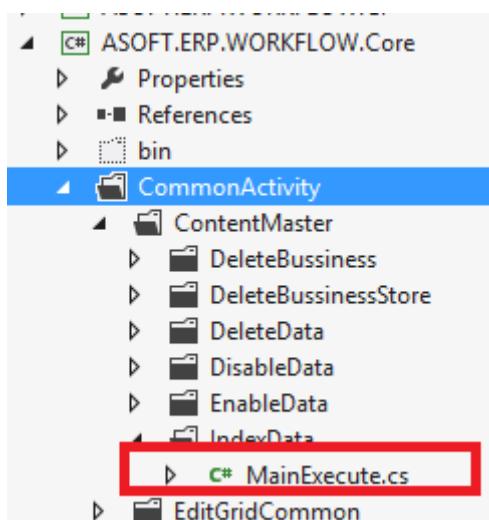
Cấu trúc màn hình tìm kiếm

Nghiệp vụ / Đầu mối

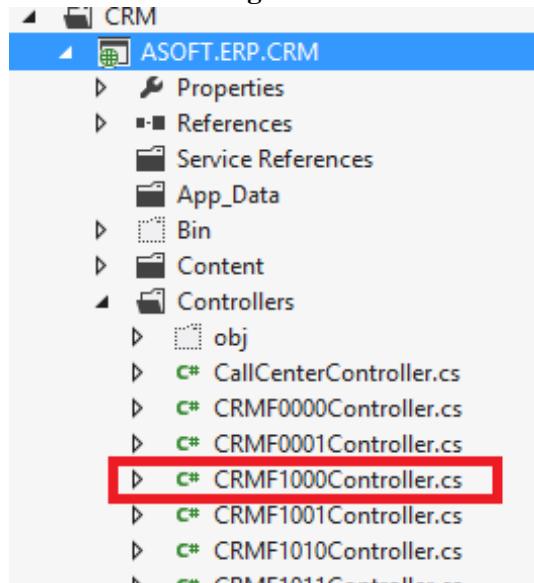
DANH MỤC ĐẦU MỐI - CRMF2030

Đơn vị	Địa chỉ	Tình trạng đầu mối
Mã đầu mối	Điện thoại	Dừng chung
Tên đầu mối	Email	Không hiển thị
Người phụ trách	Nguồn đầu mối	

- Cấu trúc màn hình tìm kiếm sử dụng luồng workflow **IndexData**:
- **IndexData** nằm trong project **ASOFT.ERP.WORKFLOW.Core -> ContentMaster**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để cấu trúc màn hình tìm kiếm.



- Số lượng control search sẽ phụ thuộc vào số lượng field của bảng mà màn hình thuộc vào, những cột hiển thị phụ thuộc vào trường **FieldsVisible** trong bảng **sysFields** trong DB Admin, bảng 0 thì ẩn, 1 thì hiển thị.
- Có 2 trường hợp customize:
 - o Trường hợp 1, customize 1 hoặc nhiều control trong màn hình tìm kiếm
 - Cách customize: (Sửa lại tất cả những chỗ khác)
 - Tạo file class **{Mã màn hình tìm kiếm} + "Controller"** trong project **ASOFT.ERP.{Module}** -> Controllers
 - Thêm action "**ChangeListFields**" vào controller.



- Trong action vừa thêm: code theo cấu trúc:

```

- public List<ASOFTSysFields> ChangeListFields(List<ASOFTSysFields> list, string id)
- {
-     //Code ở đây, tham khảo các file class khác
-     return list;
- }

```

- Trường hợp 2: custom hoàn toàn màn hình.

- Cách custom:

- Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)
- Tạo thư mục **ContentMaster** (nếu chưa có)
- Tạo thư mục **IndexData** nằm trong thư mục **ContentMaster** (nếu chưa có)
- Tạo file class **{Mã màn hình tìm kiếm} + MainExecute**
- Cấu trúc của class custom

```

- public sealed class CRMF2030MainExecute : CodeActivity
- {
-     // Define an activity input argument of type string
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }
-
-     // If your activity returns a value, derive from CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
-     {
-         // Obtain the runtime value of the Text input argument
-         string id = ScreenID.Get(context);
-
-         string module = Area.Get(context);
-
-         ASOFTSysAdminBL bl = new ASOFTSysAdminBL();
-         ASOFTSysScreen scr = new ASOFTSysScreen();
-         List<ASOFTSysFields> list = new List<ASOFTSysFields>();
-         List<ASOFTSysFieldsViewModel> list1 = new List<ASOFTSysFieldsViewModel>();
-         Dictionary<string, Object> Output = InOut.Get(context);
-
-         //Code ở đây, xem các class khác để tham khảo
-
-         Output.Add("Title", scr.TitleName);
-         Output.Add("ScreenID", scr.ScreenID);
-         Output.Add("ScreenPopup", scrpopup); //Màn hình con nhập liệu
-         Output.Add("ScreenMaster", scrmaster); //Màn hình con chi tiết
-         Output.Add("TypeInput", typeinput); //Loại màn hình của màn hình tìm kiếm
-         Output.Add("List", list1); //Danh sách fields trong màn hình
-         Output.Add("DisplayToolBar", scr.DisplayToolBar); //Các button trong
-         toolbar
-         Output.Add("TypeInputGrid", typeinputGrid); //Loại màn hình chi tiết để
-         link trên lưới dẫn tới màn hình chi tiết nào
-         Output.Add("sysCategoryBusinessID", scr.sysCategoryBusinessID); //Màn hình
-         là danh mục hoặc nghiệp vụ
-
-         // Obtain the runtime value of the Text input argument
-     }
- }
```

Cấu trúc lưới màn hình tìm kiếm:

Lưới không group:

Nghiệp vụ / Đánh giá năng lực

DANH MỤC ĐÁNH GIÁ NĂNG LỰC - PAF2000

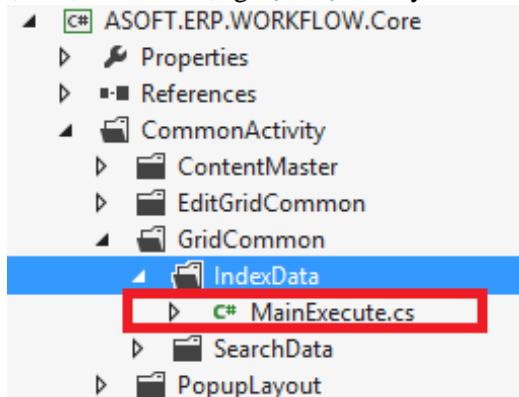
Đơn vị	Phòng ban	Đợt đánh giá
Mã nhân viên	Chức vụ	Bảng đánh giá từng vị trí
Tên nhân viên	Chức danh/vị trí	Người đánh giá

+ -
Lọc dữ liệu
Làm lại

STT	Đơn vị	Mã nhân viên	Tên nhân viên	Bảng đánh giá từng vị trí	Phòng ban	Chức vụ	Chức danh/vị trí	Đợt đánh giá
1	AS	VU	Hoàng Vũ	Bộ đánh giá năng lực năm 2017	PHÒNG KỸ THUẬT	PHÒNG BAN CHỨC NĂNG	CHUYÊN VIÊN QA	Đánh giá năng lực năm 2017

Kết quả trang 1/4

- Cấu trúc lưới sử dụng luồng workflow **IndexData**:
- **IndexData** nằm trong project **ASOFT.ERP.WORKFLOW.Core -> GridCommon**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để cấu trúc lưới.



- Số lượng cột trên lưới sẽ phụ thuộc vào số lượng field của bảng mà màn hình thuộc vào, những cột hiển thị phụ thuộc vào trường **GridVisible** trong bảng **sysFields** trong DB Admin, bảng 0 thì ẩn, 1 thì hiển thị.
- Sẽ có nghiệp vụ đặc thù yêu cầu hiện thị cột mà không tồn tại trong bảng đó:
 - o VD:

Điện thoại	Email	Nguồn đầu môi	Tình trạng đầu môi	Người phụ trách	Dùng chung
016951417000 0		G	Tình trạng đầu môi (LeadStatusName) không thuộc bảng mà màn hình phụ thuộc, lúc nay ta phải custom	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		Giới thiệu	Mới	Quang hoàng	<input type="checkbox"/>

- Cách tạo:

- Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
- Tạo thư mục **GridCommon** (nếu chưa có)
- Tạo thư mục **IndexData** nằm trong thư mục **GridCommon**(nếu chưa có)
- Tạo file class **{Mã màn hình tìm kiếm} + MainExecute**
- Cấu trúc của class custom

```

- public sealed class CMNF9001MainExecute : CodeActivity
- {
-     // Define an activity input argument of type string
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

-     // If your activity returns a value, derive from CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
-     {
-         // Obtain the runtime value of the Text input argument
-         string id = ScreenID.Get(context);
-         string module = Area.Get(context);
-         Dictionary<string, Object> Input = InOut.Get(context);
-         List<ASOFTSysFields> list = new List<ASOFTSysFields>();
-         ASOFTSysScreen scr = new ASOFTSysScreen();
-         List<string> list1 = new List<string>();
-         string table = Input["table"].ToString();

-         //Code ở đây, thám khảo các class đã làm

-         //Các biến bắt buộc phải trả về
-         Input.Add("sysFields", list); //List cột trên lưới
-         Input.Add("Language", list1); //Ngôn ngữ tương ứng của các cột
-         Input.Add("scr", scr); //Màn hình tìm kiếm
-     }
- }
```

Lưới group:

STT	Năng lực	Mức độ quan trọng	Mức năng lực tiêu chuẩn	Điểm chuẩn	Mức năng lực đánh giá	Mức năng lực đánh giá lại	Điểm đánh giá
AS005-Năng lực chuyên môn							
1	Thương lượng, đàm phán và giải quyết bối thường	2	5	25.00	4.00	3.00	20.00
2	Giao nhận hàng hóa XNK	1	5	12.50	4.00	3.00	10.00
3	Lập kế hoạch mua hàng	2	5	25.00	3.00	3.00	15.00
4	Tìm kiếm, khai thác nguồn cung và hoàn tất thủ tục mua hàng	3	5	37.50	4.00	3.00	30.00
		8.00	100.0		75.00		

- Group lướt:
 - o Khai báo template cho trường GroupOnGrid trong bảng sysFields.
 - o VD:

- Script:
`update sysFields set GroupOnGrid = '{Template của group}' where FieldID = '{ID của field muốn group}'`
 - Tổng trên lưới (tính tổng của 1 cột thuộc group):
 - Khai báo template cho trường SumOnGrid trong bảng sysFields.
 - Script:
`update sysFields set SumOnGrid = '{Template của Sum}' where FieldID = '{ID của field muốn Sum}'`
 - Trung bình trên lưới (tính trung bình của 1 cột thuộc group):
 - Khai báo template cho trường AverageOnGrid trong bảng sysFields.
 - Script:
`update sysFields set AverageOnGrid = '{Template của Average}' where FieldID = '{ID của field muốn Average}'`
 - Max trên lưới (giá trị lớn nhất của 1 cột thuộc group):
 - Khai báo template cho trường MaxOnGrid trong bảng sysFields.

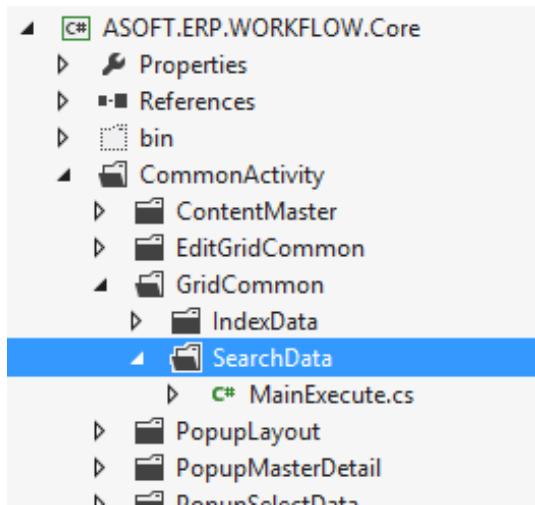
- Script:
`update sysFields set MaxOnGrid = '{Template của Max}' where FieldID = '{ID của field muốn Max}'`
- Min trên lưới (giá trị nhỏ nhất của 1 cột thuộc group):
 - Khai báo template cho trường MaxOnGrid trong bảng sysFields.
 - Script:
`update sysFields set MinOnGrid = '{Template của Min}' where FieldID = '{ID của field muốn Min}'`
- Count trên lưới (tính tổng số lượng của 1 cột thuộc group):
 - Khai báo template cho trường MaxOnGrid trong bảng sysFields.
 - Script:
`update sysFields set CountOnGrid = '{Template của Count}' where FieldID = '{ID của field muốn Count}'`

Custom cấu trúc lưới group giống với custom lưới không group

Lọc lưới/Search (button Lọc dữ liệu):

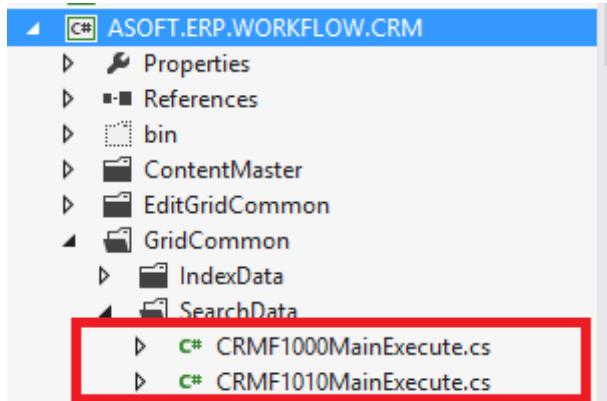


- Lọc dữ liệu/search sử dụng luồng workflow SearchData:
 - SearchData nằm trong project **ASOFT.ERP.WORKFLOW.Core -> GridCommon**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để search dữ liệu trả về controller



- Nếu nghiệp vụ search yêu cầu sử dụng store và câu SQL đặc thù thì ta sẽ custom:
- VD:
 - Cách tạo:
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **GridCommon** (nếu chưa có)
 - Tạo thư mục **SearchData** nằm trong thư mục **GridCommon** (nếu chưa có)

- Tạo file class {Mã màn hình tìm kiếm} + MainExecute



- Cấu trúc của class custom

```

- public sealed class CRMF1000MainExecute : CodeActivity
- {
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

-     // If your activity returns a value, derive from CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
{
    string id = ScreenID.Get(context).ToString();
    string area = Area.Get(context).ToString();
    ASOFTGridBL bl = new ASOFTGridBL();
    int requestPage =
int.Parse(InOut.Get(context)["requestPage"].ToString());
    int requestPageSize =
int.Parse(InOut.Get(context)["requestPageSize"].ToString());
    Dictionary<string, List<string>> args = (Dictionary<string,
List<string>>)InOut.Get(context)["args"];
    // Obtain the runtime value of the Text input argument
    ASOFTSysGrid ds = new ASOFTSysGrid();

    //Code ở đây, thám khảo các class đã làm

    //Các biến bắt buộc phải trả về
    Dictionary<string, Object> Ouput = InOut.Get(context);
    Ouput.Add("kq", ds); //Dữ liệu trả về
    InOut.Set(context, Ouput);
}
}

```

Custom button [Làm lại]

- Mặc định khi nhấn nút [Làm lại], tất cả các field trên màn hình sẽ xóa trắng, kể cả combobox, comboChecklist, DatePicker + Load lại Grid
- Nếu muốn xử lý xóa trắng theo ý của mình, ta định nghĩa hàm `clearfieldsCustomer` trong JavaCustom(Xem chi tiết JavaCustom bên dưới)

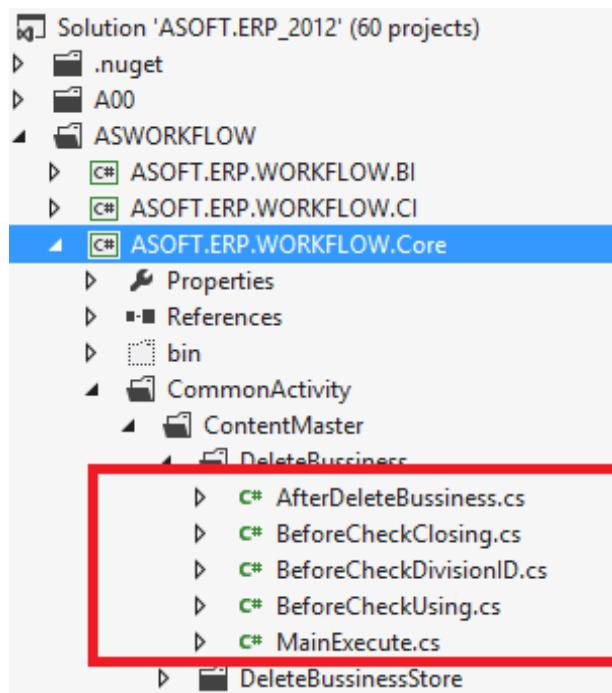
```
function clearfieldsCustomer () {
    $("#FromDatePeriodControl").data("kendoDatePicker").value(new Date());
};
```

Xóa trên lưới (button xóa)



Có 2 luồng xóa:

- Xóa danh mục sẽ có 2 luồng workflow:
 - o **DeleteData** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động **mặc định** bao gồm:
 - **BeforeCheckDivisionID**: kiểm tra khác Division thì không cho xóa.
 - **BeforeCheckUsing**: kiểm tra đã sử dụng thì không cho xóa.
 - **MainExecute**: thực hiện xóa dữ liệu.
 - **AfterDelete**: sau khi delete sẽ thực hiện gì.
 - Mỗi hành động tương ứng với 1 class nằm trong thư mục **CommonActivity** -> **DeleteData** (thuộc project **ASOFT.ERP.WORKFLOW.Core**).

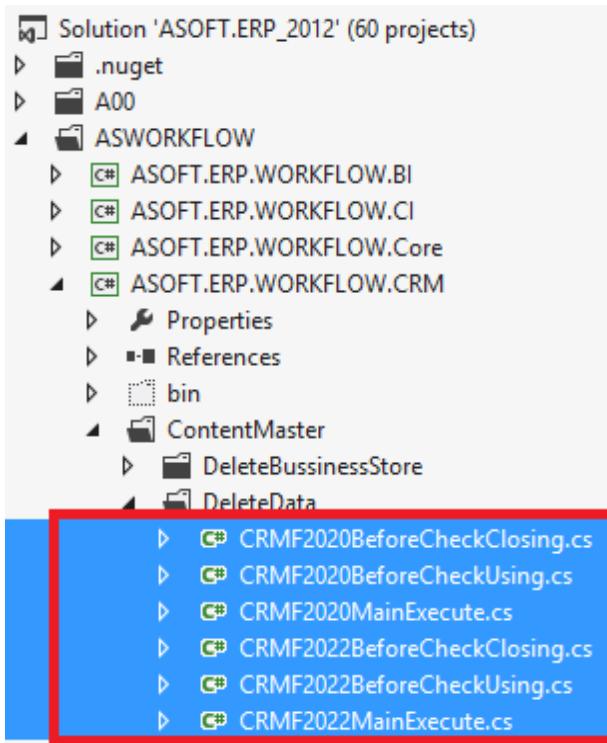


Mỗi nghiệp vụ xóa sẽ có đặc thù riêng nên nhiều khi chúng ta không thể thực hiện theo các hành động mặc định như trên được. Do đó ta phải thực hiện custom các hành động:

VD: Custom hành động **BeforeCheckUsing**.

- **Cách tạo:**

- Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)
- Tạo thư mục **ContentMaster** (nếu chưa có)
- Tạo thư mục **DeleteData** nằm trong thư mục **ContentMaster** (nếu chưa có)
- Tạo file class **{Mã màn hình tìm kiếm} + BeforeCheckUsing.**



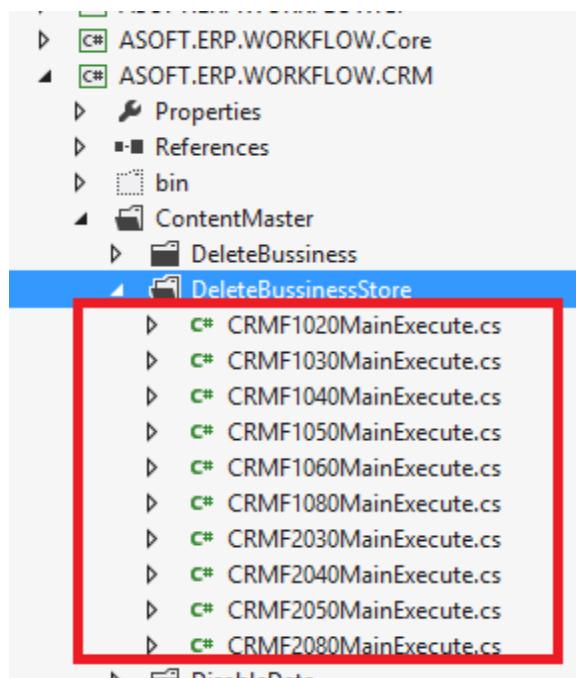
- Cấu trúc của class custom

```
public sealed class CRMF2020BeforeCheckUsing : CodeActivity
{
    public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
    //Dữ liệu được truyền vào từ controller
    public InOutArgument<string> Area { get; set; } //Module
    public InOutArgument<string> ScreenID { get; set; } //Mã màn hình

    protected override void Execute(CodeActivityContext context)
    {
        Dictionary<string, object> er = InOut.Get(context); // Cách lấy biến
        InOut để sử dụng.
        //Tương tự với biến Area ScreenID

        /*
         * {Code} Tham khảo những file đã làm rồi
         */
    }
}
```

- Tương tự với những hành động khác: {Mã màn hình tìm kiếm} + **BeforeCheckDivisionID**, : {Mã màn hình tìm kiếm} + **MainExecute**
 - **DeleteBussinessStore** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: để thực hiện được workflow này ta phải khai báo trong bảng sysScreen trường **DeleteBussinessStore** của màn hình tìm kiếm.
 - Script: `update sysScreen set DeleteBussinesStore = {Mã store} where ScreenID = {Mã màn hình tìm kiếm}`
 - VD: update sysScreen set DeleteBussinessStore = 'CRMP20302' where ScreenID = 'CRMF2030'.
- DeleteBussinessStore** có các hành động bao gồm
- **MainExecute**: thực hiện xóa dữ liệu bằng store.
 - **AfterDeleteStore**: sau khi delete sẽ thực hiện gì.
 - Tại đây workflow sẽ tự chạy store với những parameter của store được định sẵn. Và tất nhiên không phải parameter định sẵn lúc nào cũng đúng với store được khai báo. Đối với trường hợp này ta sẽ custom cũng giống như luồng workflow **DeleteData**.
- VD:
- **Cách tạo:**
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **ContentMaster** (nếu chưa có)
 - Tạo thư mục **DeleteBussinessStore** nằm trong thư mục **ContentMaster** (nếu chưa có)
 - Tạo file class **{Mã màn hình tìm kiếm} + MainExecute**.



- Cấu trúc của class custom:

```

- public sealed class CRMF2030MainExecute : CodeActivity
- {
-     // Define an activity input argument of type string
-     public InOutArgument<Dictionary<string, Object>> InOut { get;
- set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

-         // If your activity returns a value, derive from
-         CodeActivity<TResult>
-         // and return the value from the Execute method.
-         protected override void Execute(CodeActivityContext context)
{
-             // Obtain the runtime value of the Text input argument

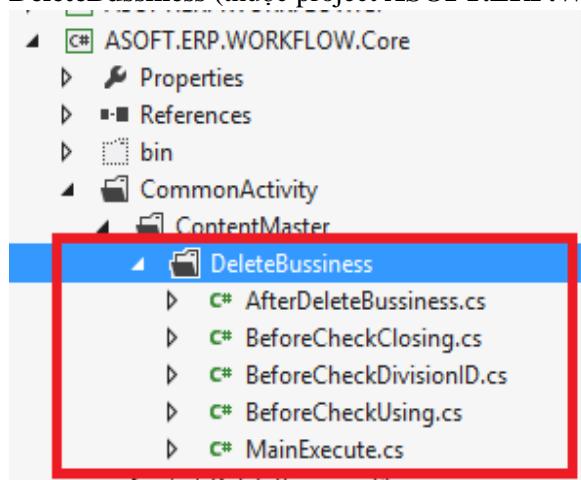
-             //VD: Code mẫu
-             Dictionary<string, Object> data = InOut.Get(context);
-             List<string> dt = (List<string>)data["data"];
-             //VD: Code mẫu

-         }
}

```

- Xóa nghiệp vụ cũng có 2 luồng workflow:
 - o **DeleteBussiness** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động bao gồm:
 - **BeforeCheckDivisionID**: kiểm tra khác Division thì không cho xóa.
 - **BeforeCheckClosing**: kiểm tra khóa sổ thì không cho xóa.
 - **BeforeCheckUsing**: kiểm tra đã sử dụng thì không cho xóa.
 - **MainExecute**: thực hiện xóa dữ liệu.
 - **AfterDeleteBussiness**: sau khi delete sẽ thực hiện gì.

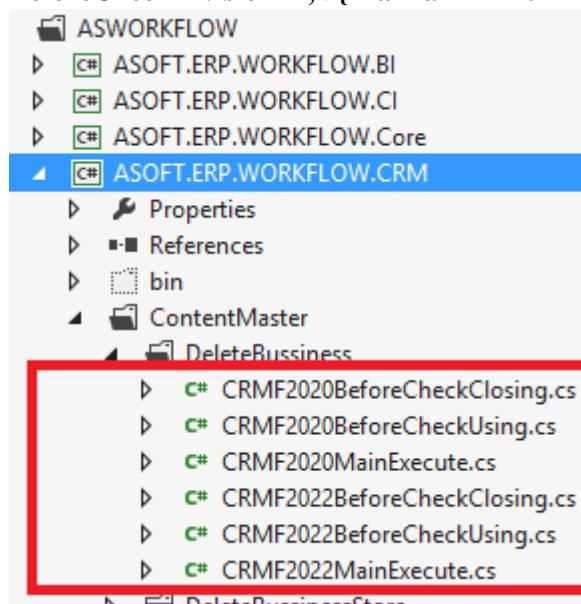
- Mỗi hành động tương ứng với 1 class nằm trong thư mục **CommonActivity -> DeleteBussiness** (thuộc project **ASOFT.ERP.WORKFLOW.Core**).



Cũng tương tự như xóa danh mục sẽ có nghiệp vụ xóa không chạy theo những hành động mặc định này. Vậy nên chúng ta phải Custom như sau:

VD: Custom hành động **BeforeCheckUsing**.

- Cách tạo:**
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **ContentMaster** (nếu chưa có)
 - Tạo thư mục **DeleteBussiness** nằm trong thư mục **ContentMaster** (nếu chưa có)
 - Tạo file class **{Mã màn hình tìm kiếm} + BeforeCheckUsing**.
 - Cấu trúc của class custom: tương tự như workflow **DeleteData**
- Tương tự với những hành động khác: **{Mã màn hình tìm kiếm} + BeforeCheckDivisionID**, **{Mã màn hình tìm kiếm} + MainExecute ...**

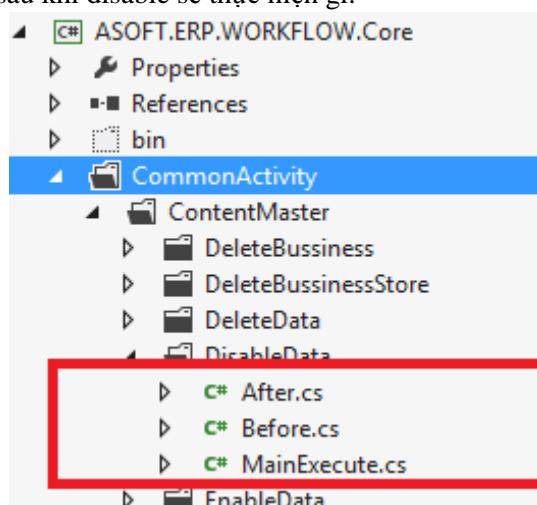


- **DeleteBussinessStore:** tương tự như **DeleteBussinessStore** trong xóa danh mục

Disable dữ liệu trên lưới (button disable)



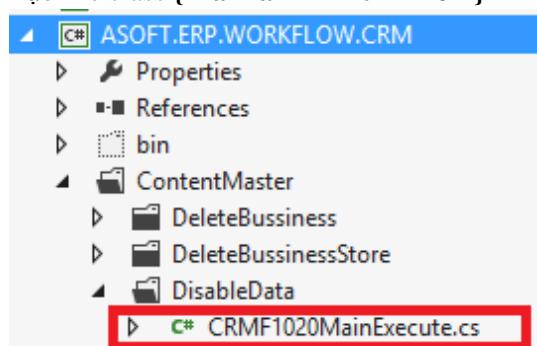
- **Disable** dữ liệu trên lưới sử dụng luồng workflow **DisableData**
- **DisableData** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động bao gồm:
 - **Before:** trước khi disable sẽ thực hiện gì.
 - **MainExecute:** thực hiện disable dữ liệu.
 - **After:** sau khi disable sẽ thực hiện gì.



Sẽ có nghiệp vụ disable không chạy theo những hành động mặc định này. Vậy nên chúng ta phải Custom nhu sau:

VD: Custom hành động **MainExecute**.

- **Cách tạo:**
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)
 - Tạo thư mục **ContentMaster** (nếu chưa có)
 - Tạo thư mục **DisableData** nằm trong thư mục **ContentMaster** (nếu chưa có)
 - Tạo file class **{Mã màn hình tìm kiếm} +MainExecute**.



- Cấu trúc của class custom:

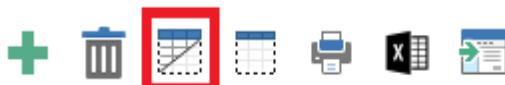
```

- public sealed class CRMF1020MainExecute : CodeActivity
- {
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

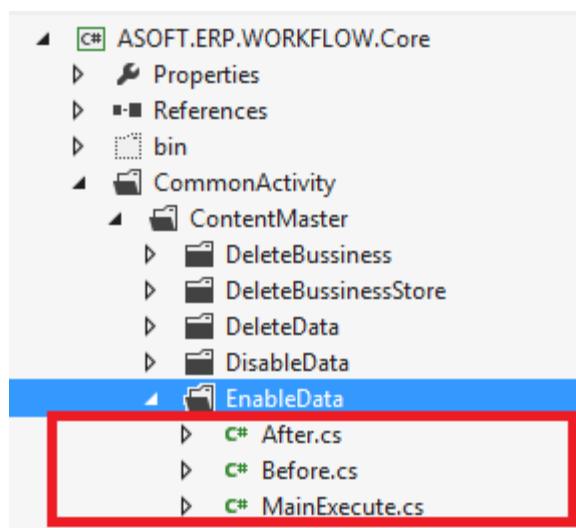
-     // If your activity returns a value, derive from CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
-     {
-         // Code ở đây, tham khảo class khác
-     }
- }
```

- Tương tự với những hành động khác: {Mã màn hình tìm kiếm} + Before, : {Mã màn hình tìm kiếm} + After

Enable dữ liệu trên lưới (button Enable)



- **Enable** dữ liệu trên lưới sử dụng luồng workflow **EnableData**
- **EnableData** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động bao gồm:
 - **Before**: trước khi disable sẽ thực hiện gì.
 - **MainExecute**: thực hiện disable dữ liệu.
 - **After**: sau khi disable sẽ thực hiện gì.

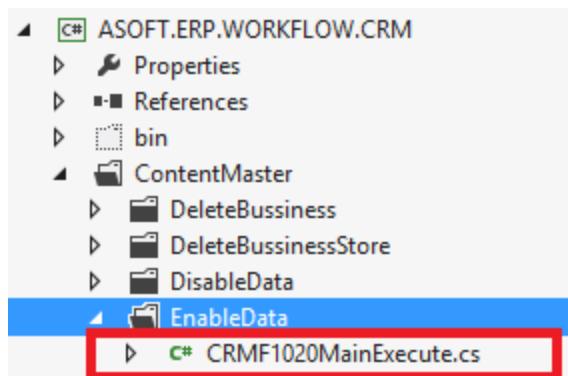


Sẽ có nghiệp vụ enable không chạy theo những hành động mặc định này. Vậy nên chúng ta phải Custom như sau:

VD: Custom hành động **MainExecute**.

- **Cách tạo:**

- Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
- Tạo thư mục **ContentMaster** (nếu chưa có)
- Tạo thư mục **EnableData** nằm trong thư mục **ContentMaster** (nếu chưa có)
- Tạo file class **{Mã màn hình tìm kiếm} +MainExecute.**



- Cấu trúc của class custom: cấu trúc giống **DisableData**

In/Export dữ liệu:



- In/Export sẽ thực hiện theo 2 luồng:
 - Luồng chuẩn: Để thực hiện được luồng chuẩn ta phải khai báo trường (SqlFilter hoặc StoreFilter), nếu khai báo SqlFilter thì không khai báo StoreFilter và ngược lại nếu khai báo StoreFilter thì khai báo thêm **StoreFilterParameter**.
 - Script: `update sysScreen set SqlFilter = {SQL} where ScreenID = {Mã màn hình tìm kiếm}` hoặc `update sysScreen set StoreFilter = {Store In}, StoreFilterParameter = {Parameter của store} where ScreenID = {Mã màn hình tìm kiếm}`
 - Tạo file excel template trong project **ASOFT.ERP.{Module}** -> thư mục **Content** -> thư mục **Report** -> **{Màn hình tìm kiếm} + Report.xlsx**
 - Luồng custom: nếu muốn custom ta định nghĩa hàm **PrintClick** đối với **In** và **ExportClick** đối với **Export** trong JavaCustom(Xem chi tiết JavaCustom bên dưới)

```

- function PrintClick _Click() {
-     //Code trong đây
- }
- function BtnExport_Click() {
-     //Code trong đây
- }
```

Import dữ liệu



- Import sẽ thực hiện theo 2 luồng:
 - o Luồng chuẩn:
 - Để thực hiện được luồng chuẩn ta phải khai báo trường IsImport = 1 trong bảng sysFields với tất cả các fields nằm trong bảng của màn hình tìm kiếm.

```
SELECT TOP (200) sysFieldID, sysTable, AllowNull, ReadOnly, sysComboBoxID, Type, ColumnName, IsImport, FieldID, RefTableComboBoxID
FROM sysFields
WHERE (sysTable = 'CRMT20301')
```

	sysFieldID	sysTable	AllowNull	ReadOnly	sysComboBoxID	Type	ColumnName	IsImport	FieldID	RefTableCom...
▶	20378	CRMT20301	1	0	NULL	1	APK	NULL	FieldAPKCRM...	NULL
	20379	CRMT20301	1	0	3004	4	DivisionID	1	FieldDivisionID...	NULL
	20380	CRMT20301	0	1	NULL	1	LeadID	1	FieldLeadIDCR...	NULL
	20381	CRMT20301	0	0	NULL	1	LeadName	1	FieldLeadName...	NULL
	20382	CRMT20301	1	0	3014	3	Prefix	NULL	FieldPrefixCRM...	NULL
	20383	CRMT20301	1	0	NULL	1	LeadTel	1	FieldLeadTelCR...	NULL
	20384	CRMT20301	1	0	NULL	1	Address	1	FieldAddressCR...	NULL

- Tạo template excel import trong project ASOFT.ERP.{Module} -> Content -> Template -> {Table của màn hình tìm kiếm}.xlsx
- Khai báo trường StartRowImport trong bảng sysTable với table của màn hình tìm kiếm.
 - Script: `update sysScreen set StartRowImport = {Dòng bắt đầu import phòng file import} where ScreenID = {Table màn hình tìm kiếm}`
 - VD:

A	B	C	D	E	F	G
1						
2						
3						
4						
5	Đơn vị*	Mã liên hệ*	Tên liên hệ*	Địa chỉ	Điện thoại *	Email
6	DivisionID	ContactID	ContactName	Address	HomeMobile	HomeEmail
7	Dữ liệu chuỗi (string)					
8						
9						
10						
11						
12						
13						
14						
15						

Ta thấy dòng bắt đầu nhập dữ liệu trong file import là 8 => `{Dòng bắt đầu import phòng file import} = 8`

- o Luồng custom: ta thực hiện unbind sự kiện của button Import trong JavaCustom(Xem chi tiết JavaCustom bên dưới)

```
- $("#btnImport").unbind();
- $("#btnImport").kendoButton({
    "click":{ //function custom }
});
```

2) Màn hình nhập liệu không lướt detail(Thêm/Sửa)

CẬP NHẬT ĐẦU MỐI - CRMF2031

Thông tin chung	Thông tin cá nhân	Thông tin công ty
Nguồn đầu mối	Nghề nghiệp	
Mã đầu mối *	Tình trạng đầu mối	Mới
Xưng hô	Người phụ trách *	Quang hoàng
Tên đầu mối *	Ghi chú	
Điện thoại	<input type="checkbox"/> Dùng chung	
Địa chỉ		
Email		
<input type="button" value="Lưu & nhập tiếp"/> <input type="button" value="Lưu & sao chép"/> <input type="button" value="Đóng"/>		

a) Các bước bắt buộc

Bước 1: Tạo bảng cho màn hình

- Nếu bảng có rồi thì không cần tạo lại.
- Nếu chưa có, thực hiện thêm 1 dòng vào bảng sysTable:

sysTable	
TableName	Tên bảng (VD: CRMT20301)
Description	Chú thích về bảng
PK	Khóa chính của bảng (VD: APK)
ModuleID	Module của bảng (VD: AsoftS)
ParentTable	Bảng cha (VD: CRMT20501; có thể có nhiều bảng cha VD: CRMT20501, CRMT10101; không có để NULL)
RefLink	Trường của bảng để click link trên lưới (Dùng trên lưới, không có để NULL)
RefScreenMainID	Mã màn hình nhập liệu của bảng.

```

declare @TableName nvarchar(50)
declare @Description nvarchar(100)
declare @PK varchar(50)
declare @ModuleID nvarchar(50)
declare @ParentTable nvarchar(50)
declare @RefLink nvarchar(50)

set @TableName = ''
set @Description = ''
set @PK = ''
set @ModuleID = ''
set @ParentTable = ''
set @RefLink = ''
set @RefScreenMainID = ''

```

```
insert into sysTable(TableName, Description, PK, ModuleID, ParentTable, RefLink,
RefScreenMainID)
values(@TableName, @Description, @PK, @ModuleID, @ParentTable, @RefLink, @
RefScreenMainID)
```

Bước 2: Tạo màn hình

Thêm 1 dòng vào bảng sysScreen

- Màn hình popup no detail

sysScreen	
APK	Tự sinh động
DivisionID	Đơn vị (VD: AS)
ModuleID	Module của bảng (VD: AsoftCRM)
ScreenID	ID Màn hình (VD: CRMF2031)
ScreenName	Tên màn hình (VD: Cập nhật đầu mối)
ScreenType	Loại màn hình (VD: 3: Màn hình nhập liệu)
ScreenNameE	Tên tiếng anh của màn hình
Parent	Màn hình cha (VD: CRMF2030 Màn hình tìm kiếm)
sysTable	Tham chiếu tới bảng sysTable, bảng mà màn hình thuộc vào (VD: CRMT20301)
Title	Title của màn hình (Mã ngôn ngữ để lấy ra tên màn hình VD: CRMF2031.Title)
TypeInput	VD: 1: Loại màn hình nhập liệu không lưới detail
sysCategoryBusinessID	1: Danh mục, 2:Nghiệp Vụ
Width	Chiều rộng của lưới (VD: 800)

- Sử dụng script

```
declare @DivisionID nvarchar(50)
declare @ModuleID nvarchar(50)
declare @ScreenID nvarchar(50)
declare @ScreenName nvarchar(250)
declare @ScreenType tinyint
declare @ScreenNameE nvarchar(250)
declare @Parent nvarchar(50)
declare @sysTable nvarchar(50)
declare @Title nvarchar(50)
declare @TypeInput varchar(100)
declare @sysCategoryBusinessID varchar(100)
declare @Width varchar(100)

set @DivisionID = ''
set @ModuleID = ''
set @ScreenID = ''
set @ScreenName = ''
set @ScreenType = 3
set @ScreenNameE = ''
```

```

set @Parent = ''
set @sysTable = ''
set @Title = ''
set @TypeInput = ''
set @sysCategoryBusinessID = ''
set @Width = ''

insert into sysScreen(DivisionID, ModuleID, ScreenID, ScreenName, ScreenType,
ScreenNameE, Parent,
        sysTable, Title, TypeInput, sysCategoryBusinessID, Width)
values(@DivisionID,@ModuleID,@ScreenID,@ScreenName,@ScreenType,@ScreenNameE,@Par
ent,
        @sysTable,@Title,@TypeInput,@sysCategoryBusinessID,@Width)

```

Bước 3: Tạo các control cho màn hình

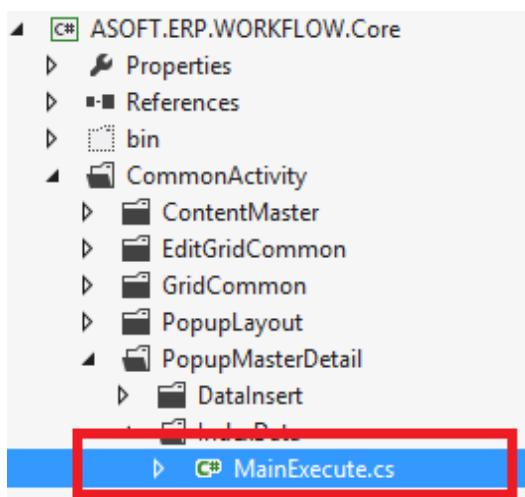
- Thêm dòng vào bảng sysField, số lượng field thêm vào bảng số lượng control trên màn hình, cách thêm giống như thêm textbox, checkbox, combobox.....

b) Customize

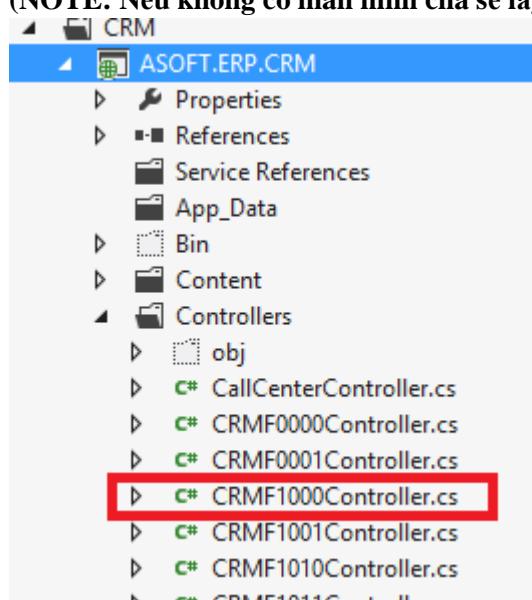
Cấu trúc màn hình nhập liệu không lưới detail



- Cấu trúc màn hình tìm kiếm sử dụng luồng workflow **IndexData**:
- **IndexData** nằm trong project **ASOFT.ERP.WORKFLOW.Core -> PopupLayout**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để cấu trúc lưới.



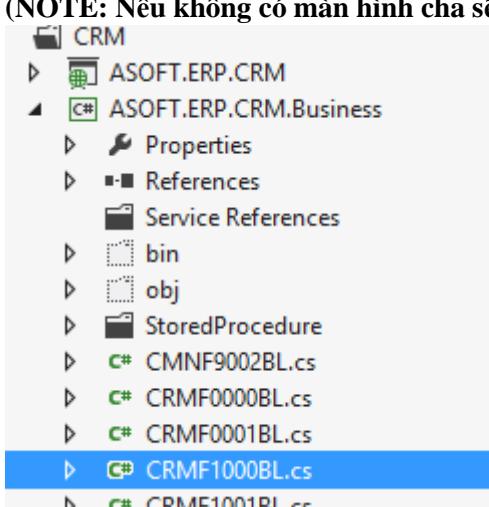
- Số lượng control search sẽ phụ thuộc vào số lượng field của bảng mà màn hình thuộc vào, những cột hiển thị phụ thuộc vào trường **PopupVisible** trong bảng **sysFields** trong **DB Admin**, bảng 0 thì ẩn, 1 thì hiển thị.
- Có 2 trường hợp custom:
 - o Trường hợp 1, custom 1 hoặc nhiều control trong màn hình tìm kiếm
 - Cách custom:
 - Tạo file class **{Mã màn hình cha của màn hình nhập liệu} + "Controller"** trong project ASOFT.ERP.{Module} -> Controllers.
 - (NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)



- Thêm action "**ChangeListFieldsPopup**" vào controller.
- Trong action vừa thêm: code theo cấu trúc:

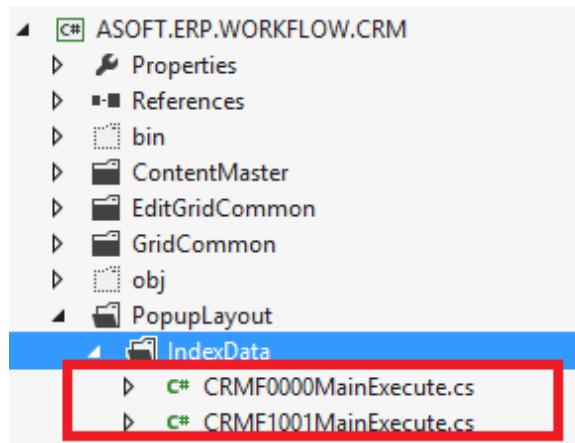
```
- public List<ASOFTSysFields> ChangeListFieldsPopup(List<ASOFTSysFields> list, string
id)
{
    //Code ở đây, tham khảo các file class khác
    return list;
}
```

- }

- Trường hợp 2: custom câu SQL load màn hình nhập liệu (trường hợp sửa)
 - Cách custom:
 - Tạo file class **{Mã màn hình cha của màn hình nhập liệu} + “BL”** trong project **ASOFT.ERP.{Module}.Business** (Nếu chưa có file) **(NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)**
- 
- Thêm hàm **“GetData”** vào class.
 - Trong class vừa thêm, code theo cấu trúc:

```
public Dictionary<string, object> GetData(string APK, string id)
{
    try
    {
        //Code tại đây
    }
    catch (ASOFTEception)
    {
        throw;
    }
    catch (Exception ex)
    {
        throw new ASOFTEception(ASOFTLayer.Business, ex.Message, ex);
    }
}
```

- Trường hợp 3: custom hoàn toàn màn hình.
 - Cách custom:
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)
 - Tạo thư mục **PopupLayout** (nếu chưa có)
 - Tạo thư mục **IndexData** nằm trong thư mục **PopupLayout** (nếu chưa có)
 - Tạo file class **{Mã màn hình nhập liệu} + MainExecute**



- Cấu trúc của class custom

```

public sealed class CRMF2031MainExecute : CodeActivity
{
    // Define an activity input argument of type string
    public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
    public InOutArgument<string> Area { get; set; }
    public InOutArgument<string> ScreenID { get; set; }

    // If your activity returns a value, derive from CodeActivity<TResult>
    // and return the value from the Execute method.
    protected override void Execute(CodeActivityContext context)
    {
        ASOFTGridBL bl1 = new ASOFTGridBL();
        // Obtain the runtime value of the Text input argument
        string id = ScreenID.Get(context).ToString();
        string module = Area.Get(context).ToString();
        string key = InOut.Get(context)[ "key" ] != null ?
        InOut.Get(context)[ "key" ].ToString() : null;
        string PK = InOut.Get(context)[ "PK" ] != null ?
        InOut.Get(context)[ "PK" ].ToString() : null;
        string Table = InOut.Get(context)[ "Table" ] != null ?
        InOut.Get(context)[ "Table" ].ToString() : null;
        string divisionID = InOut.Get(context)[ "DivisionID" ].ToString();
        var formStt = ASOFTFormActionType.AddNew;
        Dictionary<string, object> data = new Dictionary<string, object>();

        ASOFTSysAdminBL bl = new ASOFTSysAdminBL();
        ASOFTSysScreen scr = new ASOFTSysScreen();
        List<ASOFTSysFields> list = new List<ASOFTSysFields>();
        List<ASOFTSysFieldsViewModel> list1 = new List<ASOFTSysFieldsViewModel>();
        List<ASOFTSysGroup> listGroup = new List<ASOFTSysGroup>();
        List<ASOFTSysGroupViewModel> listGroup1 = new List<ASOFTSysGroupViewModel>();

        //Code tại đây, tham khảo các class khác

        Dictionary<string, Object> Output = InOut.Get(context);

        Output.Add("Title", LocalizedHelper.GetLocalizedResourceString(scr.TitleName, id,
        module) + " - " + scr.ScreenID); //Title màn hình nhập liệu
    }
}

```

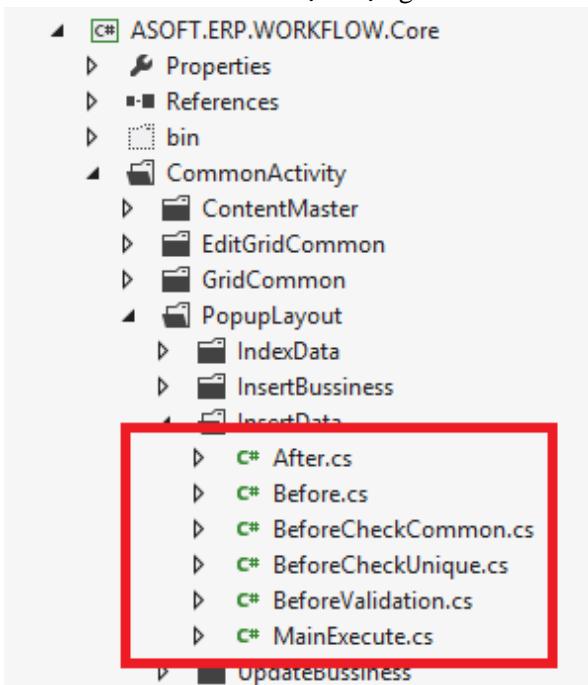
```

        Output.Add("sysTable", list[0].sysTable); //Table màn hình nhập liệu
        Output.Add("List", list1); //Filter tương ứng vs control trên màn hình
        Output.Add("sysCategoryBusinessID", scr.sysCategoryBusinessID); //Nghiệp vụ or
danh mục
        Output.Add("listGroup", listGroup1); //Group nếu có.
        Output.Add("WidthScreen", scr.Width); //Chiều dài màn hình
        Output.Add("formStt", formStt); //Sửa or thêm
    }
}

```

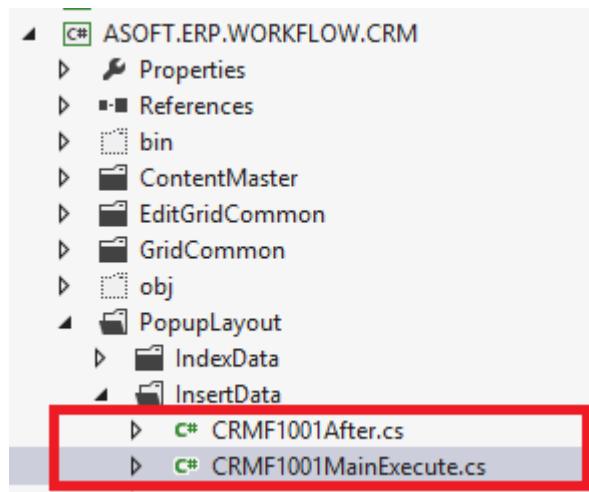
Insert dữ liệu

- Trường hợp 1: Insert danh mục (trường sysCategoryBusinessID trong bảng sysScreen bằng 1)
 - o **Insert danh mục** dữ liệu trên lối sử dụng luồng workflow **InsertData**
 - o **InsertData** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động bao gồm:
 - **Before**: trước khi insert sẽ thực hiện gì.
 - **BeforeValidation**: check validation tại server.
 - **BeforeCheckUnique**: check đã tồn tại.
 - **BeforeCheckCommon**: các loại check khác đặc thù của nghiệp vụ (mặc định: check này luôn luôn đúng).
 - **MainExecute**: thực hiện insert dữ liệu.
 - **After**: sau khi insert sẽ thực hiện gì.



- o Sẽ có nghiệp vụ insert danh mục không chạy theo những hành động mặc định này. Vậy nên chúng ta phải Custom như sau:
 - o VD: Custom hành động **MainExecute**.
 - **Cách tạo:**
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)

- Tạo thư mục **PopupLayout** (nếu chưa có)
- Tạo thư mục **InsertData** nằm trong thư mục **PopupLayout** (nếu chưa có)
- Tạo file class **{Mã màn hình nhập liệu} +MainExecute.**



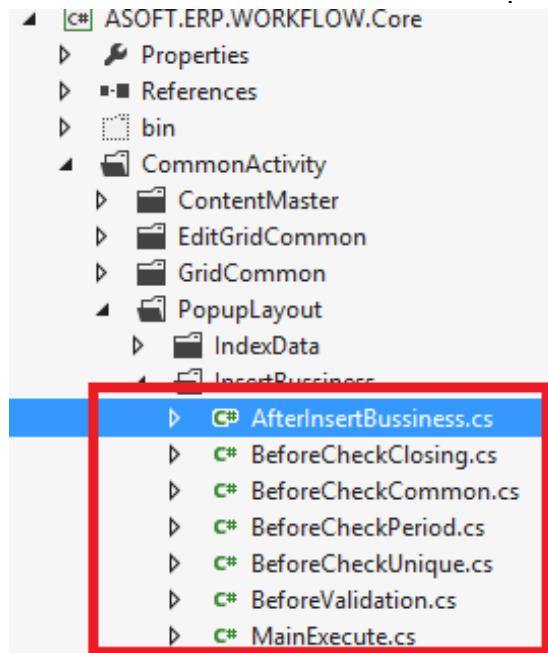
- Cấu trúc của class custom:

```
- public sealed class CRMF1001MainExecute : CodeActivity
- {
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

-     // If your activity returns a value, derive from CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
-     {
-         // Code ở đây, tham khảo class khác
-     }
- }
```

- Tương tự với những hành động khác: **{ Mã màn hình nhập liệu } + BeforeCheckUnique, : { Mã màn hình nhập liệu } + BeforeCheckCommon...**
- Trường hợp 2: Insert nghiệp vụ (trường sysCategoryBusinessID trong bảng sysScreen bằng 2)
 - **Insert nghiệp vụ** dữ liệu trên lối sử dụng luồng workflow **InsertBussiness**
 - **InsertBussiness** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động bao gồm:
 - **BeforeValidation**: check validation tại server.
 - **BeforeCheckClosing**: check khóa sổ kì kế toán.
 - **BeforeCheckPeriod**: check ngày hạch toán có hợp lệ.
 - **BeforeCheckUnique**: check đã tồn tại.

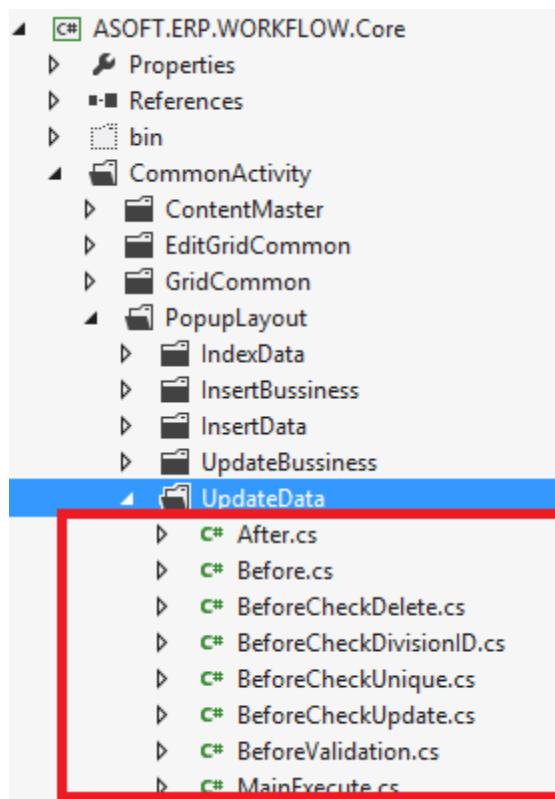
- **BeforeCheckCommon:** các loại check khác đặc thù của nghiệp vụ (mặc định: check này luôn luôn đúng).
- **MainExecute:** thực hiện insert dữ liệu.
- **AfterInsertBussiness:** sau khi insert sẽ thực hiện gì.



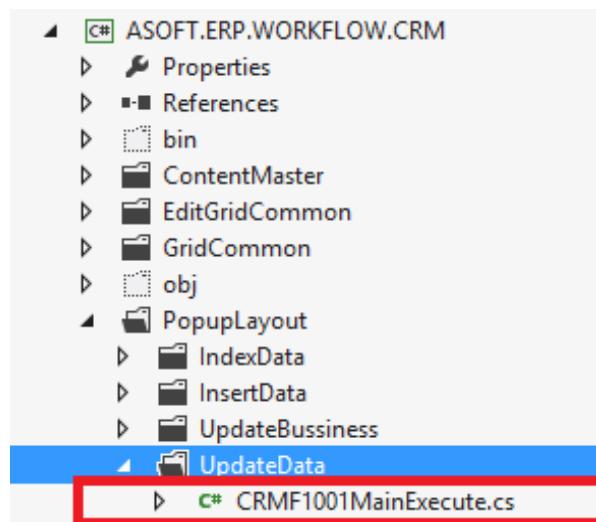
- Cũng tương tự với insert danh mục, cách custom và cấu trúc custom giống với insert danh mục chỉ khác ở chỗ thư mục **InsertData** đổi thành **InsertBussiness**, có thể tham khảo ở trên.

Update Dữ liệu

- Trường hợp 1: Update danh mục (trường sysCategoryBusinessID trong bảng sysScreen bảng 1)
 - **Update danh mục** dữ liệu trên lưới sử dụng luồng workflow **UpdateData**
 - **UpdateData** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động bao gồm:
 - **Before:** trước khi update làm gì
 - **BeforeValidation:** check validation tại server.
 - **BeforeCheckDelete:** check dữ liệu đã bị xóa.
 - **BeforeCheckDivisionID:** check dữ liệu thuộc division hiện tại.
 - **BeforeCheckUnique:** check đã tồn tại.
 - **BeforeCheckUpdate:** check dữ liệu đã bị chỉnh sửa.
 - **MainExecute:** thực hiện insert dữ liệu.
 - **After:** sau khi update sẽ thực hiện gì.



- Sẽ có nghiệp vụ update danh mục không chạy theo những hành động mặc định này. Vậy nên chúng ta phải Custom như sau:
- VD: Custom hành động **MainExecute**.
 - **Cách tạo:**
 - Tao project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **PoupLayout** (nếu chưa có)
 - Tạo thư mục **UpdateData** nằm trong thư mục **PoupLayout** (nếu chưa có)
 - Tạo file class **{Mã màn hình nhập liệu} +MainExecute**.

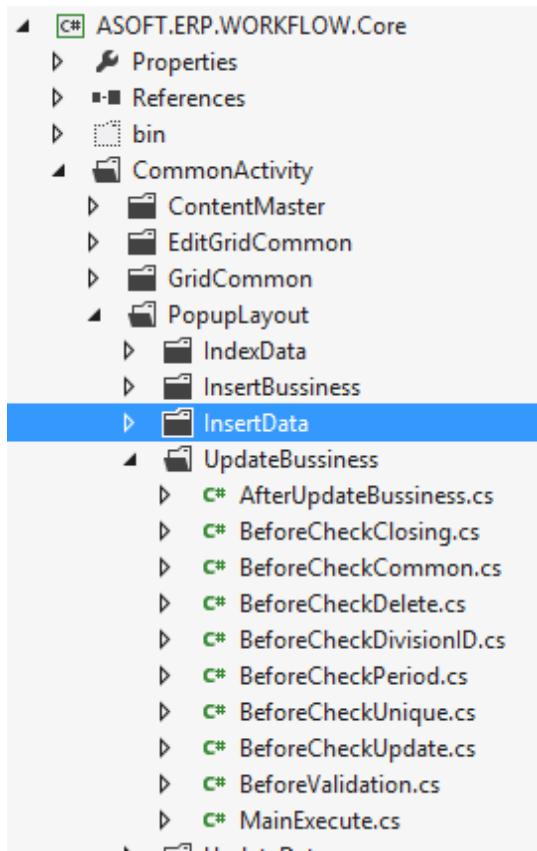


- Cấu trúc của class custom:

```
- public sealed class CRMF1001MainExecute : CodeActivity
- {
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

-     // If your activity returns a value, derive from CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
-     {
-         // Code ở đây, tham khảo class khác
-     }
- }
```

- Trường hợp 2: Update nghiệp vụ (trường sysCategoryBusinessID trong bảng sysScreen bằng 2)
 - Insert nghiệp vụ (trường sysCategoryBusinessID trong bảng sysScreen bằng 2)
 - **Update nghiệp vụ** dữ liệu trên luồng sử dụng luồng workflow **UpdateBussiness**
 - **UpdateBussiness** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động bao gồm:
 - **BeforeValidation**: check validation tại server.
 - **BeforeCheckClosing**: check khóa sổ kì kế toán.
 - **BeforeCheckPeriod**: check ngày hạch toán có hợp lệ.
 - **BeforeCheckDivisionID**: check dữ liệu thuộc division hiện tại.
 - **BeforeCheckDelete**: check dữ liệu đã bị xóa.
 - **BeforeCheckUpdate**: check dữ liệu đã bị chỉnh sửa.
 - **BeforeCheckUnique**: check đã tồn tại.
 - **BeforeCheckCommon**: các loại check khác đặc thù của nghiệp vụ (mặc định: check này luôn luôn đúng).
 - **MainExecute**: thực hiện insert dữ liệu.
 - **AfterUpdateBussiness**: sau khi update sẽ thực hiện gì.



- Cũng tương tự với Update danh mục, cách custom và cấu trúc custom giống với insert danh mục chỉ khác ở chỗ thư mục **UpdateData** đổi thành **UpdateBussiness**, có thể tham khảo ở trên.

c) Validation trên màn hình nhập liệu

- Validation được thực hiện tại actitity BeforeValidation của WorkFlow Insert.
- Đối với maxLenght và minLenght của chuỗi, validate theo kiểu dữ liệu của Field
 - + Khóa chính kiểu chuỗi → max 50
 - Trong Script tạo Fields TextBox(`set @sysTypeID = 3`)
 - + Chuỗi kí tự → max 250
 - Trong Script tạo Fields TextBox(`set @sysTypeID = 7`)
- Đối với minValue và maxValue của số nguyên: sử dụng 2 trường minValue và maxValue trong sysField để validate
 - + `set @MinValue = 1`
 - + `set @MaxValue = 100`
- Đối với regex ta dùng bảng sysRegular để định nghĩa pattern và message, sau đó tạo một field sử dụng regex.
- Thêm một regex mới.

Pattern	Biểu thức điều kiện
Message	Mã thông báo khi có lỗi

Description	Mô tả cho regex
-------------	-----------------

Sử dụng script

```
declare @Pattern varchar(100)
declare @Message nvarchar(250)
declare @Description nvarchar(250)

set @Pattern = ''
set @Message = ''
set @Description = ''

insert into sysRegular(Pattern,Message,Description)
values(@Pattern, @Message, @Description)
```

Thêm field tương tự textbox(`set @sys sysRegularID = 3`)

d) Hàm mở rộng onAfterInsertSuccess

- Sau khi insert thành công dữ liệu, mặc định sẽ xử lý theo action đã chọn(nếu là lưu nhập tiếp sẽ xóa trắng field, nếu lưu sao chép sẽ giữ lại field)
- Nếu muốn tạo thêm xử lý sau khi insert(ví dụ load lại mã tăng tự động sau khi insert) ta định nghĩa thêm hàm onAfterInsertSuccess trong javascriptCustom

```
function onAfterInsertSuccess(result, action)
{
    if (action == 1 || action == 2) {
        GetAccountID();
    }
}
```

Đối với PopupMasterDetail

- Action 1: Thêm nhập tiếp
 - Action 2: Thêm sao chép
 - Action 3: Lưu đóng
 - Action 4: Lưu(update)
- Đối với PopupLayout
- Action 1: Thêm nhập tiếp
 - Action 2: Thêm sao chép
 - Action 3: Lưu(update)

3) Màn hình nhập liệu có lưới detail

CẬP NHẬT PHIẾU BÁO GIÁ - SOF2021

Loại chứng từ *	Phiếu tính chênh lệch tỷ giá	Khách hàng
Số chứng từ *	CL/06/2017/0004	Địa chỉ
Ngày chứng từ *	07/06/2017	Điện thoại
Ngày hết hạn	07/06/2017	Địa chỉ giao hàng
Tình trạng phiếu	Chưa chấp nhận	Phương tiện vận chuyển
Loại tiền	Tiền Việt Nam	Điều khoản thanh toán
Tỷ giá	1	Phương thức thanh toán
Loại mặt hàng		Bảng giá
Người báo giá	Quang hoàng	Điễn giải
Nhân viên bán hàng	Quang hoàng	Điều kiện thanh toán
Tham số		Kính gửi
		Chú ý
		Nội dung

STT	Mã hàng	Tên mặt hàng	Đơn vị tính	Số lượng	Đơn giá	Thành tiền	Qui đổi	Nhóm thuế	% thuế VAT
1									

Lưu & nhập tiếp Lưu & sao chép Đóng

a) Các bước bắt buộc

Bước 1: Tạo bảng cho màn hình

- Thêm bảng cho phần master
 - o Nếu bảng có rồi thì không cần tạo lại.
 - o Nếu chưa có, thực hiện thêm 1 dòng vào bảng sysTable:

sysTable	
TableName	Tên bảng (VD: OT2101)
Description	Chú thích về bảng
PK	Khóa chính của bảng (VD: APK)
ModuleID	Module của bảng (VD: AsoftSO)
ParentTable	NULL
RefLink	Trường của bảng để click link trên lưới (Dùng trên lưới, không có để NULL)
RefScreenMainID	Mã màn hình nhập liệu của bảng.

- o Script: lấy tạo bảng màn hình nhập liệu không lưới detail

- Thêm bảng cho phần detail (phần lưới)

sysTable	
TableName	Tên bảng (VD: OT2102)
Description	Chú thích về bảng
PK	Khóa chính của bảng (VD: APK)
ModuleID	Module của bảng (VD: AsoftSO)
ParentTable	Table Master (VD: OT2101)
RefLink	Trường của bảng để click link trên lưới (Dùng trên lưới, không có để NULL)
RefScreenMainID	Mã màn hình nhập liệu của bảng.

- o **NOTE:** Có bao nhiêu bảng detail có parent là bảng master sẽ có bao nhiêu lưới detail.

Bước 2: Tạo màn hình

Thêm 1 dòng vào bảng sysScreen

sysScreen	
APK	Tự sinh động
DivisionID	Đơn vị (VD: AS)
ModuleID	Module của bảng (VD: AsoftSO)
ScreenID	ID Màn hình (VD: SOF2021)
ScreenName	Tên màn hình (VD: Cập nhật báo giá)
ScreenType	Loại màn hình (VD: 3: Màn hình nhập liệu)
ScreenNameE	Tên tiếng anh của màn hình
Parent	Màn hình cha (VD: SOF2020 Màn hình tìm kiếm)
sysTable	Tham chiếu tới bảng sysTable, bảng mà màn hình thuộc vào (VD: OT2101(phải là bảng của master))
Title	Title của màn hình (Mã ngôn ngữ để lấy ra tên màn hình VD: SOF2021.Title)
TypeInput	VD: 2: Loại màn hình nhập liệu có lưới detail
sysCategoryBusinessID	1: Danh mục, 2:Nghiệp Vụ
Width	Chiều rộng của lưới (VD: 800)

- Sử dụng script

```

declare @DivisionID nvarchar(50)
declare @ModuleID nvarchar(50)
declare @ScreenID nvarchar(50)
declare @ScreenName nvarchar(250)
declare @ScreenType tinyint
declare @ScreenNameE nvarchar(250)
declare @Parent nvarchar(50)
declare @sysTable nvarchar(50)
declare @Title nvarchar(50)
declare @TypeInput varchar(100)
declare @sysCategoryBusinessID varchar(100)

```

```

declare @Width varchar(100)

set @DivisionID = ''
set @ModuleID = ''
set @ScreenID = ''
set @ScreenName = ''
set @ScreenType = 3
set @ScreenNameE = ''
set @Parent = ''
set @sysTable = ''
set @title = ''
set @TypeInput = ''
set @sysCategoryBusinessID = ''
set @Width = ''

insert into sysScreen(DivisionID, ModuleID, ScreenID, ScreenName, ScreenType,
ScreenNameE, Parent,
        sysTable, Title, TypeInput, sysCategoryBusinessID, Width)
values(@DivisionID,@ModuleID,@ScreenID,@ScreenName,@ScreenType,@ScreenNameE,@Par
ent,
        @sysTable,@Title,@TypeInput,@sysCategoryBusinessID,@Width)

```

Bước 3: Tạo các control cho màn hình (Master)

- Thêm dòng vào bảng sysField, số lượng field thêm vào bảng số lượng control trên màn hình, cách thêm giống như thêm textbox, checkbox, combobox.....

Bước 4: Tạo các column trên lưới detail (Detail)

- Thêm dòng vào bảng sysField với số lượng tương ứng với số lượng cột trên màn hình.
- Trong tất cả các fields thuộc bảng detail sẽ có 1 fields(bảng con) liên quan đến 1 fields trong bảng master(bảng cha). VD: Field APKMaster (Bảng con) liên quan đến field APK(Bảng cha).
Ta thực hiện Update trường RefTable thuộc fields APKMaster trong bảng sysFields:

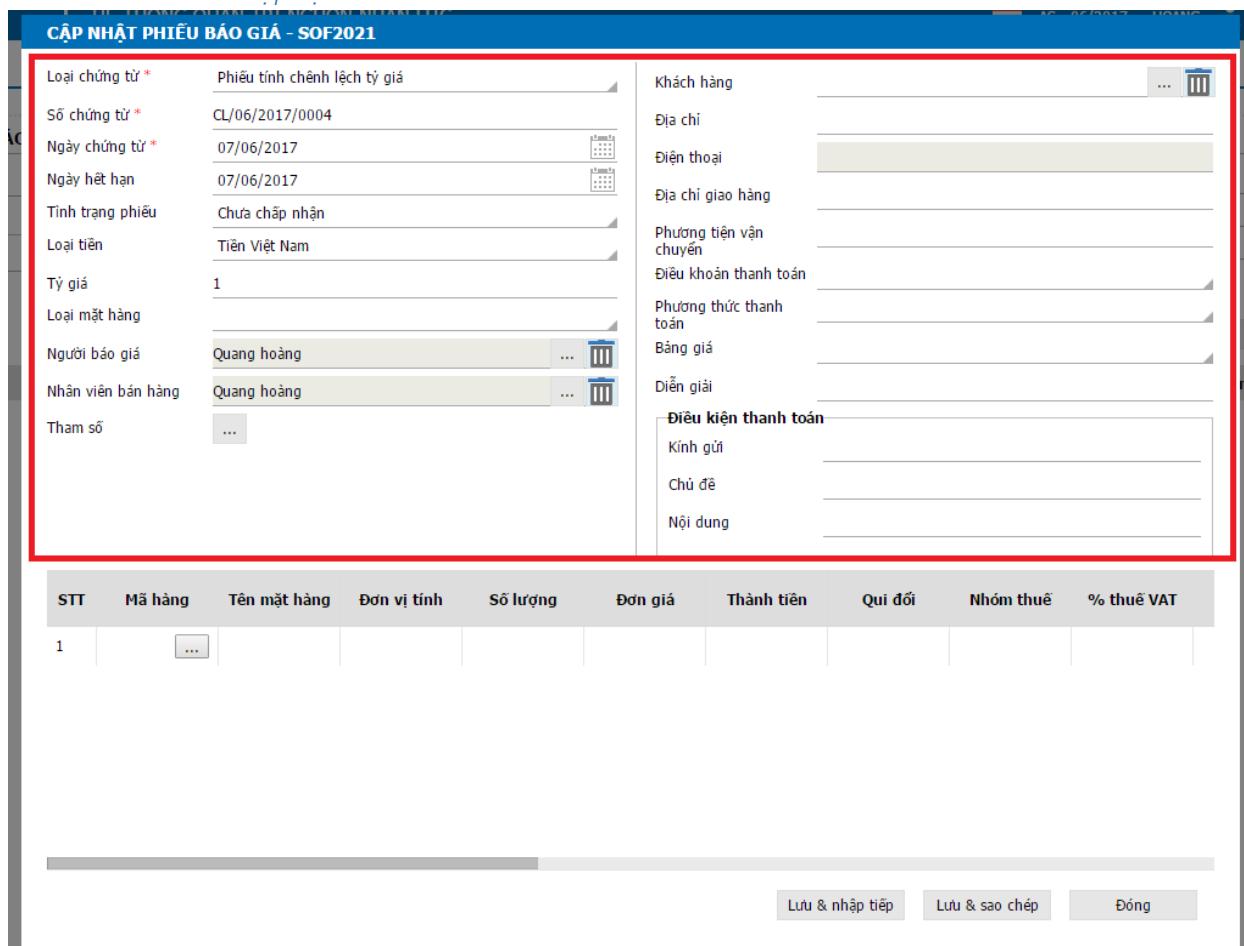
SCRIPT: `Update sysFields set RefTable = {Table bảng master/bảng cha} where ColumnName = 'Field liên quan của bảng detail' and sysTable = 'Table bảng detail/bảng con'.`

VD: Cột trong bảng **OT2102**(bảng con) là APKMaster liên quan đến cột trong bảng **OT2101**(bảng cha) là APK. Ta thực hiện update như hướng dẫn.

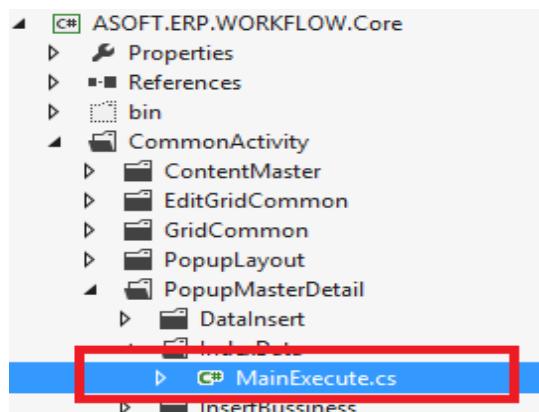
Update sysFields set RefTable = 'OT2101' where ColumnName= 'APK' and sysTable = 'OT2102'

b) Customize

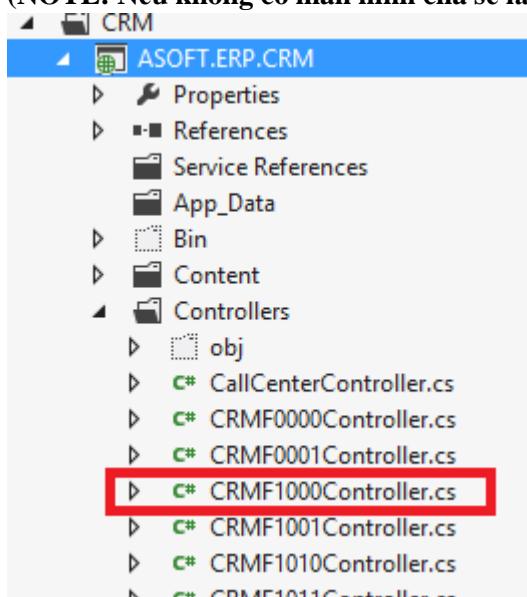
Cấu trúc màn hình nhập liệu có lưới detail



- Cấu trúc màn hình tìm kiếm sử dụng luồng workflow **IndexData**:
- **IndexData** nằm trong project **ASOFT.ERP.WORKFLOW.Core** -> **PopupMasterDetail**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để cấu trúc lưới.



- Số lượng control search sẽ phụ thuộc vào số lượng field của bảng mà màn hình thuộc vào, những cột hiển thị phụ thuộc vào trường **PopupVisible** trong bảng **sysFields** trong **DB Admin**, bảng **0** thì ẩn, **1** thì hiển thị.
- Có 2 trường hợp custom:
 - o Trường hợp 1, custom 1 hoặc nhiều control trong màn hình tìm kiếm
 - Cách custom:
 - Tạo file class **{Mã màn hình cha của màn hình nhập liệu} + “Controller”** trong project **ASOFT.ERP.{Module}** -> **Controllers**.
(NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)

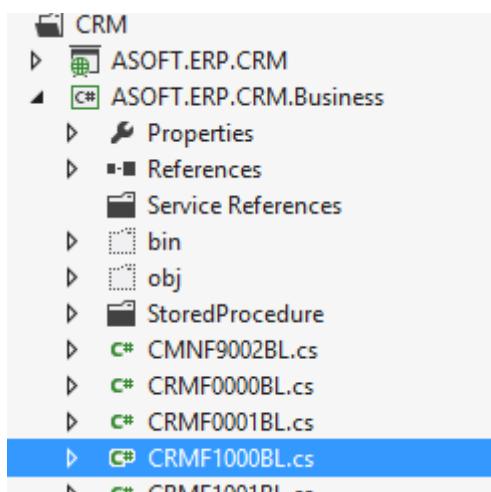


- Thêm action **“ChangeListFieldsPopupMaster”** vào controller.
- Trong action vừa thêm: code theo cấu trúc:

```

- public List<ASOFTSysFields> ChangeListFieldsPopupMaster(List<ASOFTSysFields> list,
- string id)
- {
-     //Code ở đây, tham khảo các file class khác
-     return list;
- }
```

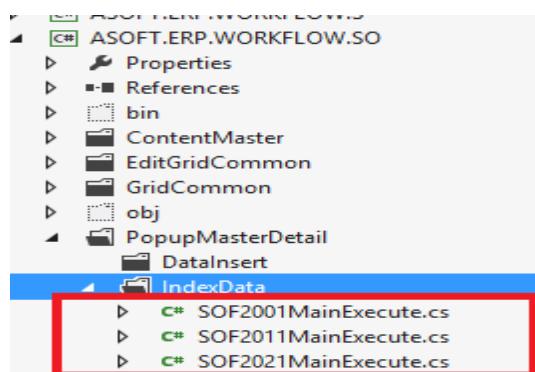
- o Trường hợp 2: custom câu SQL load màn hình nhập liệu (trường hợp sửa)
 - Cách custom:
 - Tạo file class **{Mã màn hình cha của màn hình nhập liệu} + “BL”** trong project **ASOFT.ERP.{Module}.Business** (Nếu chưa có file)
(NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)



- Thêm hàm “**GetDataPopupMaster**” vào class.
- Trong class vừa thêm, code theo cấu trúc:

```
public Dictionary<string, object> GetDataPopupMaster(string APK, string id)
{
    try
    {
        //Code tại đây
    }
    catch (ASOFTException)
    {
        throw;
    }
    catch (Exception ex)
    {
        throw new ASOFTException(ASOFTLayer.Business, ex.Message, ex);
    }
}
```

- Trường hợp 3: custom hoàn toàn màn hình.
 - Cách custom:
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)
 - Tạo thư mục **PopupMasterDetail** (nếu chưa có)
 - Tạo thư mục **IndexData** nằm trong thư mục **PopupMasterDetail** (nếu chưa có)
 - Tạo file class **{Mã màn hình nhập liệu} + MainExecute**



- Cấu trúc của class custom

```

- public sealed class SOF2021MainExecute : CodeActivity
- {
-     // Define an activity input argument of type string
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }
-
-     // If your activity returns a value, derive from CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
-     {
-         // Obtain the runtime value of the Text input argument
-         ASOFTGridBL b11 = new ASOFTGridBL();
-         // Obtain the runtime value of the Text input argument
-         string id = ScreenID.Get(context).ToString();
-         string area = Area.Get(context).ToString();
-         string key = InOut.Get(context)[ "key" ] != null ?
-             InOut.Get(context)[ "key" ].ToString() : null;
-         string PK = InOut.Get(context)[ "PK" ] != null ?
-             InOut.Get(context)[ "PK" ].ToString() : null;
-         string Table = InOut.Get(context)[ "Table" ] != null ?
-             InOut.Get(context)[ "Table" ].ToString() : null;
-         string divisionID = InOut.Get(context)[ "DivisionID" ].ToString();
-         HttpRequestBase Request = InOut.Get(context)[ "Request" ] != null ?
-             (HttpRequestBase)InOut.Get(context)[ "Request" ] : null;
-
-
-         List<ASOFTSysFieldsViewModel> list1 = new List<ASOFTSysFieldsViewModel>();
-         List<ASOFTSysTable> listtb = new List<ASOFTSysTable>();
-         List<ASOFTSysTableViewModel> listtb1 = new List<ASOFTSysTableViewModel>();
-         ASOFTSysScreen scr = new ASOFTSysScreen();
-         List<ASOFTSysFields> list = new List<ASOFTSysFields>();
-         List<ASOFTSysFieldsViewModel> listPK1 = new
-             List<ASOFTSysFieldsViewModel>();
-         List<ASOFTSysFields> listPK = new List<ASOFTSysFields>();
-         Dictionary<string, object> data = new Dictionary<string, object>();
-         object getdata = new object();
-         var formStt = ASOFTFormActionType.AddNew;
-
-         //Code tại đây, tham khảo các class khác
-
```

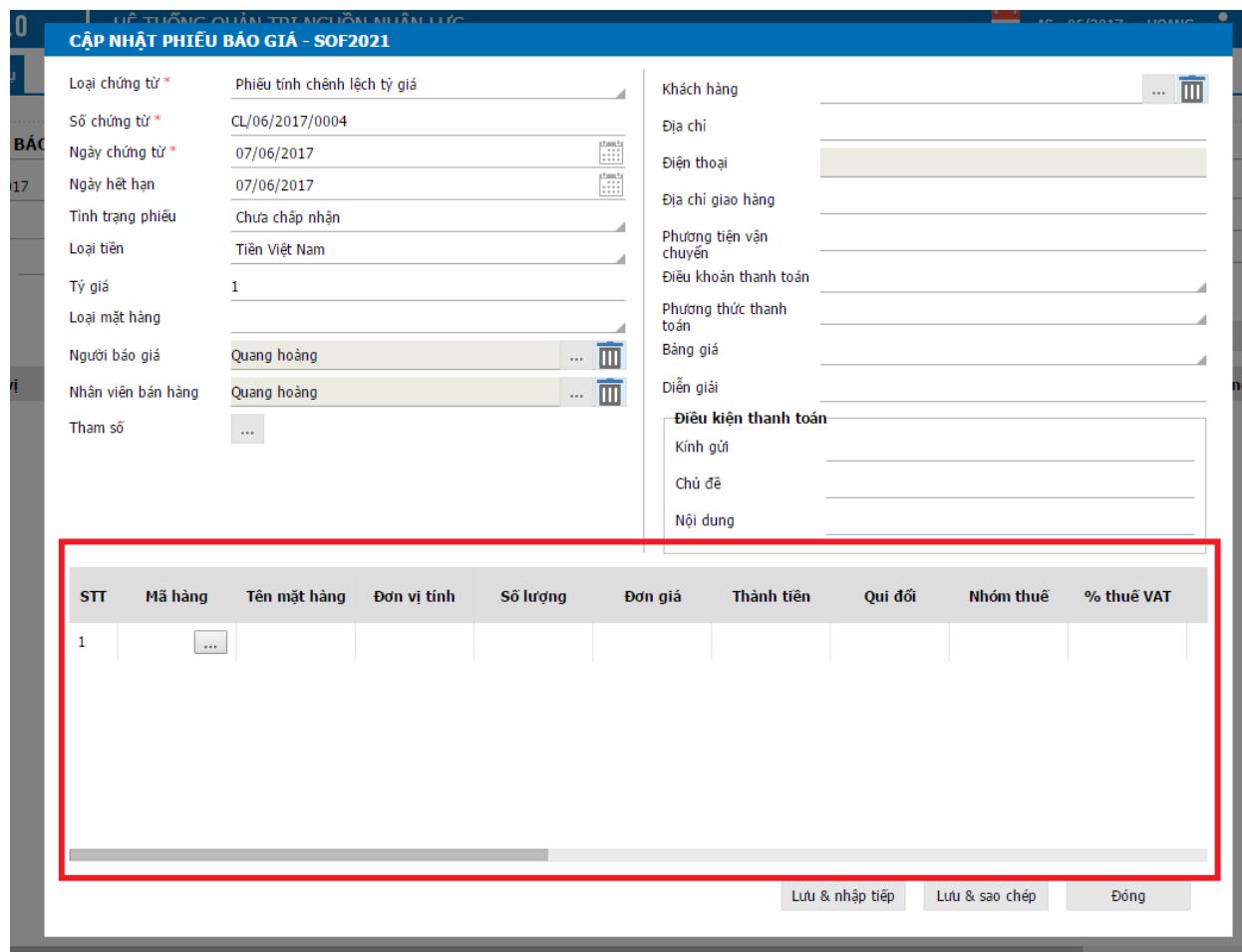
```

        - Dictionary<string, Object> Output = InOut.Get(context);
        -
        - Output.Add("Title", scr.TitleName + " - " + scr.ScreenID);
        - Output.Add("List", list1);
        - Output.Add("Listtb", listtb1);
        - Output.Add("ListPK", listPK1);
        - Output.Add("WidthScreen", scr.Width);
        - Output.Add("formStt", formStt);
        -
        }
    }
}

```

Cấu trúc lưới detail

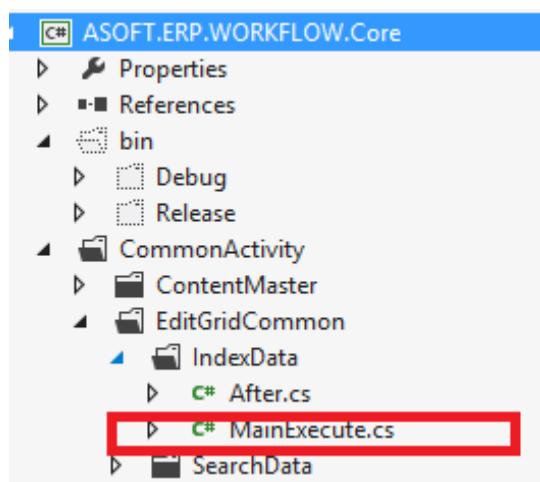
Lưới không group:



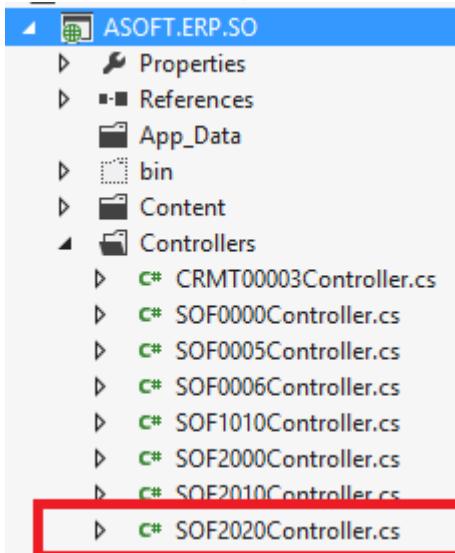
The screenshot shows a software interface for updating a price list (Phiếu báo giá). The main form contains various input fields for document details like title, date, and currency rate. On the right side, there are several empty text input fields for customer information (Khách hàng, Địa chỉ, Điện thoại, Địa chỉ giao hàng, Phương tiện vận chuyển, Điều khoản thanh toán, Phương thức thanh toán, Bảng giá, Diễn giải). Below these is a section for payment terms (Điều kiện thanh toán) with fields for Kinh gửi, Chủ đề, and Nội dung. At the bottom, a large red box highlights a detail grid table with columns: STT, Mã hàng, Tên mặt hàng, Đơn vị tính, Số lượng, Đơn giá, Thành tiền, Qui đổi, Nhóm thuế, and % thuế VAT. The first row of the grid has a value of 1.

STT	Mã hàng	Tên mặt hàng	Đơn vị tính	Số lượng	Đơn giá	Thành tiền	Qui đổi	Nhóm thuế	% thuế VAT
1									

- Cấu trúc lưới sử dụng luồng workflow **IndexData**:
- **IndexData** nằm trong project **ASOFT.ERP.WORKFLOW.Core** -> **EditGridCommon**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để cấu trúc lưới.



- Custom lưới:
 - o Trường hợp 1: custom cột trên lưới
 - Cách custom:
 - Tạo file class {Mã màn hình cha của màn hình nhập liệu} + “Controller” trong project ASOFT.ERP.{Module} -> Controllers.
(NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)



- Thêm action “ChangeFieldsGridEdit” + (Mã màn hình nhập liệu) + “_” + (Mã table của lưới detail màn hình nhập liệu) vào controller.
- Trong action vừa thêm: code theo cấu trúc:

```

- public List<ASOFTSysFields> ChangeFieldsGridEditSOF2021_OT2102(List<ASOFTSysFields>
-     list, string id)
- {
-     //Code ở đây, tham khảo các file class khác
-     return list;
- }
```

- o Trường hợp 2: custom toàn bộ lưới:
 - Cách custom:

- Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)
- Tạo thư mục **EditGridCommon** (nếu chưa có)
- Tạo thư mục **IndexData** nằm trong thư mục **EditGridCommon** (nếu chưa có)
- Tạo file class **{Mã màn hình nhập liệu} + MainExecute**
- Cấu trúc của class custom

```

- public sealed class SOF2021MainExecute : CodeActivity
- {
-     // Define an activity input argument of type string
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

-     // If your activity returns a value, derive from CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
-     {
-         // Obtain the runtime value of the Text input argument
-         string id = ScreenID.Get(context);
-         string module = Area.Get(context);
-         Dictionary<string, Object> Input = InOut.Get(context);
-         List<ASOFTSysFields> list = new List<ASOFTSysFields>();
-         ASOFTSysScreen scr = new ASOFTSysScreen();
-         List<string> list1 = new List<string>();
-         string table = Input["table"].ToString();

-         //Code ở đây, thám khảo các class đã làm
-         //Các biến bắt buộc phải trả về
-         Input.Add("sysFields", list); //List cột trên lưới
-         Input.Add("Language", list1); //Ngôn ngữ tương ứng của các cột
-         Input.Add("scr", scr); //Màn hình tìm kiếm
-     }
- }
```

Lưới group:

CẬP NHẬT CÁ NHÂN TỰ ĐÁNH GIÁ - KPIF2001

Tên nhân viên *	Hoàng Vũ	Điểm mạnh
Đợt đánh giá *	Đánh giá KPI quý IV/2017	Điểm yếu
Ngày bắt đầu *	01/10/2017	Ý kiến nhân viên
Ngày kết thúc *	31/12/2017	Đề xuất sau đánh giá
Bảng đánh giá từng vị trí *	Bộ đánh giá KPI năm 2017 phòng Sof	
Phòng ban	PHÒNG KỸ THUẬT	
Chức vụ	PHÒNG BAN CHỨC NĂNG	
Chức danh/vị trí	CHUYÊN VIÊN QA	
Người đánh giá *	Hoàng Vũ	

STT	Tên chỉ tiêu	Đơn vị tính	Công thức tính	Tần suất do	Nguồn do	Loại	Tỷ trọng
AS001 - TÀI CHÍNH (F)							
1	Chi tiêu MD sạch	MD	Tổng số MD thực hiện theo kế hoạch (không tính fixbug)	Quý	Dữ liệu trên Odoo	Càng lớn càng tốt	60
2	Tổng số MD hỗ trợ	MD	Tổng số MD thực tế nhân viên hỗ trợ công việc ngoài dự án	Quý	Dữ liệu trên Odoo	Càng lớn càng tốt	40
AS002 - KHÁCH HÀNG (C)							
3	Đảm bảo chất lượng (kiểm soát số bugs tối đa)	Bugs	[Tổng số bug trong tất cả dự án] / [Tổng số MM]	Quý	Dữ liệu trên Odoo	Càng nhỏ càng tốt	30
4	Kiểm soát thời gian xử lý bugs	%	[Tổng thời gian xử lý bugs]/[Tổng số MM đạt được] * 100	Quý	Dữ liệu trên Odoo	Càng nhỏ càng tốt	15
5	Ngăn ngừa, phát hiện lỗi	Bugs	[Tổng số bug bắt được trong tất cả dự án] / [Tổng số MM]	Quý	Dữ liệu trên Odoo	Càng lớn càng tốt	15
6	Độ tin cậy ứng dụng	%	Tổng % chinh lich tiendoi doi trong tất cả các dự án	Quý	Dữ liệu trên Odoo	Càng lớn càng	40
Tổng điểm: <input type="text" value="17.00"/> Xếp loại: <input type="text" value="C"/>							
<input type="button" value="Lưu"/> <input type="button" value="Đóng"/>							

- Group lướt:
 - o Khai báo template cho trường GroupOnGrid trong bảng sysFields.

- VD:

- #### ○ Script:

```
update sysFields set GroupOnGrid = '{Template của group}' where FieldID = '{ID của field muốn group}'
```

- Tổng trên lưới (tính tổng của 1 cột thuộc group):

- Khai báo template cho trường SumOnGrid trong bảng sysFields.
 - Script:

- update

Sinh trên lưới (tính trung bình của 1 cột thuộc group):

- Trung bình trên lưới (tính trung bình của 1 cột thuộc group):
 - o Khai báo template cho trường AverageOnGrid trong

- Khai báo template cho trường AverageOnGrid trong bảng sysFields.
 - Script:

- Script:

```
update sysFields set AverageOnGrid = '{template của Average}' where FieldID = '{ID của field muốn Average}'
```

- Max trên lưới (giá trị lớn nhất của 1 cột thuộc group):
Khái niệm: $\max_{\text{group}}(M)$ - Giá trị lớn nhất

- Khai báo template cho trường MaxOnGrid trong bảng sysFields.

- ### ○ Script:

```
update sysFields set MaxOnGrid = '{Template của Max}' where FieldID = '{ID  
của field muốn Max}'
```

- Min trên lưới (giá trị nhỏ nhất của 1 cột thuộc group):

- Khai báo template cho trường MaxOnGrid trong bảng sysFields.

- #### ○ Script:

```
update sysFields set MinOnGrid = '{Template của Min}' where FieldID = '{ID  
của field muốn Min}'
```

- Count trên lưới (tính tổng số lượng của 1 cột thuộc group):

- Khai báo template cho trường MaxOnGrid trong bảng sysFields.

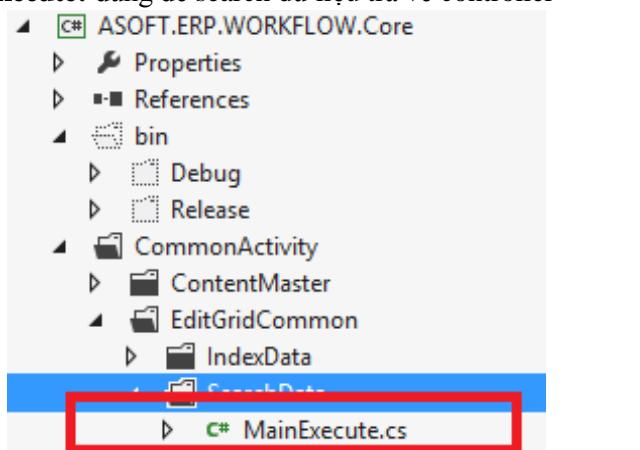
- ## o Script:

```
update sysFields set CountOnGrid = '{Template của Count}' where FieldID = '{ID của field muốn Count}'
```

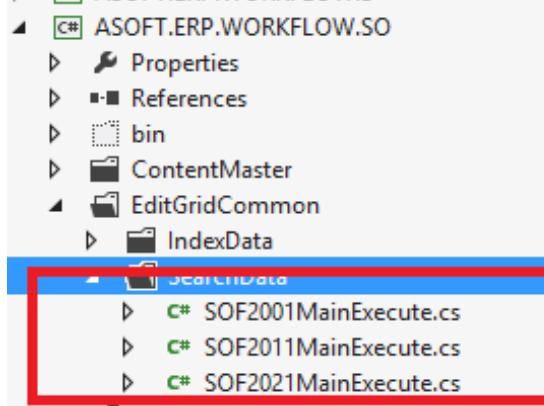
- Custom lưới giống với lưới không group.

Load lưới detail

- Load lưới detail sử dụng luồng workflow **SearchData**:
 - o **SearchData** nằm trong project **ASOFT.ERP.WORKFLOW.Core** -> **EditGridCommon**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để search dữ liệu trả về controller



- o Nếu nghiệp vụ search yêu cầu sử dụng store và câu SQL đặc thù thì ta sẽ custom:
- o VD:
 - Cách tạo:
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)
 - Tạo thư mục **EditGridCommon** (nếu chưa có)
 - Tạo thư mục **SearchData** nằm trong thư mục **EditGridCommon** (nếu chưa có)
 - Tạo file class **{Mã màn hình tìm kiếm} + MainExecute**



- Cấu trúc của class custom

```
public sealed class SOF2021MainExecute : CodeActivity
{
    // Define an activity input argument of type string
    public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
    public InOutArgument<string> Area { get; set; }
    public InOutArgument<string> ScreenID { get; set; }
```

```

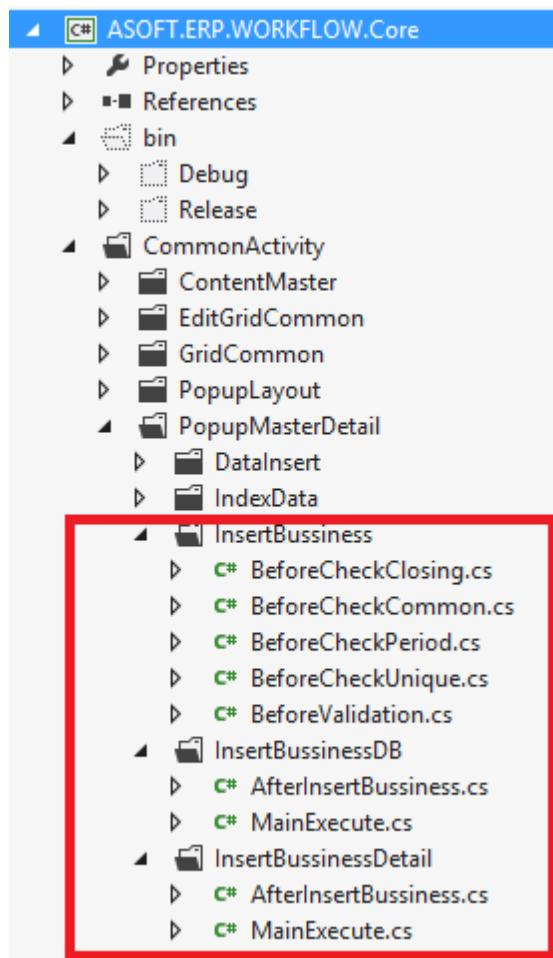
// If your activity returns a value, derive from CodeActivity<TResult>
// and return the value from the Execute method.
protected override void Execute(CodeActivityContext context)
{
    // Obtain the runtime value of the Text input argument
    // Obtain the runtime value of the Text input argument
    Dictionary<string, List<string>> args = (Dictionary<string,
List<string>>)InOut.Get(context)["args"];
    ASOFTSysGrid ds = new ASOFTSysGrid();
    List<string> list = new List<string>();
    List<string> list1 = new List<string>();

    //Code tại đây, tham khảo các class khác

    Dictionary<string, Object> Ouput = InOut.Get(context);
    Ouput.Add("kq", ds);
}
-
```

Insert dữ liệu

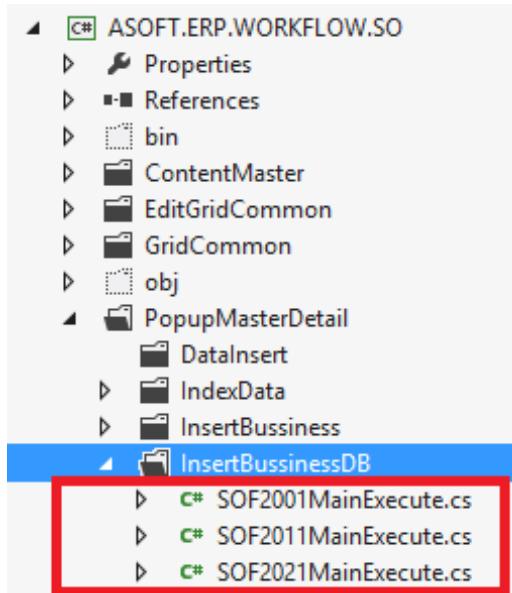
- Insert nghiệp vụ sử dụng 3 luồng workflow(trường sysCategoryBusinessID trong bảng sysScreen bảng 2)
 - o **Insert nghiệp vụ** dữ liệu trên luồng sử dụng luồng workflow bao gồm:
 - **InsertBussiness:** Bao gồm các action kiểm tra dữ liệu:
 - **BeforeValidation:** check validation tại server.
 - **BeforeCheckClosing:** check khóa sổ kì kế toán.
 - **BeforeCheckPeriod:** check ngày hạch toán có hợp lệ.
 - **BeforeCheckUnique:** check đã tồn tại.
 - **BeforeCheckCommon:** các loại check khác đặc thù của nghiệp vụ (mặc định: check này luôn luôn đúng).
 - **InsertBussinessDB:** Bao gồm các action insert dữ liệu vào bảng master:
 - **MainExecute:** thực hiện insert dữ liệu.
 - **AfterInsertBussiness:** sau khi insert sẽ thực hiện gì.
 - **InsertBussinessDetail:** Bao gồm các action insert dữ liệu vào bảng detail:
 - **MainExecute:** thực hiện insert dữ liệu.
 - **AfterInsertBussiness:** sau khi insert sẽ thực hiện gì.



- Sẽ có nghiệp vụ đặc thù của insert nghiệp vụ không chạy theo những hành động mặc định này. Vậy nên chúng ta phải Custom như sau:
- VD: Custom hành động **MainExecute** nằm trang workflow **InsertBussinessDB**
 - **Cách tạo:**
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)
 - Tạo thư mục **PoupMasterDetail** (nếu chưa có)
 - Tạo thư mục **InsertBussinessDB** nằm trong thư mục **PoupMasterDetail** (nếu chưa có).

NOTE: Tên thư mục sẽ phụ thuộc vào tên workflow mà hành động thuộc vào. Trường hợp này là **InsertBussinessDB**.

 - Tạo file class **{Mã màn hình nhập liệu} +MainExecute**.



- Cấu trúc của class custom:

```
- public sealed class SOF2021MainExecute : CodeActivity
{
    public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
    public InOutArgument<string> Area { get; set; }
    public InOutArgument<string> ScreenID { get; set; }

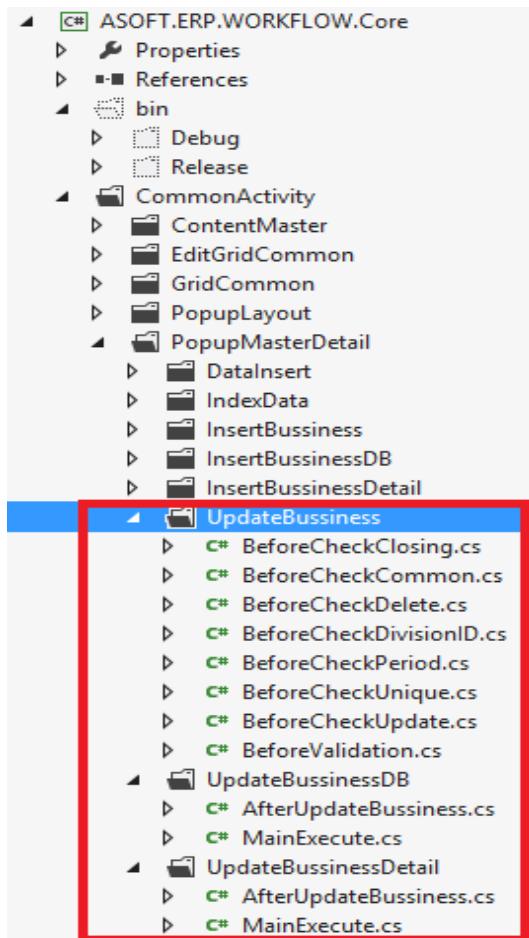
    // If your activity returns a value, derive from CodeActivity<TResult>
    // and return the value from the Execute method.
    protected override void Execute(CodeActivityContext context)
    {
        // Code ở đây, tham khảo class khác
    }
}
```

- Tương tự với những hành động khác: { **Mã màn hình nhập liệu** } + **BeforeCheckUnique**, { **Mã màn hình nhập liệu** } + **BeforeCheckCommon...**

Update dữ liệu

- Update nghiệp vụ sử dụng 3 luồng workflow(trường sysCategoryBusinessID trong bảng sysScreen bằng 2)
 - **Update nghiệp vụ** dữ liệu trên lưới sử dụng luồng workflow bao gồm:
 - **UpdatetBussiness:** Bao gồm các action kiểm tra dữ liệu:
 - **BeforeValidation:** check validation tại server.
 - **BeforeCheckClosing:** check khóa sổ kì kế toán.
 - **BeforeCheckPeriod:** check ngày hạch toán có hợp lệ.
 - **BeforeCheckDivisionID:** check dữ liệu thuộc division hiện tại.
 - **BeforeCheckDelete:** check dữ liệu đã bị xóa.

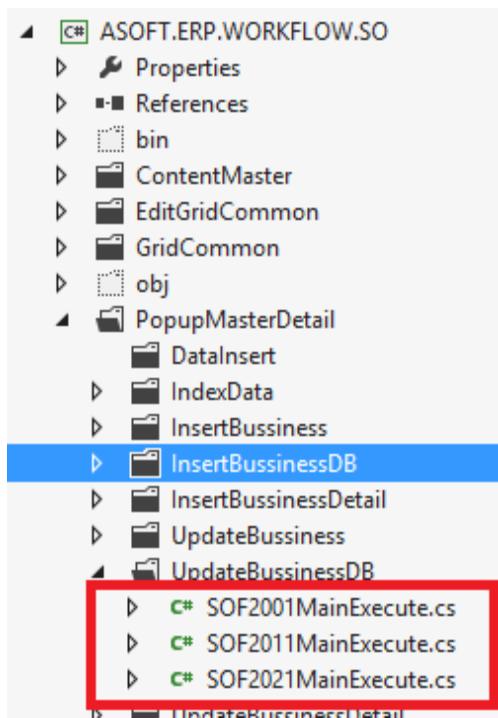
- **BeforeCheckUpdate:** check dữ liệu đã bị chỉnh sửa.
- **BeforeCheckUnique:** check đã tồn tại.
- **BeforeCheckCommon:** các loại check khác đặc thù của nghiệp vụ (mặc định: check này luôn luôn đúng).
- **UpdateBussinessDB:** Bao gồm các action insert dữ liệu vào bảng master:
 - **MainExecute:** thực hiện insert dữ liệu.
 - **AfterUpdateBussiness:** sau khi update sẽ thực hiện gì.
- **UpdateBussinessDetail:** Bao gồm các action insert dữ liệu vào bảng detail:
 - **MainExecute:** thực hiện insert dữ liệu.
 - **AfterUpdateBussiness:** sau khi update sẽ thực hiện gì.



- Sẽ có nghiệp vụ đặc thù của update nghiệp vụ không chạy theo những hành động mặc định này. Vậy nên chúng ta phải Custom như sau:
 - VD: Custom hành động **MainExecute** nằm trang workflow **UpdateBussinessDB**
 - **Cách tạo:**
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **PoupMasterDetail** (nếu chưa có)
 - Tạo thư mục **UpdateBussinessDB** nằm trong thư mục **PoupMasterDetail** (nếu chưa có).

NOTE: Tên thư mục sẽ phụ thuộc vào tên workflow mà hành động thuộc vào. Trường hợp này là **UpdateBussinessDB**.

- Tạo file class {Mã màn hình nhập liệu} +**MainExecute**.



- Cấu trúc của class custom:

```
- public sealed class SOF2021MainExecute : CodeActivity
- {
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

-     // If your activity returns a value, derive from CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
-     {
-         // Code ở đây, tham khảo class khác
-     }
- }
```

- Tương tự với những hành động khác: { Mã màn hình nhập liệu } + **BeforeCheckUnique**, : { Mã màn hình nhập liệu } + **BeforeCheckCommon...**

c) EditorTemplate trên lưới

- Các editor template ví dụ:

Số lượng	Đơn giá	Thành tiền	Qui đổi	Nh
2	990,000	1,980,000	1,980,000	

Dạng format số trên lưới.

STT	Mã hàng	Tên mặt hàng	Đơn vị tính	Số lượng	Đơn giá	Thành tiền	Qui đổi
1	F	Loa vi tính Fenda-A180X	Pc	2	990,000	1,980,000	1,980,000
		Mã mặt hàng	Tên mặt hàng				
	FD-A180X	Loa vi tính Fenda-A180X có Bluetooth + NFC					
	FD-A320	Loa vi tính Fenda-A320					
	FD-A350U	Loa vi tính Fenda-A350U					
	FD-A520U	Loa vi tính 2.1 Fenda-A520U 52W					
	FD-A521	Loa vi tính 2.1 Fenda-A521 52W					

AutoComplete trên lưới.

STT	Mã hàng	Tên mặt hàng	Đơn vị tính	Số lượng	Đơn giá	Thành tiền	Qui đổi
1	F	Loa vi tính Fenda-A180X	Pc	2	990,000	1,980,000	1,980,000
		Mã mặt hàng	Tên mặt hàng				
	FD-A180X	Loa vi tính Fenda-A180X có Bluetooth + NFC					
	FD-A320	Loa vi tính Fenda-A320					
	FD-A350U	Loa vi tính Fenda-A350U					
	FD-A520U	Loa vi tính 2.1 Fenda-A520U 52W					
	FD-A521	Loa vi tính 2.1 Fenda-A521 52W					

ComboBox trên lưới

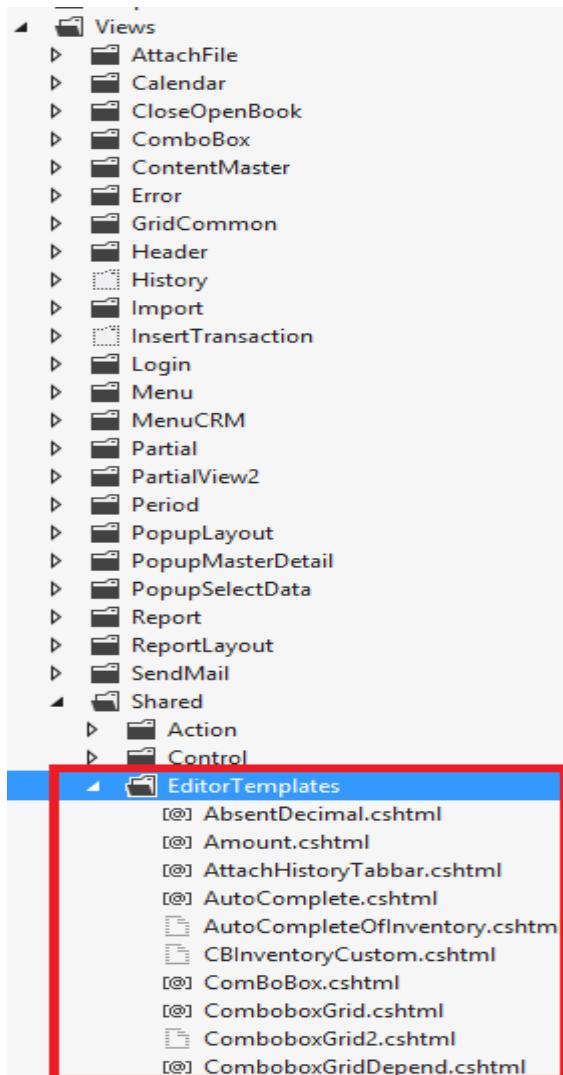
- Cách tạo EditorTemplate
 - o Khai báo trong bảng **sysEditorTemplate**

```

declare @EditorTemplateName nvarchar(MAX) declare @EditorTemplateID nvarchar(MAX)
set @EditorTemplateName=N''
set @EditorTemplateID=N''
If not exists(select top 1 1 from [dbo].[sysEditorTemplate] where [EditorTemplateID]
= @EditorTemplateID)Begin
insert into
sysEditorTemplate(EditorTemplateName,EditorTemplateID)values(@EditorTemplateName,@Edit
orTemplateID)
End

```

- Khai báo editortemplate đã tạo vào fields trên lưới cần sử dụng
 - VD: Fields InventoryID thuộc bảng OT2102 cần sử dụng EditorTemplate ta khai báo EditorTemplate Cho Fields đó.
- `Update sysFields set sysEditorTemplateID = (select sysEditorTemplateID from sysEditorTemplate where EditorTemplate = '(Mã của template được khai báo)'') where ColumnName = '(Tên Fields Chính sửa)' and sysTable = '(Bảng của fields đó)'`
- Tạo file {EditorTemplateID}.cshtml trong thư mục: **Views -> Shared -> EditorTemplates** thuộc Project **ASOFT.ERP**.



4) Màn hình xem chi tiết không có lướt detail (ViewNoDetail)

Danh mục / Loại đổi tương tăng tự động

XEM CHI TIẾT LOẠI MÃ ĐỔI TƯỢNG TĂNG TỰ ĐỘNG - CIF1092



▼ Thông tin loại mã đổi tương tăng tự động

Đơn vị : AS	Mã phân loại : B	Dùng chung : Không
Loại đổi tương : O	Tên phân loại : NGÂN HÀNG	Không hiển thị : Không
Phân loại : 001		

▼ Hệ thống

Người tạo : ASOFTADMIN	Người cập nhật : ASOFTADMIN
Ngày tạo : 19/01/2011 11:04:35	Ngày cập nhật : 19/01/2011 11:04:35

a) Các bước bắt buộc

Bước 1: Tạo Group trong sysGroup

sysGroup	
GroupName	Mã ngôn ngữ Group
Description	Thông tin về Group
ModuleID	Mã phân hệ - Nếu là Group dùng chung thì mã phân hệ là Asoft00
ScreenID	Mã màn hình – Nếu là Group dùng chung thì mã màn hình là A00
TabIndex	Thứ tự hiển thị của Group
GroupID	Mã Group

- Sử dụng script

```

declare @GroupName nvarchar(100)
declare @Description nvarchar(1000)
declare @ModuleID nvarchar(50)
declare @ScreenID nvarchar(50)
declare @TabIndex int

set @GroupName = ''
set @Description = ''
set @ModuleID = ''
set @ScreenID = ''
set @TabIndex = ''
set @GroupID = ''

```

```
insert into sysGroup(GroupName, Description, ModuleID, ScreenID,TabIndex,
GroupID)
values(@GroupName, @Description, @ModuleID, @ScreenID, @TabIndex, @GroupID)
```

Bước 2: Tạo màn hình

Thêm 1 dòng vào bảng sysScreen

sysScreen	
APK	Tự sinh động
DivisionID	Đơn vị (VD: AS)
ModuleID	Module của bảng (VD: AsoftCRM)
ScreenID	ID Màn hình (VD: CRMF2032)
ScreenName	Tên màn hình (VD: Xem chi tiết đầu mối)
ScreenType	Loại màn hình (VD: 5: Màn hình chi tiết)
ScreenNameE	Tên tiếng anh của màn hình
Parent	Màn hình cha (VD: CRMF2030 Màn hình tìm kiếm)
sysTable	Tham chiếu tới bảng sysTable, bảng mà màn hình thuộc vào (VD: CRMT20301)
Title	Title của màn hình (Mã ngôn ngữ để lấy ra tên màn hình VD: CRMF2031.Title)
TypeInput	VD: 4: Loại màn hình xem chi tiết không lướt detail
sysCategoryBusinessID	1: Danh mục (Màn hình này chỉ có danh mục)
DisplayToolBar	ToolBar1 của màn hình (VD: 1,2 (Delete, Add)) thám khảo bảng sysToolBar

- Sử dụng script

```
declare @DivisionID nvarchar(50)
declare @ModuleID nvarchar(50)
declare @ScreenID nvarchar(50)
declare @ScreenName nvarchar(250)
declare @ScreenType tinyint
declare @ScreenNameE nvarchar(250)
declare @Parent nvarchar(50)
declare @sysTable nvarchar(50)
declare @Title nvarchar(50)
declare @TypeInput varchar(100)
declare @sysCategoryBusinessID varchar(100)
declare @Width varchar(100)

set @DivisionID = ''
set @ModuleID = ''
set @ScreenID = ''
set @ScreenName = ''
set @ScreenType = 3
set @ScreenNameE = ''
set @Parent = ''
set @sysTable = ''
set @Title = ''
set @TypeInput = ''
set @sysCategoryBusinessID = ''
```

```

set @DisplayToolBar = ''

insert into sysScreen(DivisionID, ModuleID, ScreenID, ScreenName, ScreenType,
ScreenNameE, Parent,
sysTable, Title, TypeInput, sysCategoryBusinessID, DisplayToolBar)
values(@DivisionID,@ModuleID,@ScreenID,@ScreenName,@ScreenType,@ScreenNameE,@Parent,
@sysTable,@Title,@TypeInput,@sysCategoryBusinessID,@DisplayToolBar)

```

Bước 3: Tạo các Fields hiển thị trong từng Group

- Để fields hiển thị trong Group phải khai báo Mã group cho Fields đó và trường ViewVisible của fields đó phải bằng 1.
- Dùng câu select theo GroupID để lấy khóa tự tăng

```

set @GroupID = (select top 1 convert(varchar(50),sysGroupID) from sysGroup where
GroupID='SOF2002.ThongTinDonHang')

```

```

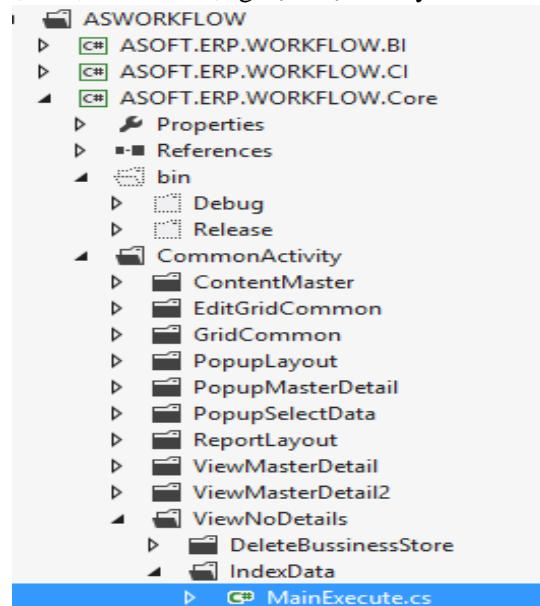
Update sysFields set ViewVisible = 1, GroupID = (select top 1
convert(varchar(50),sysGroupID) from sysGroup where GroupID = '(Mã của group được khai
báo)' ) where ColumnName = '(Tên Fields Chính sửa)' and sysTable = '(Bảng của fields đó)'

```

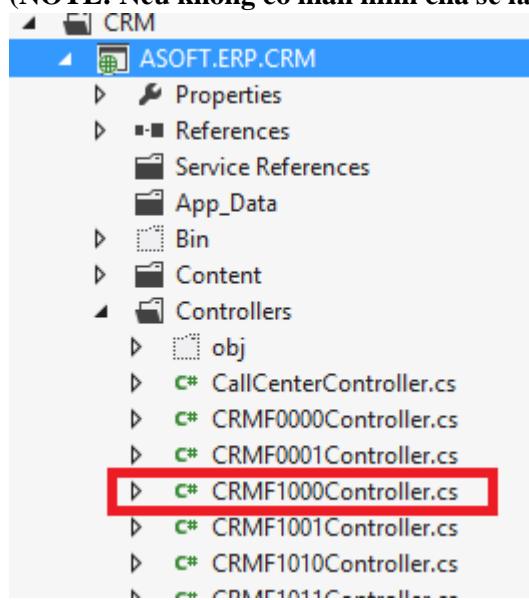
b) Customize

Cấu trúc màn hình xem chi tiết

- Cấu trúc màn hình tìm kiếm sử dụng luồng workflow **IndexData**:
- **IndexData** nằm trong project **ASOFT.ERP.WORKFLOW.Core** -> **ViewNoDetails**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để cấu trúc lưới.



- Số lượng control search sẽ phụ thuộc vào số lượng field của bảng mà màn hình thuộc vào, những cột hiển thị phụ thuộc vào trường **ViewVisible** trong bảng **sysFields** trong **DB Admin**, bảng **0** thì ẩn, **1** thì hiển thị.
- Có 2 trường hợp custom:
 - o Trường hợp 1, custom 1 hoặc nhiều control trong màn hình tìm kiếm
 - Cách custom:
 - Tạo file class **{Mã màn hình cha của màn hình xem chi tiết} + “Controller”** trong project **ASOFT.ERP.{Module}** -> Controllers.
 - (NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)

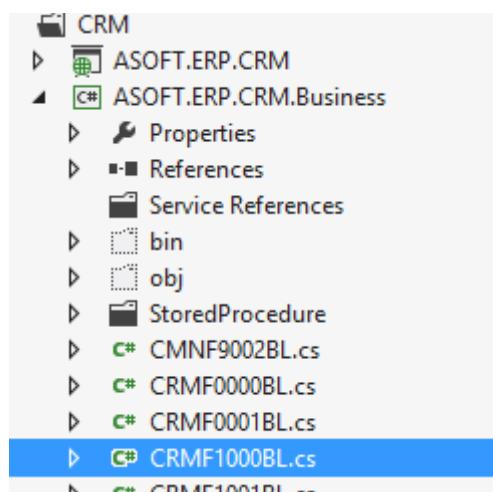


- Thêm action **“ChangeListFieldsViewNo”** vào controller.
- Trong action vừa thêm: code theo cấu trúc:

```

- public List<ASOFTSysFields> ChangeListFieldsViewNo(List<ASOFTSysFields> list, string
  id)
- {
-     //Code ở đây, tham khảo các file class khác
-     return list;
- }
```

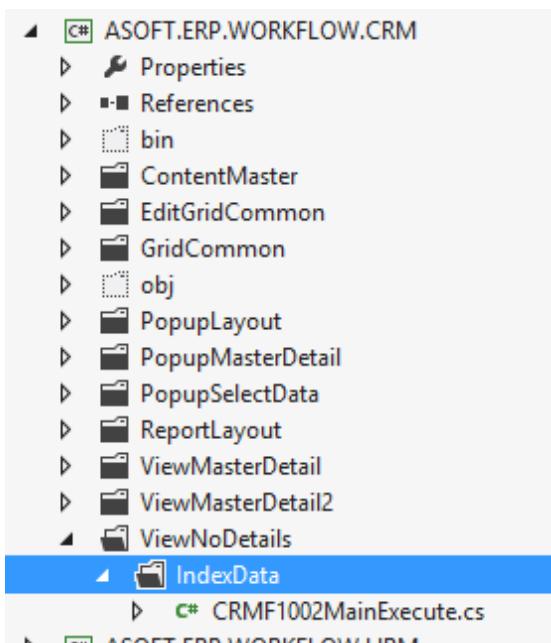
- o Trường hợp 2: custom câu SQL load màn hình nhập liệu (trường hợp sửa)
 - Cách custom:
 - Tạo file class **{Mã màn hình cha của màn hình xem chi tiết} + “BL”** trong project **ASOFT.ERP.{Module}.Business** (Nếu chưa có file)
 - (NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)



- Thêm hàm “**GetDataViewNo**” vào class.
- Trong class vừa thêm, code theo cấu trúc:

```
public Dictionary<string, object> GetDataViewNo(string APK, string id)
{
    try
    {
        //Code tại đây
    }
    catch (ASOFTException)
    {
        throw;
    }
    catch (Exception ex)
    {
        throw new ASOFTException(ASOFTLayer.Business, ex.Message, ex);
    }
}
```

- Trường hợp 3: custom hoàn toàn màn hình.
 - Cách custom:
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **ViewNoDetails** (nếu chưa có)
 - Tạo thư mục **IndexData** nằm trong thư mục **ViewNoDetails** (nếu chưa có)
 - Tạo file class **{Mã màn hình xem chi tiết} + MainExecute**



- Cấu trúc của class custom

```

- public sealed class CRMF2032MainExecute : CodeActivity
- {
-     // Define an activity input argument of type string
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

-     protected override void Execute(CodeActivityContext context)
-     {
-         Dictionary<string, Object> Out = InOut.Get(context);
-         string area = Area.Get(context);
-         string id = ScreenID.Get(context);
-         string PK = Out["PK"].ToString();
-         string Table = Out["Table"].ToString();
-         string Key = Out["key"].ToString();
-         string divisionID = Out["DivisionID"].ToString();

-         //Lấy sysField
-         List<ASOFTSysFields> ListEntities = LocalizedHelper.GetLocalizedFields(id,
-             area, Table);
-         List<ASOFTSysFieldsViewModel> ListViewModel = new
-             List<ASOFTSysFieldsViewModel>();
-             //Lấy sysScreen
-             ASOFTSysScreen scr = LocalizedHelper.GetLocalizedScreen(id, area).First();
-             scr.TitleName = LocalizedHelper.GetLocalizedResourceString(scr.Title,
-                 scr.ScreenID, area);
-             //Lấy sysGroup
-             IEnumerable<ASOFTSysGroup> ListGroupEntities =
-                 LocalizedHelper.GetLocalizesysGroup(id, area).Where(m => m.ScreenID == id ||
-                     m.ScreenID.Equals("A00")).OrderBy(m => m.TabIndex);

```

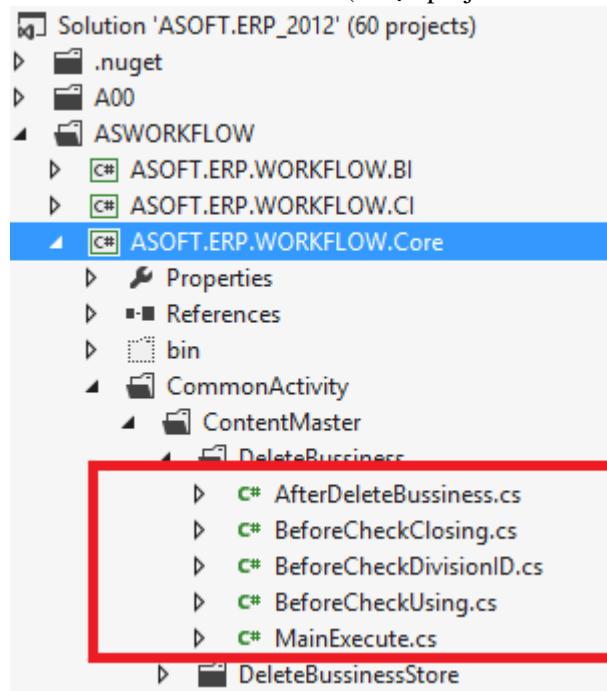
```

-     List<ASOFTSysGroupViewModel> ListGroupViewModel = new
-         List<ASOFTSysGroupViewModel>();
-             //Code ở đây, tham khảo class khác
-             Out.Add("ListViewModel", ListViewModel);
-             Out.Add("ListGroupViewModel", ListGroupViewModel);
-             Out.Add("scrpopup", scrpopup);
-
-         }
-
```

Xóa dữ liệu xem chi tiết (button xóa)



- Xóa dữ liệu xem chi tiết sẽ có 2 luồng workflow:
 - o **DeleteData** (sử dụng lại workflow xóa trong màn hình tìm kiếm) nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động **mặc định** bao gồm:
 - **BeforeCheckDivisionID**: kiểm tra khán Division thì không cho xóa.
 - **BeforeCheckUsing**: kiểm tra đã sử dụng thì không cho xóa.
 - **MainExecute**: thực hiện xóa dữ liệu.
 - **AfterDelete**: sau khi delete sẽ thực hiện gì.
 - Mỗi hành động tương ứng với 1 class nằm trong thư mục **CommonActivity -> Contentmaster DeleteData** (thuộc project **ASOFT.ERP.WORKFLOW.Core**)



Sẽ có nghiệp vụ xóa không chạy theo những hành động mặc định này. Vậy nên chúng ta phải Custom như sau:

VD: Custom hành động **BeforeCheckUsing**.

- **Cách tạo:**

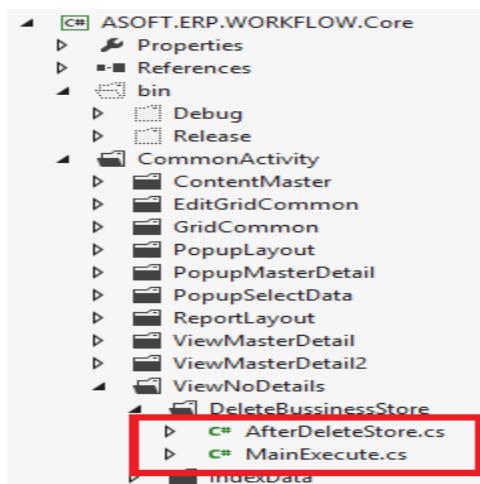
- Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
- Tạo thư mục **ViewNoDetails** (nếu chưa có)
- Tạo thư mục **DeleteBussiness** nằm trong thư mục **ViewNoDetails** (nếu chưa có)
- Tạo file class **{Mã màn hình xem chi tiết} + BeforeCheckUsing**.
- Cấu trúc của class custom

```
public sealed class CRMF2032BeforeCheckUsing : CodeActivity
{
    public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
    //Dữ liệu được truyền vào từ controller
    public InOutArgument<string> Area { get; set; } //Module
    public InOutArgument<string> ScreenID { get; set; } //Mã màn hình

    protected override void Execute(CodeActivityContext context)
    {
        Dictionary<string, object> er = InOut.Get(context); // Cách lấy biến
        InOut để sử dụng.
        //Tương tự với biến Area ScreenID

        /*
         * {Code} Tham khảo những file đã làm rồi
         */
    }
}
```

- Tương tự với những hành động khác: **{Mã màn hình xem chi tiết} + BeforeCheckDivisionID**, **: {Mã màn hình xem chi tiết} + MainExecute ...**
- **DeleteBussinessStore** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: để thực hiện được workflow này ta phải khai báo trong bảng sysScreen trường **DeleteBussinessStore** của màn hình xem chi tiết.
 - Script: `update sysScreen set DeleteBussinesStore = {Mã store} where ScreenID = {Mã màn hình xem chi tiết}`
 - VD: update sysScreen set DeleteBussinessStore = 'CRMP20302' where ScreenID = 'CRMF2032'.
- **DeleteBussinessStore** có các hành động bao gồm
 - **MainExecute**: thực hiện xóa dữ liệu bằng store.
 - **AfterDeleteStore**: sau khi delete sẽ thực hiện gì.



Cách custom:

- Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
- Tạo thư mục **ViewNoDetails** (nếu chưa có)
- Tạo thư mục **DeleteBussinessStore** nằm trong thư mục **ViewNoDetails** (nếu chưa có)
- Tạo file class **{Mã màn hình xem chi tiết} + MainExecute.**
- Cấu trúc của class custom:

```

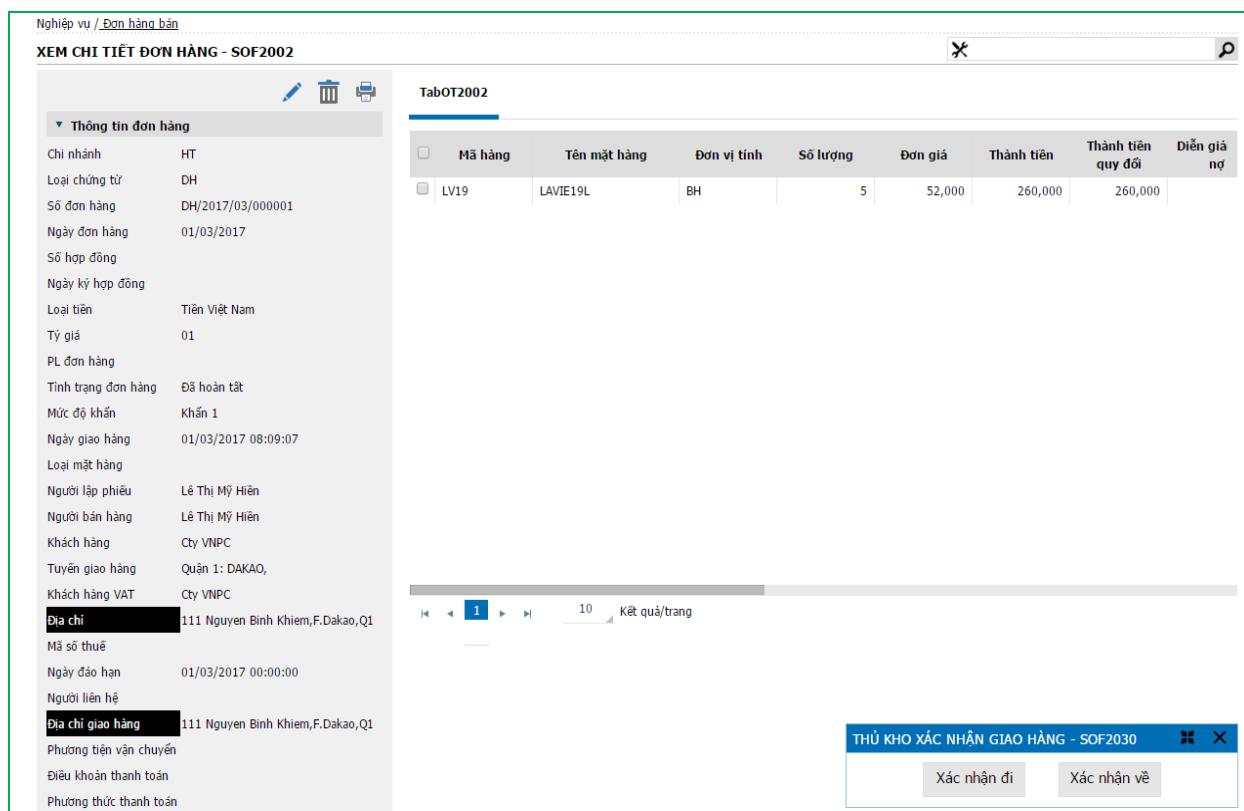
- public sealed class CRMF2032MainExecute : CodeActivity
- {
-     // Define an activity input argument of type string
-     public InOutArgument<Dictionary<string, Object>> InOut { get;
- set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

-     // If your activity returns a value, derive from
CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
{
-         // Obtain the runtime value of the Text input argument

-         //VD: Code mẫu
-         Dictionary<string, Object> data = InOut.Get(context);
-         List<string> dt = (List<string>)data["data"];
-         //VD: Code mẫu

-     }
- }
```

5) Màn hình xem chi tiết có lối detail 1 (ViewMasterDetail)



The screenshot shows a software interface titled "XEM CHI TIẾT ĐƠN HÀNG - SOF2002". On the left, there is a sidebar with various settings and details. In the center, a grid displays a list of items with columns for Mã hàng (Item ID), Tên mặt hàng (Product Name), Đơn vị tính (Unit), Số lượng (Quantity), Đơn giá (Unit Price), Thành tiền (Total Price), Thành tiền quy đổi (Converted Total Price), and Diện giá ng (Price Type). One item is selected: LV19, LAVIE19L, BH, 5, 52,000, 260,000, 260,000. At the bottom right, a modal window titled "THỦ KHO XÁC NHẬN GIAO HÀNG - SOF2030" contains two buttons: "Xác nhận đ" (Confirm) and "Xác nhận v" (Confirm and close).

a) Các bước bắt buộc

Bước 1: Tạo bảng cho màn hình

- Bước này giống với cách tạo bảng ở màn hình nhập liệu có lối detail (Tham khảo mục 3/a)

Bước 2: Tạo màn hình

Thêm 1 dòng vào bảng sysScreen

sysScreen	
APK	Tự sinh động
DivisionID	Đơn vị (VD: AS)
ModuleID	Module của bảng (VD: AsoftCRM)
ScreenID	ID Màn hình (VD: CRMF2032)
ScreenName	Tên màn hình (VD: Xem chi tiết đầu mối)
ScreenType	Loại màn hình (VD: 5: Màn hình chi tiết)
ScreenNameE	Tên tiếng anh của màn hình
Parent	Màn hình cha (VD: CRMF2030 Màn hình tìm kiếm)
sysTable	Tham chiếu tới bảng sysTable, bảng mà màn hình thuộc vào (VD: CRMT20301)

Title	Title của màn hình (Mã ngôn ngữ để lấy ra tên màn hình VD: CRMF2031.Title)
TypeInput	VD: 3: Loại màn hình xem chi tiết có lưới detail
sysCategoryBusinessID	1: Danh mục (Màn hình này chỉ có danh mục)
DisplayToolBar	ToolBar1 của màn hình (VD: 1,2 (Delete, Add)) thám khảo bảng sysToolBar
DisplayToolBar2	ToolBar2 của lưới detail (VD: 1,2 (Delete, Add)) thám khảo bảng sysToolBar

- Sử dụng script

```

declare @DivisionID nvarchar(50)
declare @ModuleID nvarchar(50)
declare @ScreenID nvarchar(50)
declare @ScreenName nvarchar(250)
declare @ScreenType tinyint
declare @ScreenNameE nvarchar(250)
declare @Parent nvarchar(50)
declare @sysTable nvarchar(50)
declare @Title nvarchar(50)
declare @TypeInput varchar(100)
declare @sysCategoryBusinessID varchar(100)
declare @Width varchar(100)

set @DivisionID = ''
set @ModuleID = ''
set @ScreenID = ''
set @ScreenName = ''
set @ScreenType = 3
set @ScreenNameE = ''
set @Parent = ''
set @sysTable = ''
set @Title = ''
set @TypeInput = ''
set @sysCategoryBusinessID = ''
set @DisplayToolBar = ''
set @DisplayToolBar2 = ''


insert into sysScreen(DivisionID, ModuleID, ScreenID, ScreenName, ScreenType,
ScreenNameE, Parent,
        sysTable, Title, TypeInput, sysCategoryBusinessID, DisplayToolBar,
DisplayToolBar2)
values(@DivisionID,@ModuleID,@ScreenID,@ScreenName,@ScreenType,@ScreenNameE,@Par
ent,
        @sysTable,@Title,@TypeInput,@sysCategoryBusinessID,@DisplayToolBar,
@DisplayToolBar2)

```

Bước 3: Tạo Group trong sysGroup

- Tham khảo màn hình ViewNoDetail

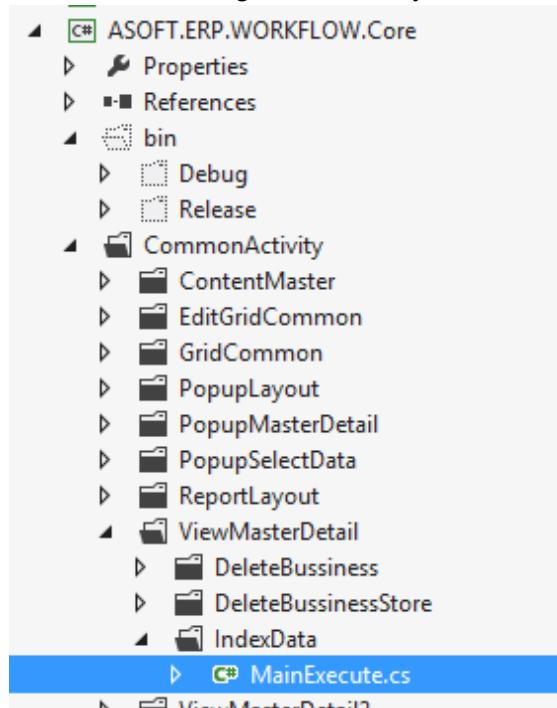
Bước 4: : Tạo các Fields hiển thị trong từng Group

- Tham khảo màn hình ViewNoDetail

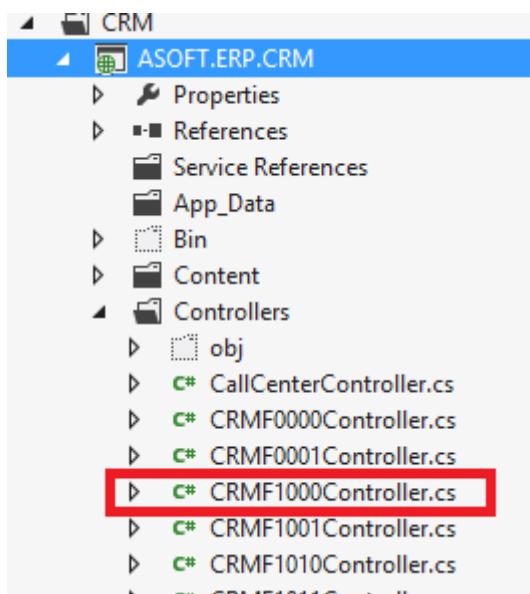
b) Customize

Cấu trúc màn hình xem chi tiết

- Cấu trúc màn hình tìm kiếm sử dụng luồng workflow **IndexData**:
- **IndexData** nằm trong project **ASOFT.ERP.WORKFLOW.Core** -> **ViewMasterDetail**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để cấu trúc lưới.



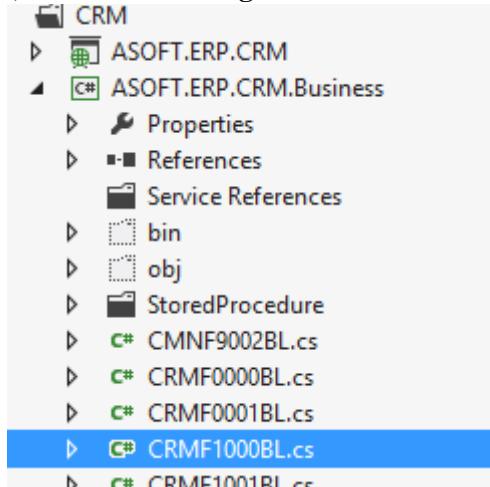
- Số lượng control search sẽ phụ thuộc vào số lượng field của bảng mà màn hình thuộc vào, những cột hiển thị phụ thuộc vào trường **ViewVisible** trong bảng **sysFields** trong **DB Admin**, bảng 0 thì ẩn, 1 thì hiển thị.
- Có 2 trường hợp custom:
 - Trường hợp 1, custom 1 hoặc nhiều control trong màn hình tìm kiếm
 - Cách custom:
 - Tạo file class {Mã màn hình cha của màn hình xem chi tiết} + “Controller” trong project ASOFT.ERP.{Module} -> Controllers. (NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)



- Thêm action “**ChangeListFieldsViewMaster**” vào controller.
- Trong action vừa thêm: code theo cấu trúc:

```
- public List<ASOFTSysFields> ChangeListFieldsViewMaster(List<ASOFTSysFields> list,
string id)
{
    {
        //Code ở đây, tham khảo các file class khác
        return list;
}
```

- Trường hợp 2: custom câu SQL load màn hình nhập liệu (trường hợp sửa)
 - Cách custom:
 - Tạo file class {Mã màn hình cha của màn hình xem chi tiết} + “BL” trong project ASOFT.ERP.{Module}. Business (Nếu chưa có file) **(NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)**



- Thêm hàm “**GetDataViewMaster**” vào class.
- Trong class vừa thêm, code theo cấu trúc:

```

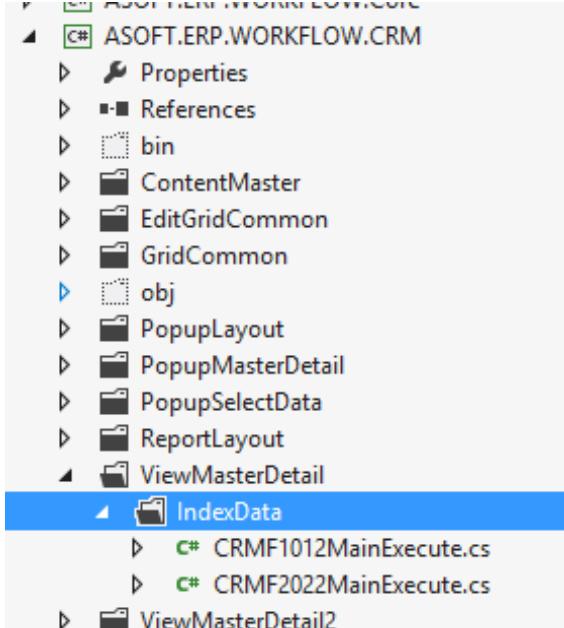
public Dictionary<string, object> GetDataViewMaster(string APK, string id)
{
    try
    {
        //Code tại đây
    }
    catch (ASOFTException)
    {
        throw;
    }
    catch (Exception ex)
    {
        throw new ASOFTException(ASOFTLayer.Business, ex.Message, ex);
    }
}

```

- Trường hợp 3: custom hoàn toàn màn hình.

- Cách custom:

- Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
- Tạo thư mục **ViewMasterDetail** (nếu chưa có)
- Tạo thư mục **IndexData** nằm trong thư mục **ViewMasterDetail** (nếu chưa có)
- Tạo file class **{Mã màn xem chi tiết} + MainExecute**



- Cấu trúc của class custom

```

- public sealed class CRMF2032MainExecute : CodeActivity
- {
-     // Define an activity input argument of type string
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }
-
```

```

- // If your activity returns a value, derive from CodeActivity<TResult>
- // and return the value from the Execute method.
- protected override void Execute(CodeActivityContext context)
{
-     Dictionary<string, Object> Out = InOut.Get(context);
-     string id = ScreenID.Get(context);
-     string area = Area.Get(context);
-     string PK = Out["PK"].ToString();
-     string Table = Out["Table"].ToString();
-     string key = Out["key"].ToString();
-     string divisionID = Out["DivisionID"].ToString();

-     ASOFTGridBL b11 = new ASOFTGridBL();

-     List<ASOFTSysFieldsViewModel> list1 = new List<ASOFTSysFieldsViewModel>();
-     List<ASOFTSysScreen> listScreenChild = new List<ASOFTSysScreen>();
-     List<ASOFTSysFields> list = new List<ASOFTSysFields>();
-     List<ASOFTSysGroup> listgr = new List<ASOFTSysGroup>();
-     List<ASOFTSysGroupViewModel> listgr1 = new List<ASOFTSysGroupViewModel>();
-     List<ASOFTSysTable> listtb = new List<ASOFTSysTable>();
-     List<ASOFTSysTableViewModel> listtb1 = new List<ASOFTSysTableViewModel>();
-     List<ASOFTSysFieldsViewModel> listPK1 = new
List<ASOFTSysFieldsViewModel>();
-     List<ASOFTSysFields> listPK = new List<ASOFTSysFields>();

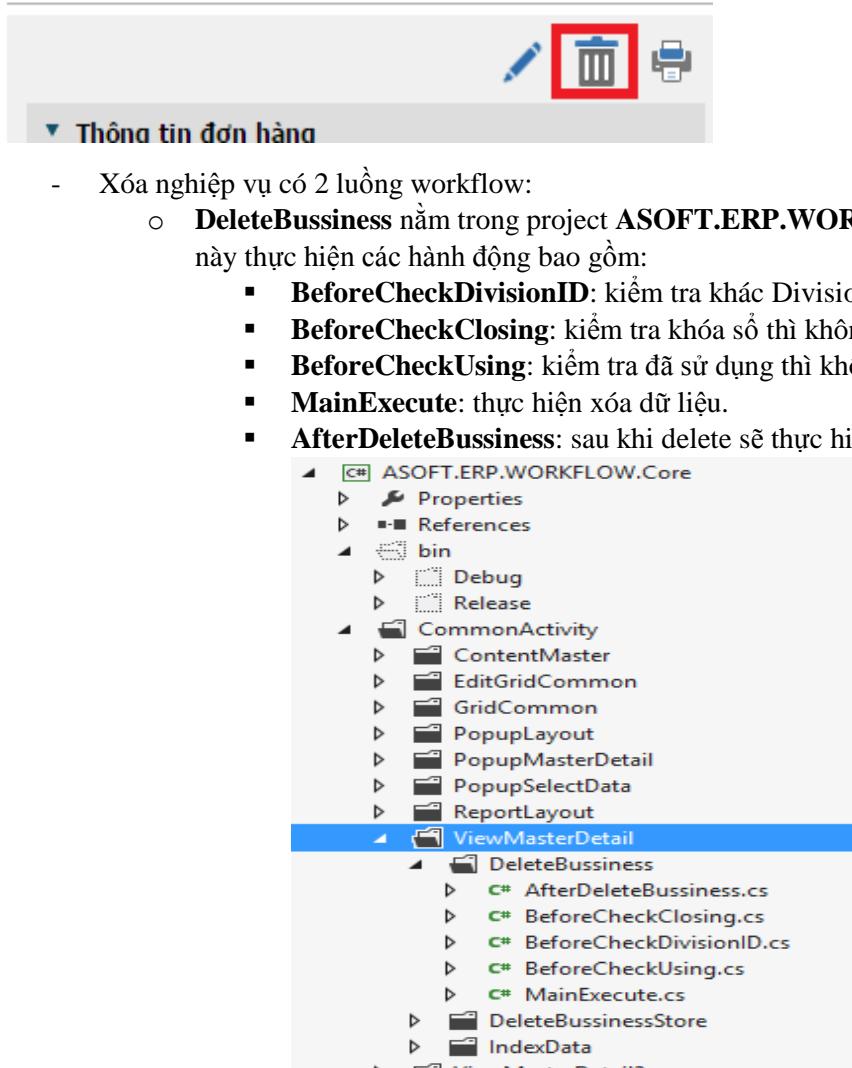
-     ASOFTSysScreen scr = new ASOFTSysScreen();
-     Dictionary<string, object> data = new Dictionary<string, object>();

-     scr = LocalizedHelper.GetLocalizedScreen(id, area).First();
-     scr.TitleName = LocalizedHelper.GetLocalizedString(scr.Title,
scr.ScreenID, area);
-     list = LocalizedHelper.GetLocalizedFields(id, area, scr.sysTable);
-     listgr = LocalizedHelper.GetLocalizedGroup(id, area);
-     listScreenChild = LocalizedHelper.GetLocalizedChild(scr.Parent, area);
-     listtb = LocalizedHelper.GetLocalizedTable(id, area, scr.sysTable);

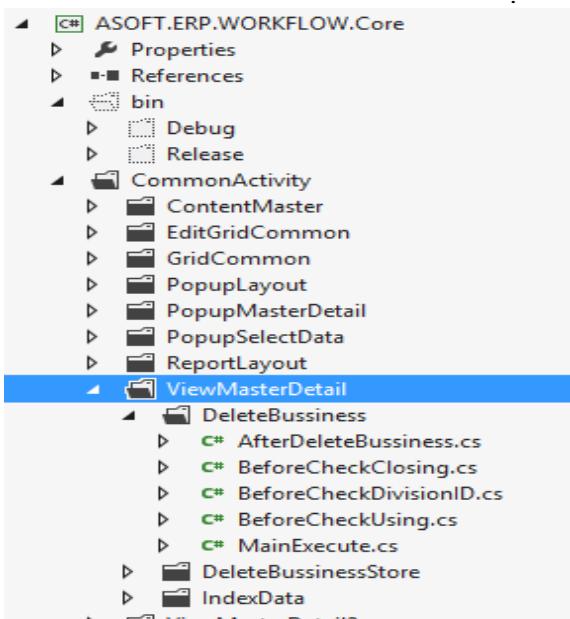
//Code ở đây, tham khảo những class khác

-     Out["list1"] = list1;
-     Out["typeinput"] = typeinput;
-     Out["listgr1"] = listgr1;
-     Out["listtb1"] = listtb1;
-     Out["listPK1"] = listPK1;
-     Out["scrpopup"] = scrpopup;
-     Out["sysCategoryBusinessID"] = scr.sysCategoryBusinessID;
// Obtain the runtime value of the Text input argument
- }
- }
```

Xóa dữ liệu màn hình xem chi tiết (button Xóa master)



- Xóa nghiệp vụ có 2 luồng workflow:
 - o **DeleteBussiness** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động bao gồm:
 - **BeforeCheckDivisionID**: kiểm tra khán Division thì không cho xóa.
 - **BeforeCheckClosing**: kiểm tra khóa sổ thì không cho xóa.
 - **BeforeCheckUsing**: kiểm tra đã sử dụng thì không cho xóa.
 - **MainExecute**: thực hiện xóa dữ liệu.
 - **AfterDeleteBussiness**: sau khi delete sẽ thực hiện gì.



Sẽ có nghiệp vụ xóa không chạy theo những hành động mặc định này. Vậy nên chúng ta phải Custom như sau:

VD: Custom hành động **BeforeCheckUsing**.

- **Cách tạo:**
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)
 - Tạo thư mục **ViewMasterDetail** (nếu chưa có)
 - Tạo thư mục **DeleteBussiness** nằm trong thư mục **ViewMasterDetail** (nếu chưa có)
 - Tạo file class **{Mã màn hình xem chi tiết} + BeforeCheckUsing**.
 - Cấu trúc của class custom

```
public sealed class CRMF2032BeforeCheckUsing : CodeActivity
```

```

public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
//Dữ liệu được truyền vào từ controller
public InOutArgument<string> Area { get; set; } //Module
public InOutArgument<string> ScreenID { get; set; } //Mã màn hình

protected override void Execute(CodeActivityContext context)
{
    Dictionary<string, object> er = InOut.Get(context); // Cách lấy biến
    InOut để sử dụng.
    //Tương tự với biến Area ScreenID

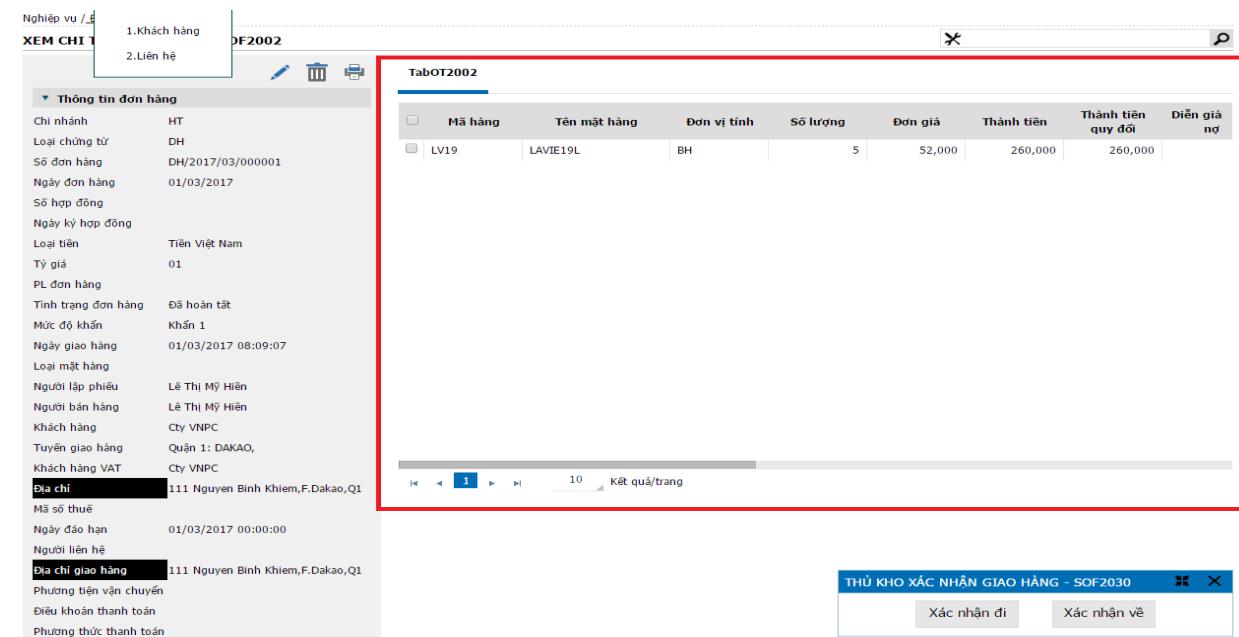
    /*
     * {Code} Tham khảo những file đã làm rồi
     */
}
}

```

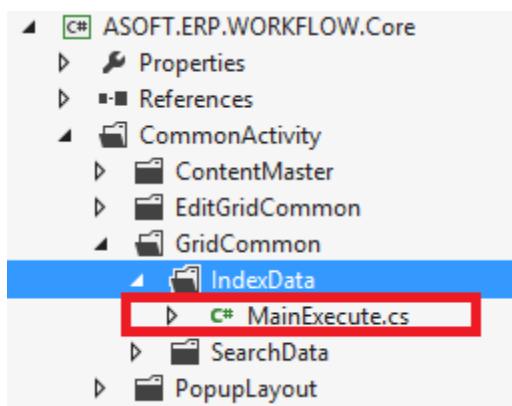
- Tương tự với những hành động khác: {Mã màn hình xem chi tiết} + BeforeCheckDivisionID, : {Mã màn hình xem chi tiết} + MainExecute ...

- **DeleteBussinessStore:** tương tự như DeleteBussinessStore giống như màn hình xem chi tiết, chỉ thay tên thư mục **ViewNoDetail** -> **ViewMasterDetail**

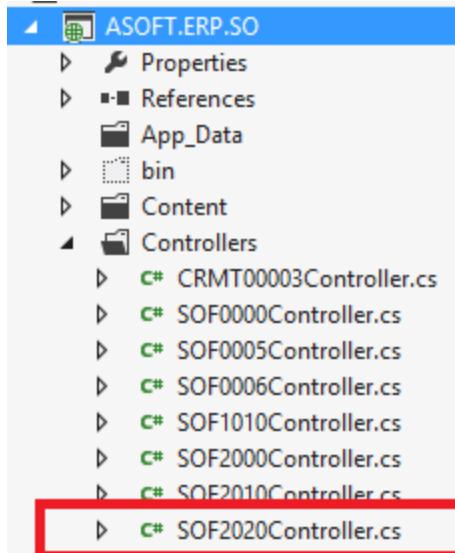
Cấu trúc lưới detail



- Cấu trúc lưới sử dụng luồng workflow **IndexData**:
- **IndexData** nằm trong project **ASOFT.ERP.WORKFLOW.Core** -> **GridCommon**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để cấu trúc lưới.



- Custom lưới:
 - o Trường hợp 1: custom cột trên lưới
 - Cách custom:
 - Tạo file class {Mã màn hình cha của màn hình xem chi tiết} + “Controller” trong project ASOFT.ERP.{Module} -> Controllers. (NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)



- Thêm action “ChangeFieldsGrid” + (Mã màn hình xem chi tiết thuộc bảng detail) + “_” + (Mã table của lưới detail màn hình xem chi tiết) vào controller.
- Trong action vừa thêm: code theo cấu trúc:

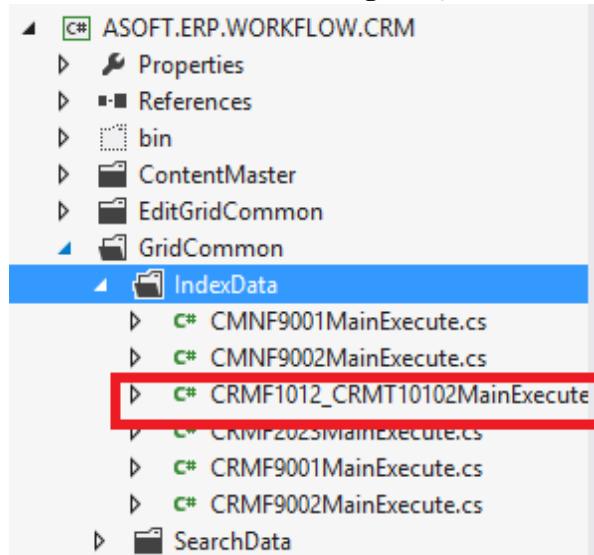
```

- public List<ASOFTSysFields> ChangeFieldsGridSOF2021_0T2102(List<ASOFTSysFields> list,
  string id)
{
    //Code ở đây, tham khảo các file class khác
    return list;
}

```

- o Trường hợp 2: custom toàn bộ lưới:
 - Cách custom:
 - Tạo project ASOFT.ERP.WORKFLOW.{Module}(nếu chưa có)

- Tạo thư mục **GridCommon** (nếu chưa có)
- Tạo thư mục **IndexData** nằm trong thư mục **GridCommon** (nếu chưa có)
- Tạo file class {Mã màn hình xem chi tiết} + “_” + {Mã table của lưới detail màn hình chi tiết (bảng con)} + **MainExecute**



- Cấu trúc của class custom

```
- public sealed class CRMF1012_CRMT10102MainExecute : CodeActivity
- {
-     // Define an activity input argument of type string
-     public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

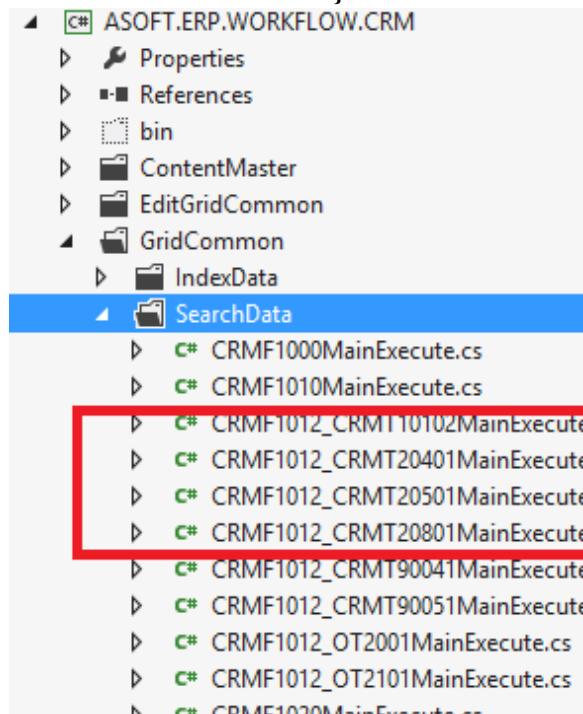
-     // If your activity returns a value, derive from CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
{
    // Obtain the runtime value of the Text input argument
    string id = ScreenID.Get(context);
    string module = Area.Get(context);
    Dictionary<string, Object> Input = InOut.Get(context);
    List<ASOFTSysFields> list = new List<ASOFTSysFields>();
    ASOFTSysScreen scr = new ASOFTSysScreen();
    List<string> list1 = new List<string>();
    string table = Input["table"].ToString();

    //Code ở đây, thám khảo các class đã làm

    //Các biến bắt buộc phải trả về
    Input.Add("sysFields", list); //List cột trên lưới
    Input.Add("Language", list1); //Ngôn ngữ tương ứng của các cột
    Input.Add("scr", scr); //Màn hình tìm kiếm
}
}
```

Load lưới detail

- Load lưới detail sử dụng luồng workflow **SearchData**:
 - o **SearchData** nằm trong project **ASOFT.ERP.WORKFLOW.Core -> GridCommon**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để search dữ liệu trả về controller
 - o Nếu nghiệp vụ search yêu cầu sử dụng store và câu SQL đặc thù thì ta sẽ custom:
 - o VD:
 - Cách tạo:
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **GridCommon** (nếu chưa có)
 - Tạo thư mục **SearchData** nằm trong thư mục **GridCommon** (nếu chưa có)
 - Tạo file class **{Mã màn hình xem chi tiết} + “_” + {Mã table detail của màn hình xem chi tiết} + MainExecute**



- Cấu trúc của class custom

```
public sealed class CRMF1012_CRM110102MainExecute : CodeActivity
{
    // Define an activity input argument of type string
    public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
    public InOutArgument<string> Area { get; set; }
    public InOutArgument<string> ScreenID { get; set; }

    // If your activity returns a value, derive from CodeActivity<TResult>
    // and return the value from the Execute method.
}
```

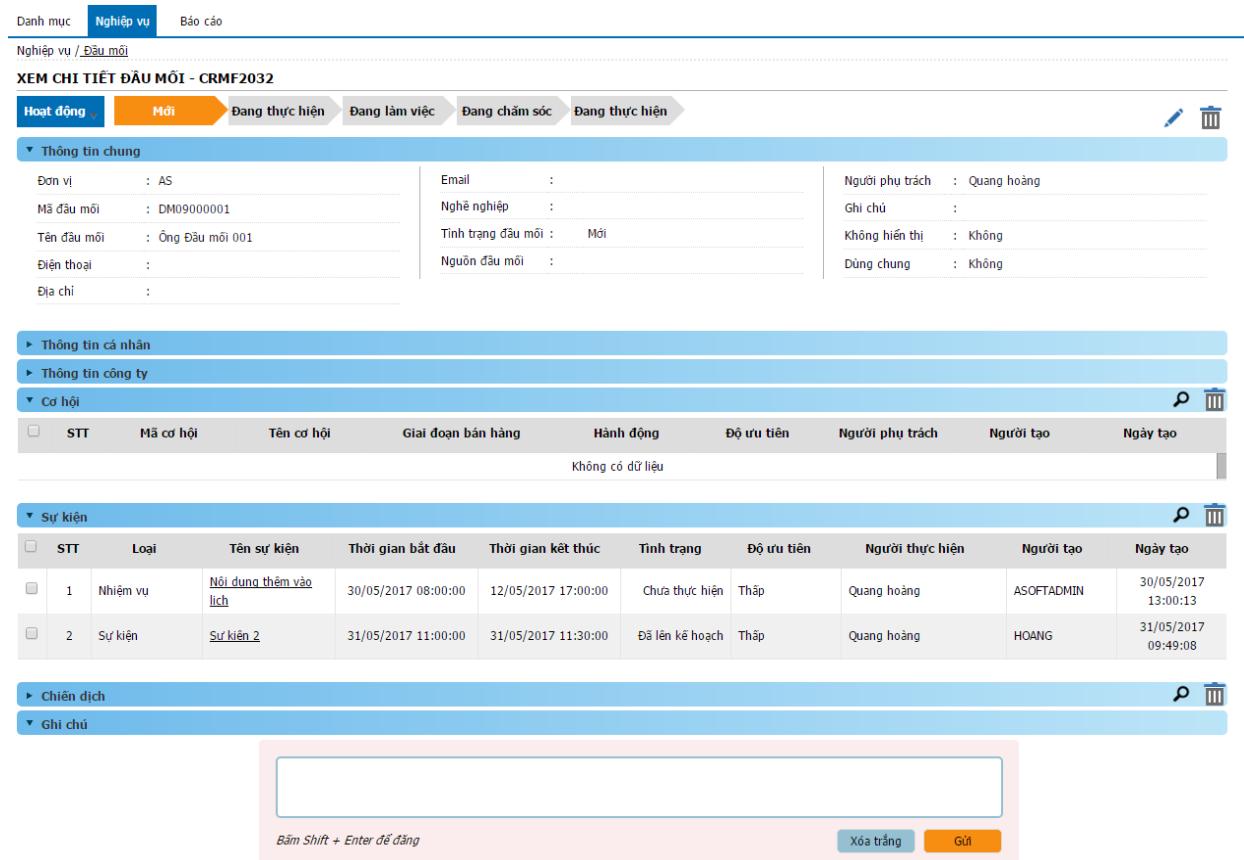
```

protected override void Execute(CodeActivityContext context)
{
    // Obtain the runtime value of the Text input argument
    // Obtain the runtime value of the Text input argument
    Dictionary<string, List<string>> args = (Dictionary<string,
List<string>>)InOut.Get(context)[ "args" ];
    ASOFTSysGrid ds = new ASOFTSysGrid();
    List<string> list = new List<string>();
    List<string> list1 = new List<string>();

    //Code tại đây, tham khảo các class khác

    Dictionary<string, Object> Ouput = InOut.Get(context);
    Ouput.Add("kq", ds);
}
-
```

6) Màn hình xem chi tiết có lưới detail 2 (ViewMasterDetail2)



The screenshot shows a software interface for managing leads. At the top, there's a navigation bar with tabs: 'Danh mục' (selected), 'Nghiệp vụ' (highlighted in blue), and 'Báo cáo'. Below this, a breadcrumb trail shows 'Nghiệp vụ / Đầu mối'. The main title is 'XEM CHI TIẾT ĐẦU MỐI - CRMF2032'. A horizontal navigation bar below the title has tabs: 'Hoạt động' (selected), 'Mới' (highlighted in orange), 'Đang thực hiện', 'Đang làm việc', 'Đang chăm sóc', and 'Đang thực hiện'. To the right of this bar are edit and delete icons.

The main content area is divided into sections:

- Thông tin chung:** Contains fields for basic information like 'Đơn vị' (AS), 'Email', 'Người phụ trách' (Quang hoàng), 'Mã đầu mối' (DM09000001), 'Nghề nghiệp', 'Ghi chú', 'Tên đầu mối' (Ông Đầu mối 001), 'Tình trạng đầu mối' (Mới), 'Không hiển thị' (Không), 'Điện thoại', 'Nguồn đầu mối', 'Đóng chung', and 'Địa chỉ'.
- Thông tin cá nhân:** (Collapsible section)
- Thông tin công ty:** (Collapsible section)
- Cơ hội:** (Collapsible section) Contains a table with columns: STT, Mã cơ hội, Tên cơ hội, Giai đoạn bán hàng, Hành động, Độ ưu tiên, Người phụ trách, Người tạo, Ngày tạo. The table shows one row: 'Không có dữ liệu'.
- Sự kiện:** (Collapsible section) Contains a table with columns: STT, Loại, Tên sự kiện, Thời gian bắt đầu, Thời gian kết thúc, Tình trạng, Độ ưu tiên, Người thực hiện, Người tạo, Ngày tạo. The table shows two rows:

STT	Loại	Tên sự kiện	Thời gian bắt đầu	Thời gian kết thúc	Tình trạng	Độ ưu tiên	Người thực hiện	Người tạo	Ngày tạo
1	Nhiệm vụ	Nội dung thêm vào lịch	30/05/2017 08:00:00	12/05/2017 17:00:00	Chưa thực hiện	Thấp	Quang hoàng	ASOFTADMIN	30/05/2017 13:00:13
2	Sự kiện	Sự kiện_2	31/05/2017 11:00:00	31/05/2017 11:30:00	Đã lên kế hoạch	Thấp	Quang hoàng	HOANG	31/05/2017 09:49:08
- Chiến dịch:** (Collapsible section)
- Ghi chú:** (Collapsible section) Contains a text input field with placeholder 'Bấm Shift + Enter để đăng' and buttons for 'Xóa trắng' and 'Gửi'.

a) Các bước bắt buộc

Bước 1: Tạo bảng cho màn hình

- Bước này giống với cách tạo bảng ở màn hình nhập liệu có lưới detail (Tham khảo mục 3/a)

Bước 2: Tạo màn hình

Thêm 1 dòng vào bảng sysScreen

sysScreen	
APK	Tự sinh động
DivisionID	Đơn vị (VD: AS)
ModuleID	Module của bảng (VD: AsoftCRM)
ScreenID	ID Màn hình (VD: CRMF2032)
ScreenName	Tên màn hình (VD: Xem chi tiết đầu mối)
ScreenType	Loại màn hình (VD: 5: Màn hình chi tiết)
ScreenNameE	Tên tiếng anh của màn hình
Parent	Màn hình cha (VD: CRMF2030 Màn hình tìm kiếm)
sysTable	Tham chiếu tới bảng sysTable, bảng mà màn hình thuộc vào (VD: CRMT20301)
Title	Title của màn hình (Mã ngôn ngữ để lấy ra tên màn hình VD: CRMF2031.Title)
TypeInput	VD: 5: Loại màn hình xem chi tiết có lối detail 2
sysCategoryBusinessID	1: Danh mục (Màn hình này chỉ có danh mục)
DisplayToolBar	ToolBar1 của màn hình (VD: 1,2 (Delete, Add)) thám khảo bảng sysToolBar
DisplayToolBar2	ToolBar2 của lối detail (VD: 1,2 (Delete, Add)) thám khảo bảng sysToolBar

- Sử dụng script

```

declare @DivisionID nvarchar(50)
declare @ModuleID nvarchar(50)
declare @ScreenID nvarchar(50)
declare @ScreenName nvarchar(250)
declare @ScreenType tinyint
declare @ScreenNameE nvarchar(250)
declare @Parent nvarchar(50)
declare @sysTable nvarchar(50)
declare @Title nvarchar(50)
declare @TypeInput varchar(100)
declare @sysCategoryBusinessID varchar(100)
declare @Width varchar(100)

set @DivisionID = ''
set @ModuleID = ''
set @ScreenID = ''
set @ScreenName = ''
set @ScreenType = 3
set @ScreenNameE = ''
set @Parent = ''
set @sysTable = ''
set @Title = ''
set @TypeInput = ''
set @sysCategoryBusinessID = ''
set @DisplayToolBar = ''

```

```

set @DisplayToolBar2 = ''

insert into sysScreen(DivisionID, ModuleID, ScreenID, ScreenName, ScreenType,
ScreenNameE, Parent,
sysTable, Title, TypeInput, sysCategoryBusinessID, DisplayToolBar,
DisplayToolBar2)
values(@DivisionID,@ModuleID,@ScreenID,@ScreenName,@ScreenType,@ScreenNameE,@Parent,
@sysTable,@Title,@TypeInput,@sysCategoryBusinessID,@DisplayToolBar,
@DisplayToolBar2)

```

Bước 3: Tạo Group trong sysGroup

- Tạo group cho lưới detail

sysGroup	
GroupName	Mã ngôn ngữ Group
Description	Thông tin về Group
ModuleID	Mã phân hệ - Nếu là Group dùng chung thì mã phân hệ là Asoft00
ScreenID	Mã màn hình – Nếu là Group dùng chung thì mã màn hình là A00
TabIndex	Thứ tự hiển thị của Group
GroupID	Mã Group
sysTable	Mã bảng của lưới detail

```

declare @GroupName nvarchar(100)
declare @Description nvarchar(1000)
declare @ModuleID nvarchar(50)
declare @ScreenID nvarchar(50)
declare @TabIndex int

set @GroupName = ''
set @Description = ''
set @ModuleID = ''
set @ScreenID = ''
set @TabIndex = ''
set @GroupID = ''
set @sysTable = ''

insert into sysGroup(GroupName, Description, ModuleID, ScreenID, TabIndex,
GroupID, sysTable)
values(@GroupName, @Description, @ModuleID, @ScreenID, @TabIndex, @GroupID,
@sysTable)

```

- Tạo group cho master: Tham khảo màn hình nhập liệu không lưới chi tiết (ViewNoDetail)

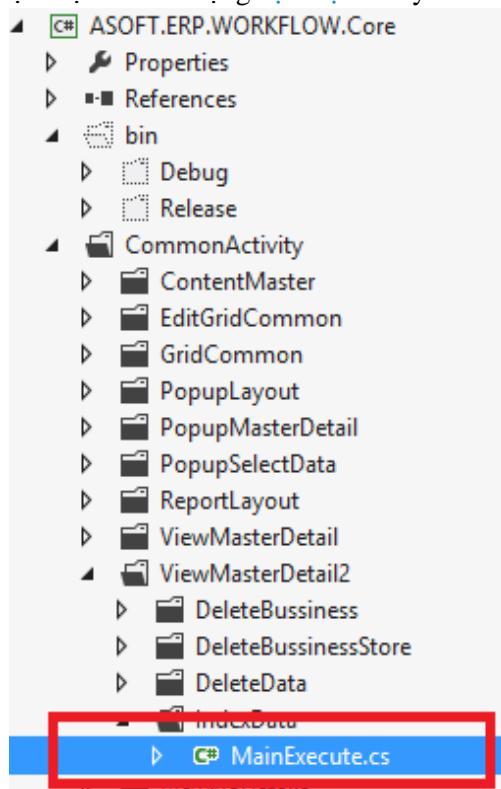
Bước 4: Tạo các Fields hiển thị trong từng Group

- Tham khảo màn hình nhập liệu không lưới chi tiết (ViewNoDetail)

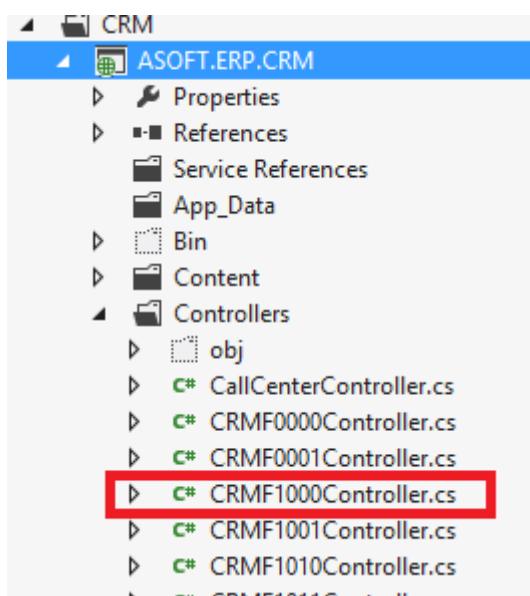
b) Customize

Cấu trúc màn hình xem chi tiết

- Cấu trúc màn hình tìm kiếm sử dụng luồng workflow **IndexData**:
- **IndexData** nằm trong project **ASOFT.ERP.WORKFLOW.Core** -> **ViewMasterDetail2**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để cấu trúc lưới.



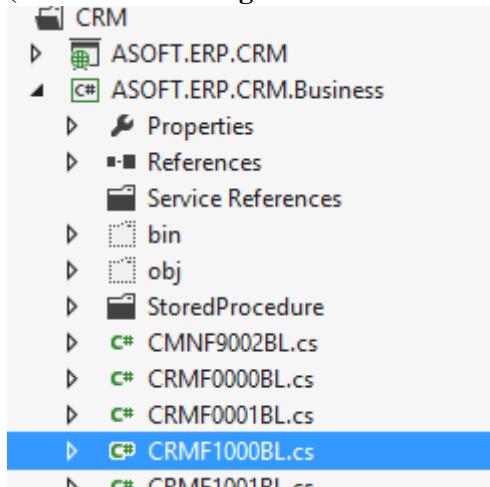
- Số lượng control search sẽ phụ thuộc vào số lượng field của bảng mà màn hình thuộc vào, những cột hiển thị phụ thuộc vào trường **ViewVisible** trong bảng **sysFields** trong **DB Admin**, bảng 0 thì ẩn, 1 thì hiển thị.
- Có 2 trường hợp custom:
 - o Trường hợp 1, custom 1 hoặc nhiều control trong màn hình tìm kiếm
 - Cách custom:
 - Tạo file class {Mã màn hình cha của màn hình xem chi tiết} + “Controller” trong project ASOFT.ERP.{Module} -> Controllers. (NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)



- Thêm action “**ChangeListFieldsViewMaster2**” vào controller.
- Trong action vừa thêm: code theo cấu trúc:

```
- public List<ASOFTSysFields> ChangeListFieldsViewMaster2(List<ASOFTSysFields> list,
string id)
{
    {
        //Code ở đây, tham khảo các file class khác
        return list;
}
```

- Trường hợp 2: custom câu SQL load màn hình nhập liệu (trường hợp sửa)
 - Cách custom:
 - Tạo file class {Mã màn hình cha của màn hình xem chi tiết} + “BL” trong project ASOFT.ERP.{Module}.Business (Nếu chưa có file) **(NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)**



- Thêm hàm “**GetDataViewMaster2**” vào class.
- Trong class vừa thêm, code theo cấu trúc:

```

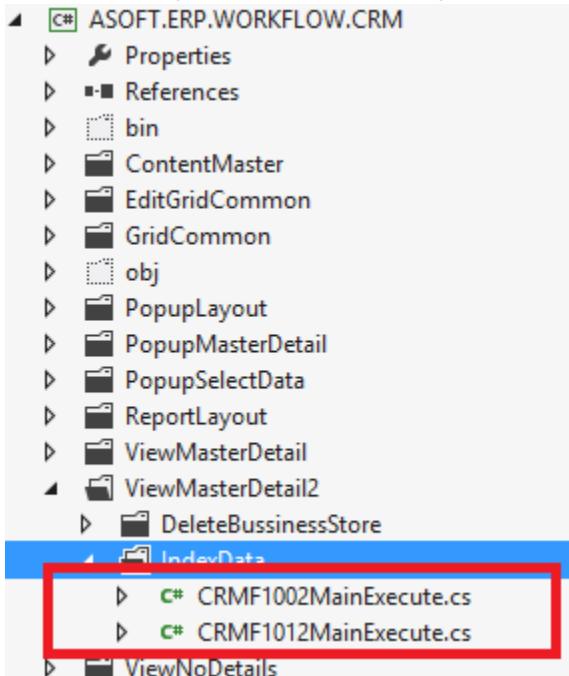
public Dictionary<string, object> GetDataViewMaster2(string APK, string id)
{
    try
    {
        //Code tại đây
    }
    catch (ASOFTException)
    {
        throw;
    }
    catch (Exception ex)
    {
        throw new ASOFTException(ASOFTLayer.Business, ex.Message, ex);
    }
}

```

- Trường hợp 3: custom hoàn toàn màn hình.

- Cách custom:

- Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
- Tạo thư mục **ViewMasterDetail2** (nếu chưa có)
- Tạo thư mục **IndexData** nằm trong thư mục **ViewMasterDetail2**(nếu chưa có)
- Tạo file class **{Mã màn xem chi tiết} + MainExecute**



- Cấu trúc của class custom

```

public sealed class CRMF2032MainExecute : CodeActivity
{
    // Define an activity input argument of type string
    public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
    public InOutArgument<string> Area { get; set; }
}

```

```

public InOutArgument<string> ScreenID { get; set; }

// If your activity returns a value, derive from CodeActivity<TResult>
// and return the value from the Execute method.
protected override void Execute(CodeActivityContext context)
{
    // Obtain the runtime value of the Text input argument
    Dictionary<string, Object> Out = InOut.Get(context);
    string area = Area.Get(context);
    string id = ScreenID.Get(context);
    string PK = Out["PK"].ToString();
    string Table = Out["Table"].ToString();
    string key = Out["key"].ToString();
    string divisionID = Out["DivisionID"].ToString();

    ASOFTGridBL bl1 = new ASOFTGridBL();
    ASOFTSysScreen scr = new ASOFTSysScreen();
    Dictionary<string, object> data = new Dictionary<string, object>();
    List<ASOFTSysFieldsViewModel> list1 = new List<ASOFTSysFieldsViewModel>();
    List<ASOFTSysScreen> listScreenChild = new List<ASOFTSysScreen>();
    List<ASOFTSysFields> list = new List<ASOFTSysFields>();
    List<ASOFTSysGroup> listgr = new List<ASOFTSysGroup>();
    List<ASOFTSysGroupViewModel> listgr1 = new List<ASOFTSysGroupViewModel>();
    List<ASOFTSysTable> listtb = new List<ASOFTSysTable>();
    List<ASOFTSysTableViewModel> listtb1 = new List<ASOFTSysTableViewModel>();
    List<ASOFTSysFieldsViewModel> listPK1 = new List<ASOFTSysFieldsViewModel>();
    List<ASOFTSysFields> listPK = new List<ASOFTSysFields>();

    scr = LocalizedHelper.GetLocalizedScreen(id, area).First();
    scr.TitleName = LocalizedHelper.GetLocalizedResourceString(scr.Title,
scr.ScreenID, area);
    list = LocalizedHelper.GetLocalizedFields(id, area, scr.sysTable);
    listgr = LocalizedHelper.GetLocalizedGroup(id, area);
    listScreenChild = LocalizedHelper.GetLocalizedChild(scr.Parent, area);

    //Code ở đây, tham khảo màn hình khác

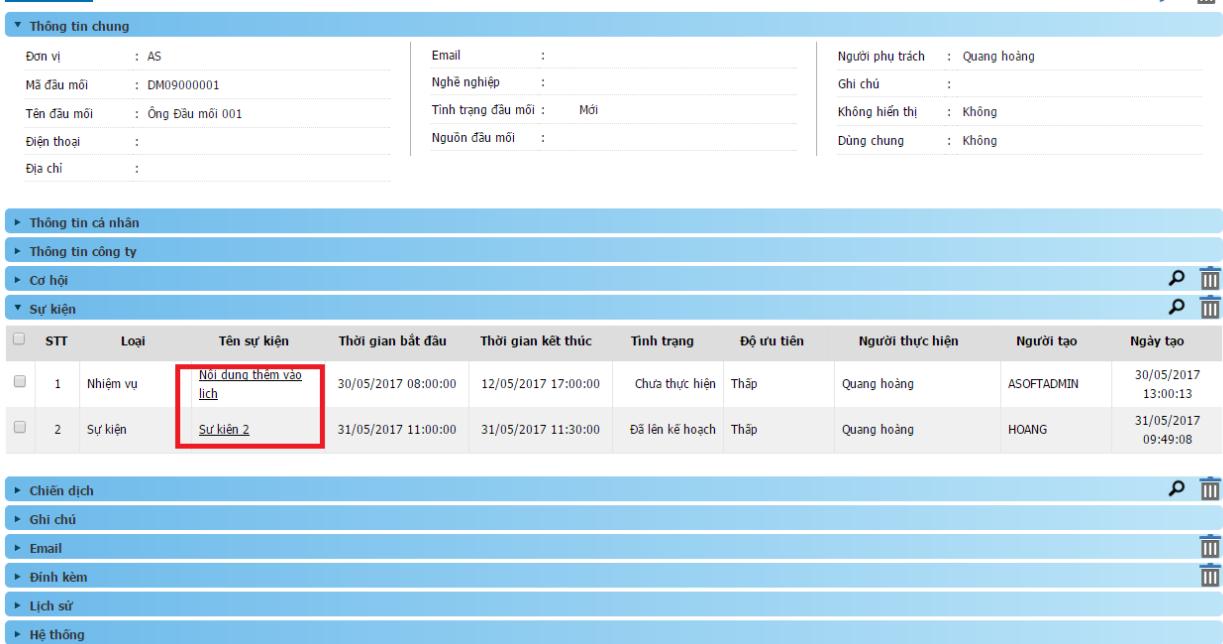
    Out["ListViewModel"] = list1;
    Out["typeinput"] = typeinput;
    Out["ListGroupViewModel"] = listgr1;
    Out["listtb1"] = listtb1;
    Out["listPK1"] = listPK1;
    Out["scrpopup"] = scrpopup;
    Out["sysCategoryBusinessID"] = scr.sysCategoryBusinessID;

}
-
```

Cấu trúc lưới detail

- Tương tự màn hình nhập liệu có lưới detail 1 (ViewMasterDetail)

- Custom link trên lưới:



The screenshot shows a software application interface for managing events. At the top, there are several tabs: 'Thông tin chung', 'Thông tin cá nhân', 'Thông tin công ty', 'Cá nhân', and 'Sự kiện'. The 'Sự kiện' tab is active, displaying a table of events. The table has columns: STT, Loại, Tên sự kiện, Thời gian bắt đầu, Thời gian kết thúc, Tình trạng, Độ ưu tiên, Người thực hiện, Người tạo, and Ngày tạo. Two events are listed:

STT	Loại	Tên sự kiện	Thời gian bắt đầu	Thời gian kết thúc	Tình trạng	Độ ưu tiên	Người thực hiện	Người tạo	Ngày tạo
1	Nhiệm vụ	Nội dung thêm vào lịch	30/05/2017 08:00:00	12/05/2017 17:00:00	Chưa thực hiện	Thấp	Quang hoàng	ASOFTADMIN	30/05/2017 13:00:13
2	Sự kiện	Sự kiện 2	31/05/2017 11:00:00	31/05/2017 11:30:00	Đã lên kế hoạch	Thấp	Quang hoàng	HOANG	31/05/2017 09:49:08

Below the table, there are more tabs: 'Chiến dịch', 'Ghi chú', 'Email', 'Đính kèm', 'Lịch sử', and 'Hệ thống'. The 'Lịch sử' tab is active.

- Cách custom: khai báo trường RefUrl trong bảng sysTable đối với bảng của lưới detail.

```
Update sysTable set RefUrl = {Link muốn custom} where TableName = {Table detail}
```

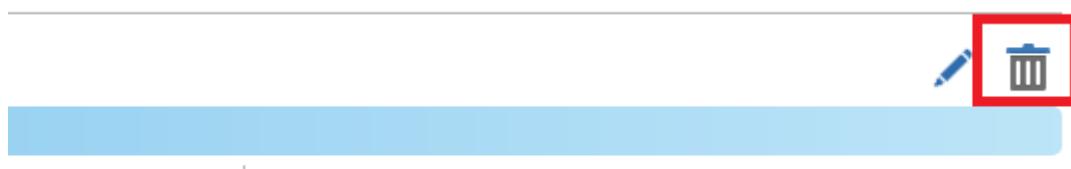
VD:

```
update sysTable set RefUrl =
'/PopupLayout/Index/CRM/CRMF9005?EventID=#=EventID#&RelatedToID=#=RelatedToID#&RelatedTo
ypeID_REL=#=RelatedTypeID_REL#&Type=1' where TableName = 'CRMT90051'
```

Load lưới detail

- Tương tự màn hình nhập liệu có lưới detail 1 (ViewMasterDetail)

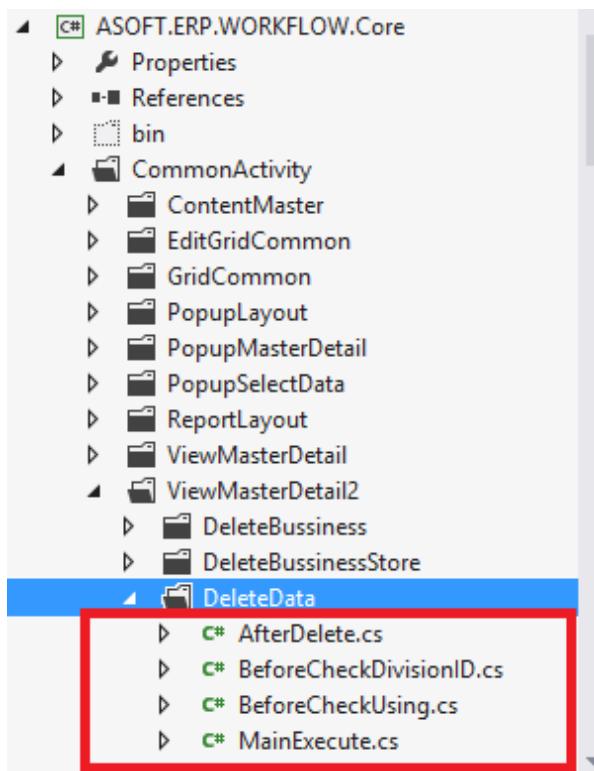
Xóa dữ liệu master màn hình xem chi tiết 2 (button Xóa master)



Xóa master màn hình này có 2 loại gồm:

- Xóa danh mục sẽ có 2 luồng workflow:
 - o **DeleteData** nằm trong project **ASOFT.ERP.WORKFLOW.Core** -> **ViewMasterDetail2**: tại workflow này thực hiện các hành động **mặc định** bao gồm:
 - **BeforeCheckDivisionID**: kiểm tra khác Division thì không cho xóa.
 - **BeforeCheckUsing**: kiểm tra đã sử dụng thì không cho xóa.
 - **MainExecute**: thực hiện xóa dữ liệu.
 - **AfterDelete**: sau khi delete sẽ thực hiện gì.

- Mỗi hành động tương ứng với 1 class nằm trong thư mục **CommonActivity** -> **ViewMasterDetail2** -> **DeleteData** (thuộc project **ASOFT.ERP.WORKFLOW.Core**).



Mỗi nghiệp vụ xóa sẽ có đặc thù riêng nên nhiều khi chúng ta không thể thực hiện theo các hành động mặc định như trên được. Do đó ta phải thực hiện custom các hành động:

VD: Custom hành động **BeforeCheckUsing**.

- **Cách tạo:**

- Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
- Tạo thư mục **ViewMasterDetail2** (nếu chưa có)
- Tạo thư mục **DeleteData** nằm trong thư mục **ViewMasterDetail2** (nếu chưa có)
- Tạo file class **{Mã màn hình xem chi tiết} + BeforeCheckUsing**.
- Cấu trúc của class custom

```
public sealed class CRMF2032BeforeCheckUsing : CodeActivity
{
    public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
//Dữ liệu được truyền vào từ controller
    public InOutArgument<string> Area { get; set; } //Module
    public InOutArgument<string> ScreenID { get; set; } //Mã màn hình

    protected override void Execute(CodeActivityContext context)
    {
```

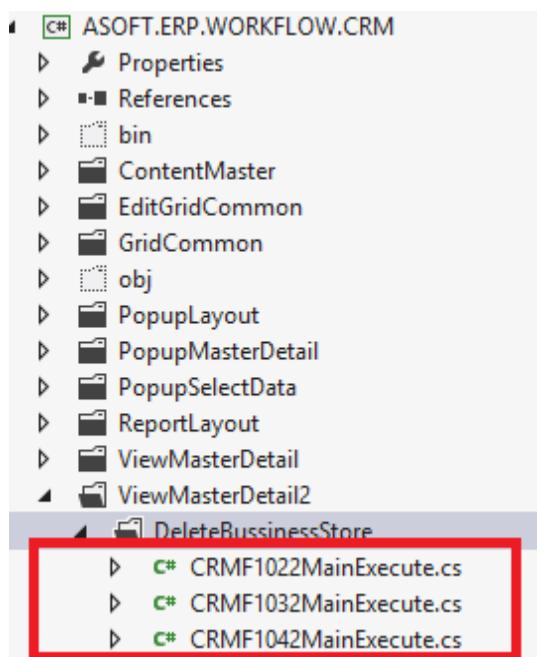
```
        Dictionary<string, object> er = InOut.Get(context); // Cách lấy biến
InOut để sử dụng.
        //Tương tự với biến Area ScreenID

        /*
         * {Code} Tham khảo những file đã làm rồi
         */
    }
}
```

- Tương tự với những hành động khác: {Mã màn hình xem chi tiết} + BeforeCheckDivisionID, { Mã màn hình xem chi tiết } + MainExecute
 - DeleteBussinessStore nằm trong project ASOFT.ERP.WORKFLOW.Core -> ViewMasterDetail2: để thực hiện được workflow này ta phải khai báo trong bảng sysScreen trường DeleteBussinessStore của màn hình xem chi tiết.
 - Script: `update sysScreen set DeleteBussinesStore = {Mã store} where ScreenID = {Mã màn hình tìm kiếm}`
 - VD: `update sysScreen set DeleteBussinessStore = 'CRMP20302' where ScreenID = 'CRMF2030'.`
 - DeleteBussinessStore có các hành động bao gồm
 - **MainExecute:** thực hiện xóa dữ liệu bằng store.
 - **AfterDeleteStore:** sau khi delete sẽ thực hiện gì.
 - Tại đây workflow sẽ tự chạy store với những parameter của store được định sẵn. Và tất nhiên không phải parameter định sẵn lúc nào cũng đúng với store được khai báo. Đối với trường hợp này ta sẽ custom cũng giống như luồng workflow **DeleteData**.

VD:

- **Cách tạo:**
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **ViewMasterDetail2** (nếu chưa có)
 - Tạo thư mục **DeleteBusinessStore** nằm trong thư mục **ViewMasterDetail2** (nếu chưa có)
 - Tạo file class **{Mã màn hình xem chi tiết} + MainExecute.**



- Cấu trúc của class custom:

```

- public sealed class CRMF2032MainExecute : CodeActivity
- {
-     // Define an activity input argument of type string
-     public InOutArgument<Dictionary<string, Object>> InOut { get;
- set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

-     // If your activity returns a value, derive from
CodeActivity<TResult>
-     // and return the value from the Execute method.
protected override void Execute(CodeActivityContext context)
{
    // Obtain the runtime value of the Text input argument

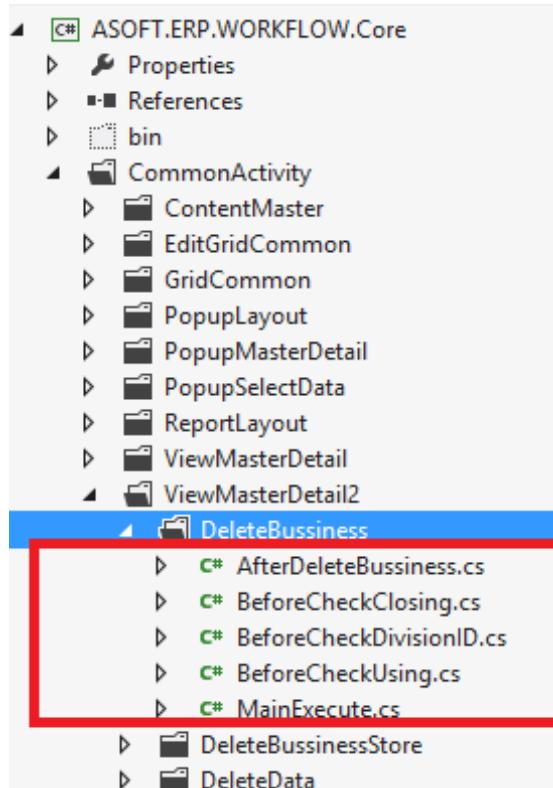
    //VD: Code mẫu
    Dictionary<string, Object> data = InOut.Get(context);
    List<string> dt = (List<string>)data["data"];
    //VD: Code mẫu

}
}

```

- Xóa nghiệp vụ cũng có 2 luồng workflow:
 - o **DeleteBusiness** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động bao gồm:
 - **BeforeCheckDivisionID**: kiểm tra khác Division thì không cho xóa.
 - **BeforeCheckClosing**: kiểm tra khóa sổ thì không cho xóa.

- **BeforeCheckUsing:** kiểm tra đã sử dụng thì không cho xóa.
- **MainExecute:** thực hiện xóa dữ liệu.
- **AfterDeleteBussiness:** sau khi delete sẽ thực hiện gì.
- Mỗi hành động tương ứng với 1 class nằm trong thư mục **CommonActivity** -> **ViewMasterDetail2 -> DeleteBussiness** (thuộc project **ASOFT.ERP.WORKFLOW.Core**).



Cũng tương tự như xóa danh mục sẽ có nghiệp vụ xóa không chạy theo những hành động mặc định này. Vậy nên chúng ta phải Custom như sau:

VD: Custom hành động **BeforeCheckUsing**.

- **Cách tạo:**
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)
 - Tạo thư mục **ViewMasterDetail2** (nếu chưa có)
 - Tạo thư mục **DeleteBussiness** nằm trong thư mục **ViewMasterDetail2** (nếu chưa có)
 - Tạo file class **{Mã màn hình xem chi tiết} + BeforeCheckUsing**.
 - Cấu trúc của class custom: tương tự như workflow **DeleteData**
- Tương tự với những hành động khác: **{Mã màn hình xem chi tiết } + BeforeCheckDivisionID**, **: {Mã màn hình xem chi tiết } + MainExecute ...**
- **DeleteBussinessStore:** tương tự như **DeleteBussinessStore** trong xóa danh mục

Xóa dữ liệu detail màn hình xem chi tiết 2 (các button Xóa Detail)

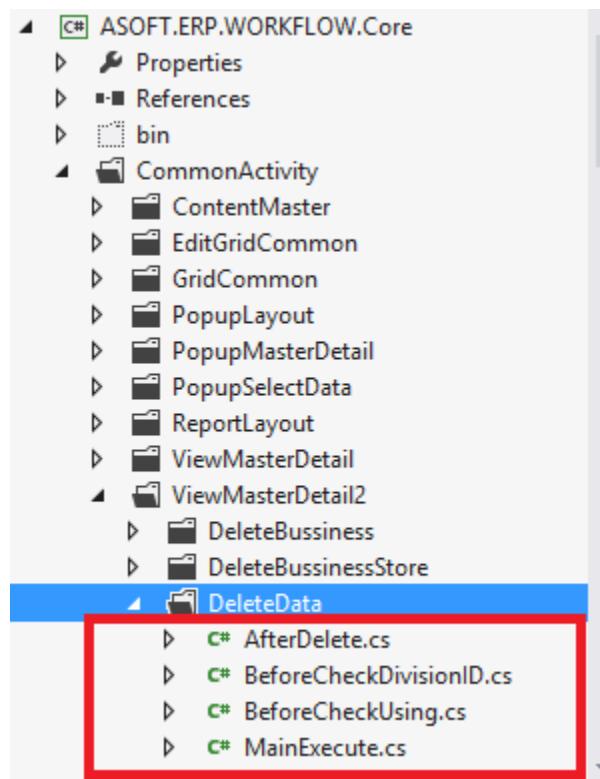
Xóa master màn hình này có 2 loại gồm: (Giống mục xóa dữ liệu detail, khác một số chỗ được **in đậm**)

- Xóa danh mục sẽ có 2 luồng workflow:

- o **DeleteData** nằm trong project **ASOFT.ERP.WORKFLOW.Core** ->

ViewMasterDetail2: tại workflow này thực hiện các hành động **mặc định** bao gồm:

- **BeforeCheckDivisionID**: kiểm tra khán Division thì không cho xóa.
- **BeforeCheckUsing**: kiểm tra đã sử dụng thì không cho xóa.
- **MainExecute**: thực hiện xóa dữ liệu.
- **AfterDelete**: sau khi delete sẽ thực hiện gì.
- Mỗi hành động tương ứng với 1 class nằm trong thư mục **CommonActivity** -> **ViewMasterDetail2** -> **DeleteData** (thuộc project **ASOFT.ERP.WORKFLOW.Core**).



Mỗi nghiệp vụ xóa sẽ có đặc thù riêng nên nhiều khi chúng ta không thể thực hiện theo các hành động mặc định như trên được. Do đó ta phải thực hiện custom các hành động:

VD: Custom hành động **BeforeCheckUsing**.

- **Cách tạo:**

- Tạo project **ASOFT.ERP.WORKFLOW.{Module}** (nếu chưa có)
- Tạo thư mục **ViewMasterDetail2** (nếu chưa có)
- Tạo thư mục **DeleteData** nằm trong thư mục **ViewMasterDetail2** (nếu chưa có)

- Tạo file class { **Mã màn hình nhập liệu của bảng detail** } + **BeforeCheckUsing**.
- Cấu trúc của class custom

```
public sealed class CRMF2051BeforeCheckUsing : CodeActivity
{
    public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
    //Dữ liệu được truyền vào từ controller
    public InOutArgument<string> Area { get; set; } //Module
    public InOutArgument<string> ScreenID { get; set; } //Mã màn hình

    protected override void Execute(CodeActivityContext context)
    {
        Dictionary<string, object> er = InOut.Get(context); // Cách lấy biến
        InOut để sử dụng.
        //Tương tự với biến Area ScreenID

        /*
         * {Code} Tham khảo những file đã làm rồi
         */
    }
}
```

- Tương tự với những hành động khác: { **Mã màn hình nhập liệu của bảng detail** } + **BeforeCheckDivisionID**, { **Mã màn hình nhập liệu của bảng detail** } + **MainExecute**
- **DeleteBussinessStore** nằm trong project **ASOFT.ERP.WORKFLOW.Core** -> **ViewMasterDetail2**: để thực hiện được workflow này ta phải khai báo trong bảng sysScreen trường **DeleteBussinessStore** của màn hình xem chi tiết.
 - Script: `update sysScreen set DeleteBussinesStore = {Mã store} where ScreenID = { Mã màn hình nhập liệu của bảng detail }`
 - VD: update sysScreen set DeleteBussinessStore = 'CRMP20502' where ScreenID = 'CRMF2051'.

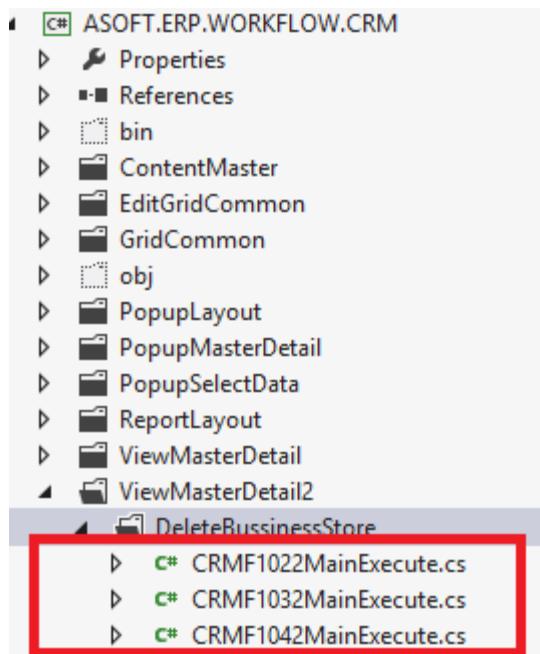
DeleteBussinessStore có các hành động bao gồm

- **MainExecute**: thực hiện xóa dữ liệu bảng store.
- **AfterDeleteStore**: sau khi delete sẽ thực hiện gì.
- Tại đây workflow sẽ tự chạy store với những parameter của store được định sẵn. Và tất nhiên không phải parameter định sẵn lúc nào cũng đúng với store được khai báo. Đối với trường hợp này ta sẽ custom cũng giống như luồng workflow **DeleteData**.

VD:

- **Cách tạo:**
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **ViewMasterDetail2** (nếu chưa có)

- Tạo thư mục **DeleteBussinessStore** nằm trong thư mục **ViewMasterDetail2** (nếu chưa có)
- Tạo file class { **Mã màn hình nhập liệu của bảng detail** } + **MainExecute**.



- Cấu trúc của class custom:

```

- public sealed class CRMF2032MainExecute : CodeActivity
- {
-     // Define an activity input argument of type string
-     public InOutArgument<Dictionary<string, Object>> InOut { get;
set; }
-     public InOutArgument<string> Area { get; set; }
-     public InOutArgument<string> ScreenID { get; set; }

-     // If your activity returns a value, derive from
CodeActivity<TResult>
-     // and return the value from the Execute method.
-     protected override void Execute(CodeActivityContext context)
{
-         // Obtain the runtime value of the Text input argument

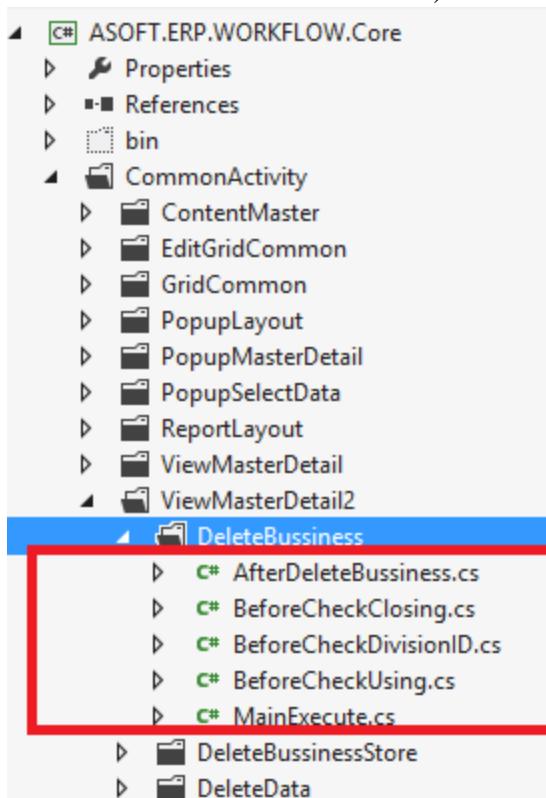
-         //VD: Code mẫu
-         Dictionary<string, Object> data = InOut.Get(context);
-         List<string> dt = (List<string>)data["data"];
-         //VD: Code mẫu

-     }
}

```

- Xóa nghiệp vụ cũng có 2 luồng workflow:

- **DeleteBusiness** nằm trong project **ASOFT.ERP.WORKFLOW.Core**: tại workflow này thực hiện các hành động bao gồm:
 - **BeforeCheckDivisionID**: kiểm tra khán Division thì không cho xóa.
 - **BeforeCheckClosing**: kiểm tra khóa sổ thì không cho xóa.
 - **BeforeCheckUsing**: kiểm tra đã sử dụng thì không cho xóa.
 - **MainExecute**: thực hiện xóa dữ liệu.
 - **AfterDeleteBusiness**: sau khi delete sẽ thực hiện gì.
 - Mỗi hành động tương ứng với 1 class nằm trong thư mục **CommonActivity -> ViewMasterDetail2 -> DeleteBusiness** (thuộc project **ASOFT.ERP.WORKFLOW.Core**).



Cũng tương tự như xóa danh mục sẽ có nghiệp vụ xóa không chạy theo những hành động mặc định này. Vậy nên chúng ta phải Custom như sau:

VD: Custom hành động **BeforeCheckUsing**.

- **Cách tạo:**
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **ViewMasterDetail2** (nếu chưa có)
 - Tạo thư mục **DeleteBusiness** nằm trong thư mục **ViewMasterDetail2** (nếu chưa có)
 - Tạo file class { **Mã màn hình nhập liệu của bảng detail** } + **BeforeCheckUsing**.
 - Cấu trúc của class custom: tương tự như workflow **DeleteData**

- Tương tự với những hành động khác: { **Mã màn hình nhập liệu của bảng detail** } + **BeforeCheckDivisionID**, : { **Mã màn hình nhập liệu của bảng detail** } + **MainExecute** ...

- **DeleteBussinessStore**: tương tự như **DeleteBussinessStore** trong xóa danh mục

Xóa dữ liệu detail trường hợp có bảng trung gian

VD bảng trung gian:

Các loại bảng trung (sẽ có thêm nữa tùy nghiệp vụ)

- Loại bảng trung gian (1:1):
 - Bảng đầu mối (CRMT20301) và bảng Cơ hội (CRMT20501) liên kết với nhau qua một bảng Cơ hội _ Đầu mối (CRMT20301_CRMT20501_REL) bảng đó gọi là bảng trung gian. Và bảng trung gian này chỉ liên kết 2 bảng này với nhau mà không có bảng thứ 3 can thiệp vào.
 - Bảng trung gian 1:1 này sẽ chứa các trường tương ứng với các khóa chính của 2 bảng đầu mối và cơ hội. VD: **OpportunityID** lưu dữ liệu khóa chính bảng Cơ hội (CRMT20501), **LeadID** lưu dữ liệu khóa chính bảng đầu mối (CRMT20301)

	APK	OpportunityID	LeadID
1	7D86E83F-1740-485E-A7F0-F86BE9F5A64F	663fcf34-6d35-4e3b-829f-2c886362cd95	310f76bf-76b4-4bc9-ad4f-4b7e4273697a
2	F15AF8FB-3DAD-403C-9B4B-88CE7B0A5FAD	9c2d5872-b11e-4a42-bc94-827d10a6a5fd	f5d85f61-4cb9-4027-9ab4-a9d7563e9f91
3	0C5636D1-8E78-4266-8359-21F722BBC64	ab5eb494-0625-448f-9653-c9079bffeb56	f8a5483e-ffd2-4769-aabb-95de3158bcd
4	5D46FDD2-8443-42D8-B056-A5453866A9D1	dd45f760-93f4-4191-b143-0d5a7be0e735	8a0929a9f370-4731-8ca1-b48337369462
5	92D3F39C-2D24-4E97-BDF5-B32AC21549A8	dd45f760-93f4-4191-b143-0d5a7be0e735	fd21d169-7c99-4a8e-8f5a-d9f54fc9e0d

- Loại bảng trung gian (1:n):
 - Bảng liên quan này cho phép nhiều bảng có thể liên kết với một bảng.
 VD: Bảng Sự kiện (CRMT90051) liên kết với tất cả các bảng khác (Đầu mối (CRMT20301), Cơ hội (CRMT20501), Chiến dịch (CRMT20401), Yêu cầu (CRMT20801)) thông qua bảng trung gian CRMT90051_REL.
 - Trường **RelatedToID** lưu khóa chính của những bảng liên quan đến bảng sự kiện
 Trường **EventID** lưu dữ liệu khóa chính của bảng Sự kiện.
 Trường **RelatedTypeID_REL** lưu loại đối tượng của những bản liên quan đến bảng sự kiện

	APK	DivisionID	EventID	RelatedTypeID_REL	RelatedToID
1	EF052D9C-4669-478E-AE6B-BAA8A80F0301	AS	1	4	851ae468-1a3a-44d7-8a46-25303010e8e1
2	D245090B-C424-4A8C-8199-0AAD8A9EBB6E	AS	1	4	dd45f760-93f4-4191-b143-0d5a7be0e735
3	AC243E8B-A8ED-4B40-B384-2AD21B6C51D4	AS	1	4	dd9b4c0b-2e1f-44d6-a334-504ecacf78f
4	230E8420-8432-4128-BC29-8201BBD61511	AS	2	3	663367af-814e-479a-b889-28c8bb55a8e9
5	2F75E1FB-9C60-4314-A127-5B4A18954144	AS	2	4	851ae468-1a3a-44d7-8a46-25303010e8e1

Trường hợp xóa bảng trung gian này vẫn chạy theo luồng workflow (**DeleteData** hoặc **DeleteBussiness**) nhưng nó chỉ xóa được dữ liệu trong bảng detail.

Sẽ có 1 số nghiệp vụ yêu cầu chỉ xóa trong bảng trung gian không được xóa trong bảng detail, hoặc là phải xóa cả 2 bảng. Để làm được điều này có 2 cách:

- Cách 1: Custom giống luồng **DeleteData** (Danh mục) hoặc **DeleteBusiness** (Nghiệp vụ)
- Cách 2: Khai báo trong bảng sysTable các trường **RelTable**, **RelColumn**, **TableDelete**, **TypeREL**, **RealRelColumn**
 - o RelTable: Tên bảng trung gian.
 - o RelColumn: Tên column lưu giá trị khóa chính của bảng master(bảng cha) trong bảng trung gian.
 - o TableDelete: Những bảng sẽ được xóa.
 - o TypeREL: Loại đối tượng liên quan.
 - o RealRelColumn: Tên cột khóa chính bảng con

Các bước khai báo:

- o Bước 1: Khai báo trường **RelTable**, **RelColumn**, **TableDelete**, **RealRelColumn** cho bảng detail(bảng con).
 - Trường **RelTable** và **RelColumn** sẽ liên quan với nhau.
 VD:
 - Bảng **đầu mối CRMT20301 (bảng con)** sử dụng bảng liên quan **CRMT20501_CRMT20301_REL** để liên kết với bảng **Cơ hội CRMT20501 (bảng cha)** ta phải:
 - o Khai báo vào trường **RelTable** tên các bảng liên quan và ngăn cách nhau bởi dấu phẩy, VD: CRMT20501_CRMT20301_REL,
 - o Khai báo vào trường **RelColumn** tên trường trong bảng liên quan mà lưu khóa chính của **bảng con** tương ứng với thứ tự khai báo bảng liên quan trong trường **RelTable**. VD: OpportunityID.
 - o NOTE:
 - Nếu bảng đầu mối (Bảng con) vẫn còn liên kết với 1 bảng khác (**bảng cha**) thì ta vẫn khai báo tiếp trong **RelTable** và **RelColumn** như trên nhưng ngăn cách nhau bởi dấu phẩy.
 - VD: **RelTable**:
 CRMT20501_CRMT20301_REL, CRMT20301_CRMT10101_REL
 - **RelColumn**: OpportunityID, AccountID
 - Bảng nào đã khai báo bảng liên quan 1:1 thì không được khai báo bảng liên quan 1:n và ngược lại.

RelTable	RelColumn
CRMT20501_CRMT20301_REL, CRMT10301_REL	OpportunityID, RelatedToID /
NULL	NULL /

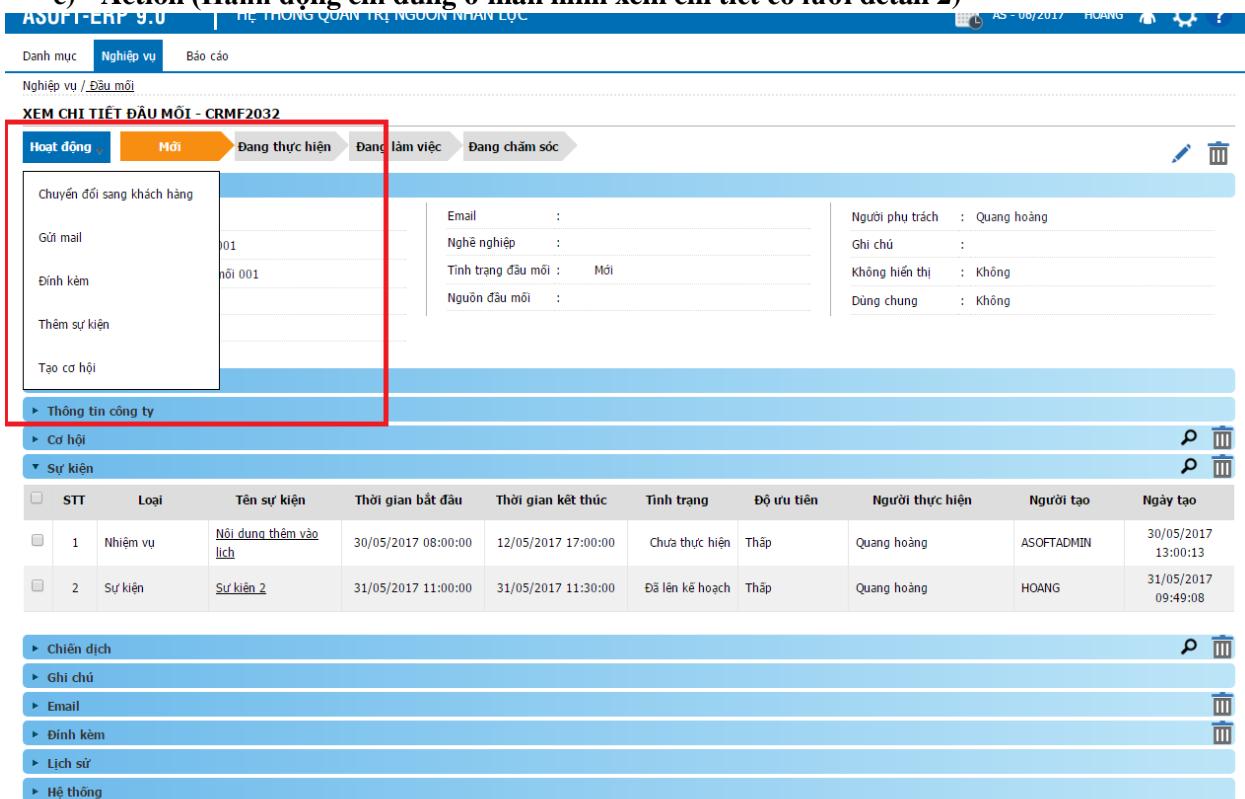
- Trường **TableDelete**: Khai báo những bảng sẽ được xóa theo cấu trúc:
 - ;{Bảng Detail};{Bảng liên quan}

- NOTE: Chỉ khai báo với trường hợp bảng liên quan 1:n, ngược lại để null.
- VD: Bảng sự kiện CRMT90051 (bảng con) sử dụng bảng liên quan CRMT90051_REL để liên kết với bảng Cơ hội CRMT20501 (bảng cha). Lúc này nếu khi xóa dữ liệu trong bảng sự kiện CRMT90051 chỉ yêu cầu xóa bảng trung gian nhưng không xóa trong bảng sự kiện CRMT90051. Ta khai báo: ;;CRMT90051_REL. Còn nếu xóa hết cả 2 bảng ta khai báo: ;CRMT90051;CRMT90051_REL

TableName	TableDelete
CRMT90031	;CRMT90031;CRMT90031_REL
CRMT90051	::CRMT90051_REL
NULL	NULL

- Trường **RealRelColumn**: khóa chính câu bảng detail

c) Action (Hành động chỉ dùng ở màn hình xem chi tiết có lối detail 2)



STT	Loại	Tên sự kiện	Thời gian bắt đầu	Thời gian kết thúc	Tình trạng	Độ ưu tiên	Người thực hiện	Người tạo	Ngày tạo
1	Nhiệm vụ	<u>Nội dung thêm vào lịch</u>	30/05/2017 08:00:00	12/05/2017 17:00:00	Chưa thực hiện	Thấp	Quang hoàng	ASOFTADMIN	30/05/2017 13:00:13
2	Sự kiện	<u>Sự kiện_2</u>	31/05/2017 11:00:00	31/05/2017 11:30:00	Đã lên kế hoạch	Thấp	Quang hoàng	HOANG	31/05/2017 09:49:08

- Khai báo trong bảng sysAction

sysAction	
sysActionID	Tự sinh động

Description	Điễn giải (VD: Hành động gửi mail)
IDLanguage	Mã ngôn ngữ của action đó (VD: A00.SendMail)
Url	NULL
ClickEvent	Sự kiện jquery của action đó (VD: sendMail_Click())
ScreenID	Dùng để phân quyền action (VD: CMNF9005 (Màn hình gửi mail))
ActionID	Mã của Action (VD: SendMail)
ModuleID	Module phân quyền cho action đó (VD: AsoftCRM)

```

declare @Description nvarchar(MAX)
declare @IDLanguage nvarchar(MAX)
declare @Url nvarchar(MAX)
declare @ClickEvent nvarchar(MAX)
declare @ScreenID nvarchar(MAX)
declare @ActionID nvarchar(MAX)
declare @ModuleID nvarchar(MAX)
set @Description=N''
set @IDLanguage=N''
set @Url=N''
set @ClickEvent=N''
set @ScreenID=N''
set @ActionID=N''
set @ModuleID=N''
If not exists(select top 1 1 from [dbo].[sysAction] where [ActionID] = @ActionID)Begin
insert into
sysAction([Description],[IDLanguage],[Url],[ClickEvent],[ScreenID],[ActionID],[ModuleID])values(@Description,@IDLanguage,@Url,@ClickEvent,@ScreenID,@ActionID,@ModuleID)
End

```

- Gắn action vào màn hình xem chi tiết.
- Khai báo trường sysActionID trong bảng Screen đối với màn hình xem chi tiết đó.

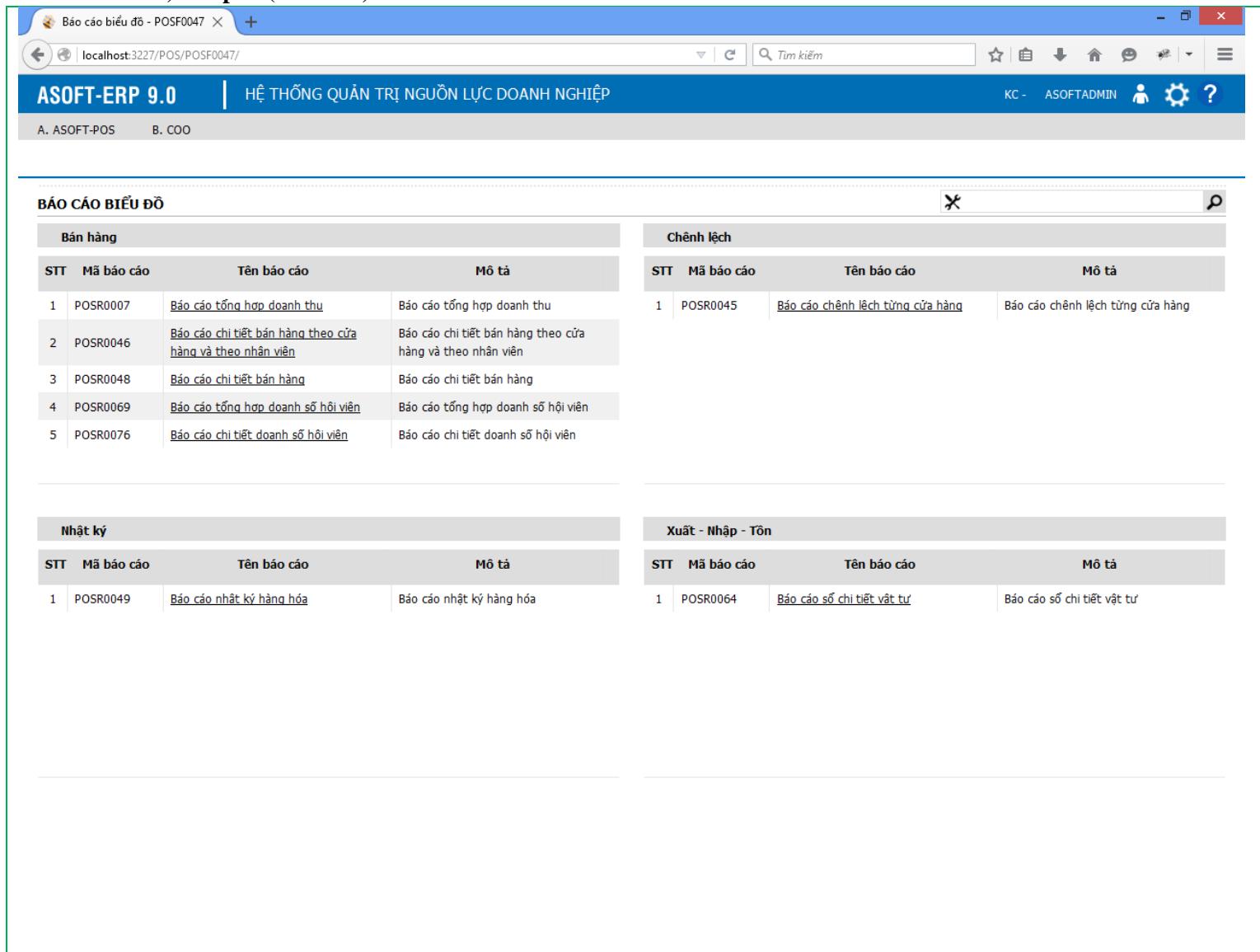
VD: Thêm hành động gửi mail (SendMail) và hành động Thêm sự kiện (AddEvent) cho màn hình xem chi tiết đầu mối (CRMF2032)

```

Update sysScreen
set
sysActionID = (select top 1 convert(varchar(50),sysActionID) from sysAction where
ActionID = 'AddEvent') +',(select top 1 convert(varchar(50),sysActionID) from sysAction
where ActionID = 'SendMail')
where ModuleID = N'AsoftCRM' and ScreenID = N'CRMF2032'

```

7) Report(Báo cáo)



The screenshot shows the ASOFT-ERP 9.0 application window. At the top, it displays 'Báo cáo biểu đồ - POSF0047' and the URL 'localhost:3227/POS/POSF0047/'. The main title is 'HỆ THỐNG QUẢN TRỊ NGUỒN LỰC DOANH NGHIỆP'. The navigation bar includes 'KC - ASOFTADMIN', user icons, and a gear icon for settings.

The interface is divided into several sections:

- BÁO CÁO BIỂU ĐỒ**: This section contains four tables:
 - Bán hàng**: Shows sales reports with columns for STT, Mã báo cáo, Tên báo cáo, and Mô tả. Examples include 'Báo cáo tổng hợp doanh thu' and 'Báo cáo chi tiết bán hàng theo cửa hàng và theo nhân viên'.
 - Chênh lệch**: Shows profit margin reports with similar columns.
 - Nhật ký**: Shows log reports.
 - Xuất - Nhập - Tồn**: Shows inventory and purchase reports.

a) Cấu trúc

- Mỗi module có một màn hình xem báo cáo, như hình trên
- Trong màn hình xem báo cáo có nhiều GroupReport(Ví dụ: Group bán hàng, group chênh lệch, group Nhật kí và group Xuất – Nhập – Tòn)
- Trong mỗi group có nhiều báo cáo, mỗi báo cáo tương ứng với một dòng trong bảng sysReport và có các field tương ứng trong bảng ReportFilter

b) Tạo các Group Report cho từng module

sysReportGroup	
ID	Khóa tự tăng
Module	Mã phân hệ
GroupID	Mã Group
GroupName	Mã ngôn ngữ của group

- Sử dụng script

```

declare @Module nvarchar(50)
declare @GroupID nvarchar(50)
declare @GroupName nvarchar(500)

set @Module = ''
set @GroupID = ''
set @GroupName = ''

insert into sysReportGroup(Module, GroupID, GroupName)
values(@Module, @GroupID, @GroupName)

```

c) Tạo các Report

sysReport	
ID	Khóa tự tăng
ReportID	Mã phân hệ
ReportName	Mã Group
Description	Mã ngôn ngữ của group
GroupID	Mã Group tham chiếu sang bảng sysReportGroup
Disabled	Các trường quy định trạng thái report
IsDelete	
StoreName	Tên Store
StoreParameter	Tham số, tương tự combobox
SQLstring	Câu truy vấn

- Trong SQLString có thêm biến đặc biệt là @p_s
 Nếu xuất hiện biến @p_s thì sẽ được thay thế bằng điều kiện lọc của tất cả các Field có SpecialFilter = 0
 Cú pháp trong StoreParameter: @para1=@Column1,@p_s
 Cú pháp trong SQLString: select * from Table where key1 =@Colum1 and key2=@DivisionID or key3=@UserName and @p_s
- Trong đó: @Column1 → Biến trên control
 @DivisionID → Biến môi trường
- Sử dụng script

```

declare @ReportID varchar(50)
declare @ReportName nvarchar(50)
declare @Description nvarchar(50)
declare @GroupID nvarchar(50)
declare @Disabled tinyint
declare @IsDelete tinyint
declare @StoreName varchar(20)
declare @StoreParameter varchar(500)
declare @SQLstring nvarchar(1000)

set @ReportID = ''
set @ReportName = ''
set @Description = ''
set @GroupID = ''

```

```

set @Disabled = 0
set @IsDelete = 0
set @StoreName = ''
set @StoreParameter = ''
set @SQLstring = ''

insert into sysReport( ReportID, ReportName, Description, GroupID,
                      Disabled, IsDelete, StoreName, StoreParameter, SQLstring)
values(@ReportID,@ReportName,@Description,@GroupID,
       @Disabled,@IsDelete,@StoreName,@StoreParameter,@SQLstring)

```

d) Tạo các mã màn hình tương ứng với các report(để phân quyền)

- Tạo các màn hình trong bảng sysScreen
- Mỗi màn hình ứng với một báo cáo, một báo cáo có thể được nhiều màn hình sử dụng
- Dùng Script tạo sysScreen như mục 1b) chú ý các trường sau.

sysScreen	
DivisionID	Mã đơn vị
ModuleID	Mã phân hệ
ScreenID	Mã màn hình
ReportID	Mã Report

e) Tạo ReportFilter để quy định các Field trên màn hình lọc

sysReportFilter	
sysReportID	Mã Report
sysFieldID	Mã Field(có thể sử dụng mọi Field đã tồn tại)
SpecialFilter	Điều kiện lọc đặc biệt <ul style="list-style-type: none"> - Khi xuất hiện biến @p_s trong câu truy vấn hoặc store, biến @p_s sẽ được thay thế bằng điều kiện lọc của tất cả các Field có SpecialFilter = 0

- Sử dụng script

```

declare @ReportID varchar(50)
declare @sysFieldID int
declare @SpecialFilter tinyint

set @ReportID = ''
set @sysFieldID = 0
set @SpecialFilter = ''

insert into sysReportFilter(ReportID, sysFieldID, SpecialFilter)
values(@ReportID, @sysFieldID, @SpecialFilter)

```

f) Thêm Template cho các báo cáo có sử dụng nhiều template

sysReportTemplate	
sysReportID	Mã Report
FileName	Tên file Template

- Mặc định File sẽ được lấy từ đường dẫn: `~/Areas/{Mã Module}/Content/Report/{Tên File}`
 Trong đó mã module được lấy từ Group chứa Report hiện tại.
- Sử dụng script

```

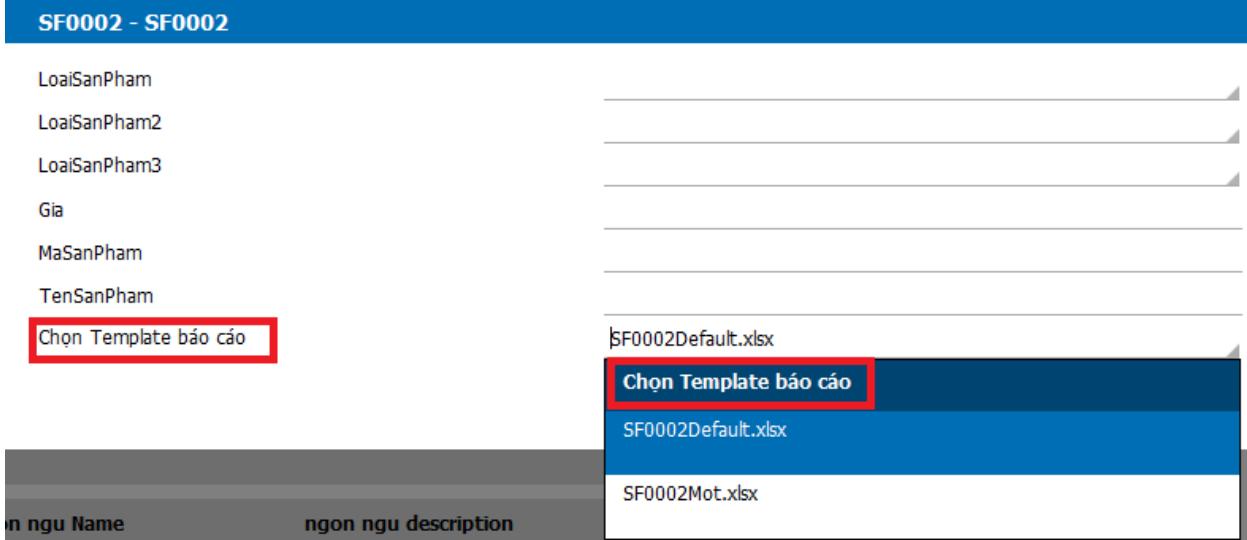
declare @ReportID varchar(50)
declare @FileName varchar(100)

set @ReportID = ''
set @FileName = ''

insert into sysReportTemplate(ReportID, FileName)
values(@ReportID, @FileName)

```

- Khi báo cáo có nhiều template sẽ xuất hiện thêm combobox GetPathTemplate



Để khai báo ngôn ngữ cho ComboBox này(khung đỏ trên hình) ta sử dụng Script

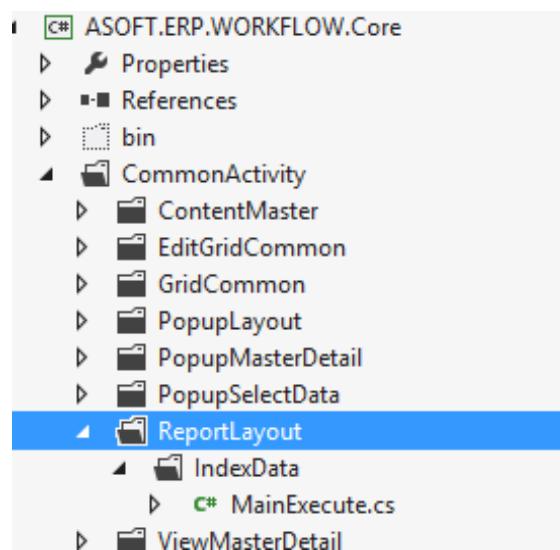
```

DECLARE
@ModuleID VARCHAR(10),
@FormID VARCHAR(200),
@Language VARCHAR(10),
@Name NVARCHAR(4000)
SET @Language = 'vi-VN'
SET @ModuleID = 'S'
SET @FormID = 'SF0002'
SET @Name = N'Chọn Template báo cáo'
insert into
A00001(ID,Deleted,LanguageID,Name,Module,FormID)values('GetPathTemplate',0,@Language,@Name,@ModuleID,@FormID)

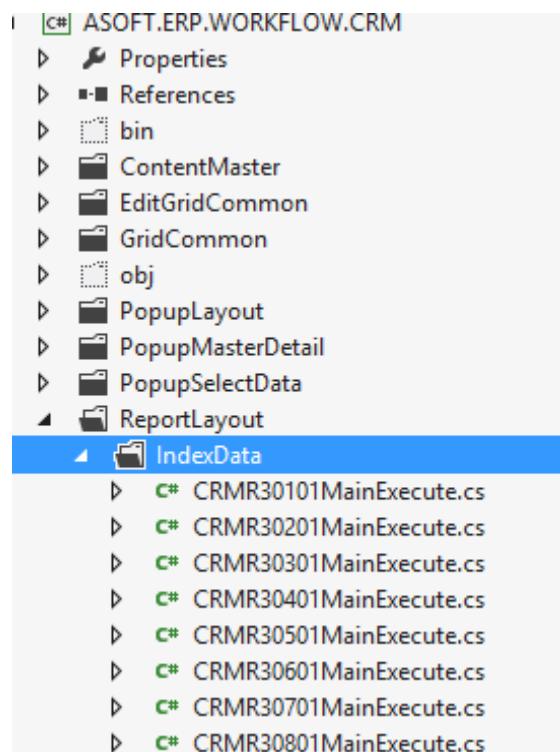
```

g) Custom Report (Layout lại report)

- Custom Report sử dụng luồng workflow **IndexData**:
- **IndexData** nằm trong project **ASOFT.ERP.WORKFLOW.Core -> ReportLayout**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để search dữ liệu trả về controller



- Custom Layout Report:
 - o Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - o Tạo thư mục **ReportLayout** (nếu chưa có)
 - o Tạo thư mục **IndexData** nằm trong thư mục **ReportLayout** (nếu chưa có)
 - o Tạo file class **{Mã Report}** + **MainExecute**



- o Cấu trúc của class custom


```

                public sealed class CRMR30101MainExecute : CodeActivity
                {
            
```

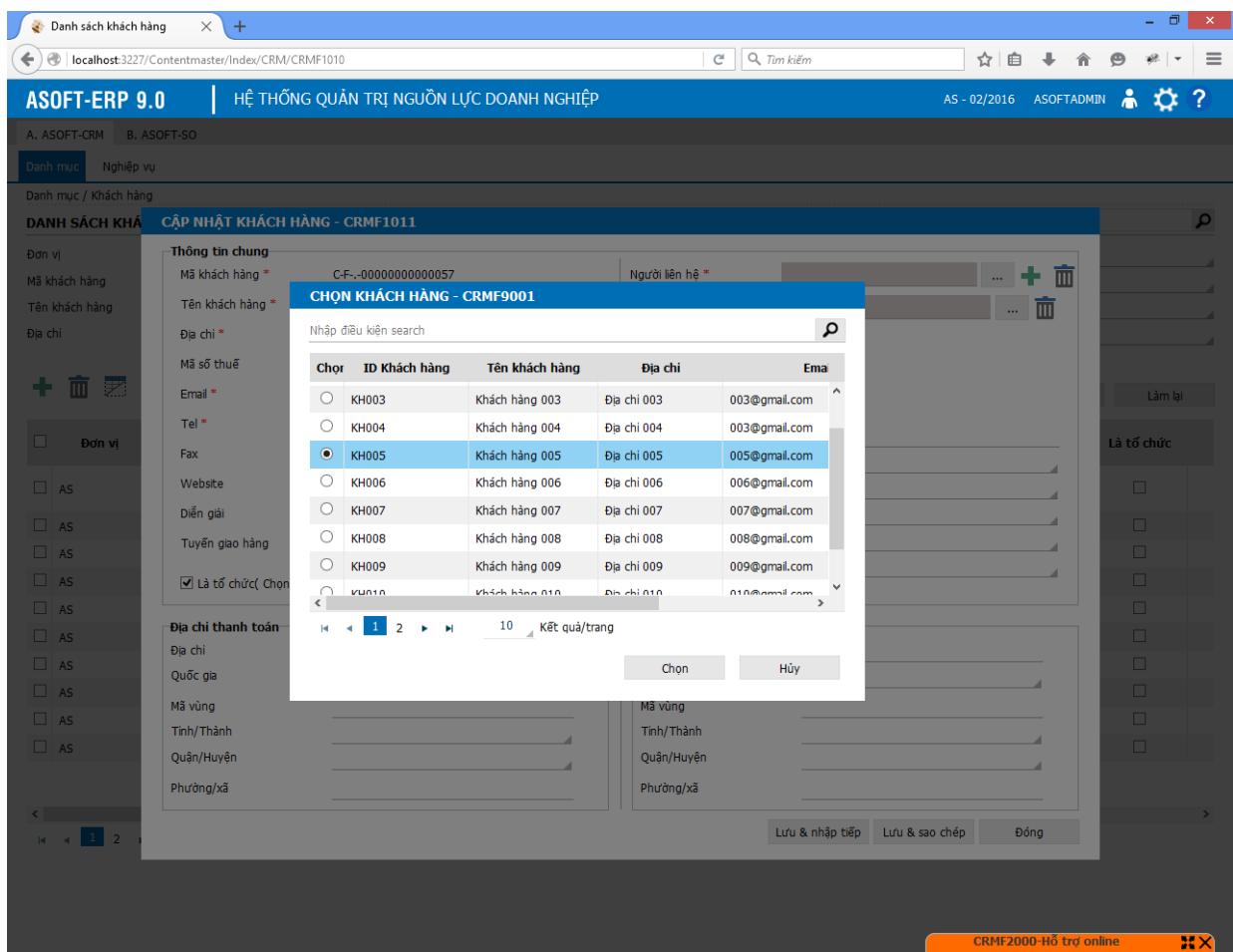
```

-         // Define an activity input argument of type string
-         public InOutArgument<Dictionary<string, Object>> InOut { get;
-             set; }
-         public InOutArgument<string> Area { get; set; }
-         public InOutArgument<string> ScreenID { get; set; }

-
-         // If your activity returns a value, derive from
CodeActivity<TResult>
-         // and return the value from the Execute method.
-         protected override void Execute(CodeActivityContext context)
{
-             Dictionary<string, Object> Input = InOut.Get(context);
-             Dictionary<string, object> model = (Dictionary<string,
object>)Input["Model"];
-             DataSet dataset = (DataSet)Input["DataModel"];
-             string templatePath = (string)Input["templatePath"];
-             DataTable datamodel = dataset.Tables[0];
-             WorkbookDesigner designer = new WorkbookDesigner();
-             designer.Workbook = ASOFTExcel.OpenExcelFile(templatePath);
-             designer.Workbook.CalculateFormula();

-             //Code tại đây, tham khảo các class khác
-             Input.Add("designer", designer);
}
-
```

8) Màn hình chọn (PopupSelectData)



a) Tạo bảng cho màn hình

Nếu bảng có rồi thì không cần tạo lại.

Nếu chưa có, thực hiện thêm 1 dòng vào bảng sysTable:

sysTable	
TableName	Tên bảng (VD: AT002)
Description	Chú thích về bảng
PK	Khóa chính của bảng (VD: APK)
ModuleID	Module của bảng (VD: AsoftS)
RefLink	Trường của bảng để click link trên lưới (Dùng trên lưới, không có để NULL)

b) Tạo màn hình

Thêm 1 dòng vào bảng sysScreen

sysScreen	
APK	Tự sinh động
DivisionID	Đơn vị (VD: AS)
ModuleID	Module của bảng (VD: AsoftCRM)
ScreenID	ID Màn hình (VD: CRMF9008)

ScreenName	Tên màn hình (VD: Chọn chiến dịch)
ScreenType	4: Màn hình khác
ScreenNameE	Tên tiếng anh của màn hình
Parent	Màn hình cha (Không có để NULL)
sysTable	Tham chiếu tới bảng sysTable, bảng mà màn hình thuộc vào (VD: CRMT20401)
Title	Title của màn hình (Mã ngôn ngữ để lấy ra tên màn hình VD: CRMF2040.Title)
TypeInput	1
SQLFilter	Câu truy vấn khi load màn hình
StoreFilter	Tên store khi load màn hình
StoreFilterParameter	Parameter của store

- Các câu SQLFilter, StoreFilter, StoreFilterParameter có cú pháp giống combobox
- Trên màn hình có một Textbox duy nhất với tên @TxtSearch
- Đổi số page và pagesize để phân trang được lưu trong biến @requestPage và @requestPageSize

c) Gọi màn hình từ javascript

- Sử dụng javascript custom để gọi màn hình popupselectdata

```
ASOFT.asoftPopup.showIframe("/PopupSelectData/Index/{Module}/{Mã màn hình}", {});
```

Ví dụ:

```
ASOFT.asoftPopup.showIframe("/PopupSelectData/Index/CRM/CRMF9002", {});
```

- Để lấy dữ liệu dòng được chọn, ta định nghĩa hàm receiveResult trong javascript custom

```
function receiveResult(result) {
    if (ChooseStatus == 1) {
        $("#RouteName").val(result["RouteName"]);
        $("#RouteID").val(result["RouteID"]);
    }
    else if (ChooseStatus == 2) {
        $("#VATAccountName").val(result["AccountName"]);
        $("#VATAccountID").val(result["AccountID"]);
    }
    else if (ChooseStatus == 3 || ChooseStatus == 0) {
        $("#ContactName").val(result["ContactName"]);
        $("#ContactID").val(result["ContactID"]);
    }
};
```

- Để truyền thêm biến vào câu sql ta thêm đối số phía sau đường dẫn khi gọi màn hình

```
ASOFT.asoftPopup.showIframe("/PopupSelectData/Index/CRM/CRMF9002?Bien=Giatri", {});
```

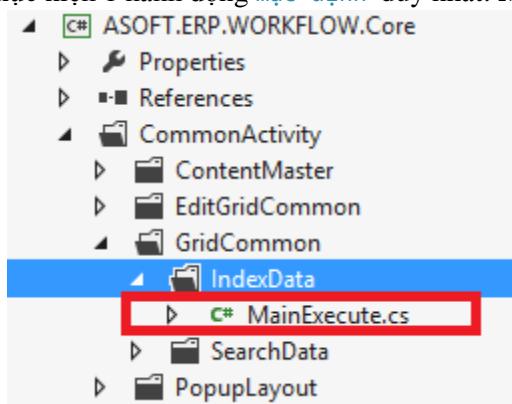
Khi đó trong StoreFilterParameter ta có thể sử dụng: @Bien=@Bien hoặc trong SQLFilter có thể sử dụng Bien=@Bien

d) Customer

Cấu trúc lưới search

- Cấu trúc lưới sử dụng luồng workflow **IndexData**:

- **IndexData** nằm trong project **ASOFT.ERP.WORKFLOW.Core -> GridCommon**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất: **MainExecute**: dùng để cấu trúc lưới.



- Custom lưới:

- o Trường hợp 1: custom cột trên lưới

- Cách custom:

- Tạo file class **{Mã màn hình cha của màn hình chọn} + “Controller”** trong project **ASOFT.ERP.{Module}** -> **Controllers**. (NOTE: Nếu không có màn hình cha sẽ lấy mã của màn hình con)
 - Thêm action **“ChangeFieldsGrid” + (Mã màn hình chọn) + “_” + (Mã table của màn hình chọn)** vào controller.
 - Trong action vừa thêm: code theo cấu trúc:

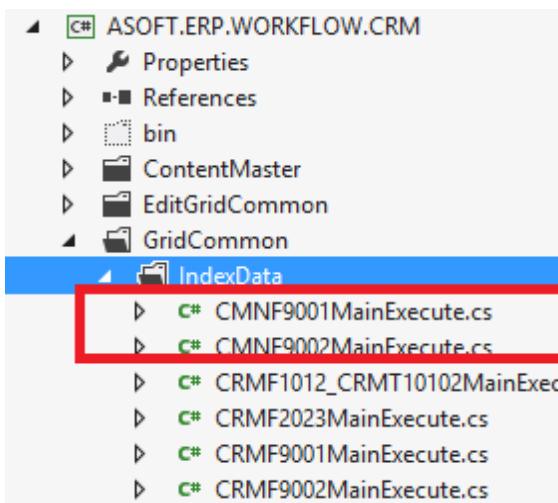
```

- public List<ASOFTSysFields> ChangeFieldsGridCRMF9008_CRMT20401 (List<ASOFTSysFields>
-     list, string id)
- {
-     //Code ở đây, tham khảo các file class khác
-     return list;
- }
```

- o Trường hợp 2: custom toàn bộ lưới:

- Cách custom:

- Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **GridCommon** (nếu chưa có)
 - Tạo thư mục **IndexData** nằm trong thư mục **GridCommon** (nếu chưa có)
 - Tạo file class **{ Mã màn hình chọn } + MainExecute**



- Cấu trúc của class custom

```
- public sealed class CRMF9008MainExecute : CodeActivity
{
    // Define an activity input argument of type string
    public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
    public InOutArgument<string> Area { get; set; }
    public InOutArgument<string> ScreenID { get; set; }

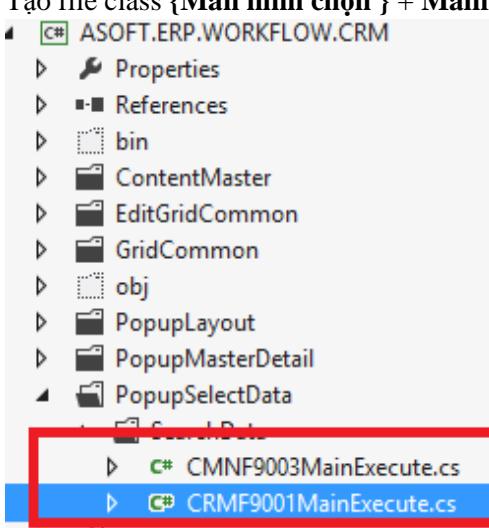
    // If your activity returns a value, derive from CodeActivity<TResult>
    // and return the value from the Execute method.
    protected override void Execute(CodeActivityContext context)
    {
        // Obtain the runtime value of the Text input argument
        string id = ScreenID.Get(context);
        string module = Area.Get(context);
        Dictionary<string, Object> Input = InOut.Get(context);
        List<ASOFTSysFields> list = new List<ASOFTSysFields>();
        ASOFTSysScreen scr = new ASOFTSysScreen();
        List<string> list1 = new List<string>();
        string table = Input["table"].ToString();

        //Code ở đây, thám khảo các class đã làm

        //Các biến bắt buộc phải trả về
        Input.Add("sysFields", list); //List cột trên lưới
        Input.Add("Language", list1); //Ngôn ngữ tương ứng của các cột
        Input.Add("scr", scr); //Màn hình tìm kiếm
    }
}
```

Load lưới detail

- Load lưới detail sử dụng luồng workflow **SearchData**:
 - **SearchData** nằm trong project **ASOFT.ERP.WORKFLOW.Core** ->
 - **PopupSelectData**: tại workflow này thực hiện 1 hành động **mặc định** duy nhất:
 - **MainExecute**: dùng để search dữ liệu trả về controller

- Nếu nghiệp vụ search yêu cầu sử dụng store và câu SQL đặc thù thì ta sẽ custom:
- VD:
 - Cách tạo:
 - Tạo project **ASOFT.ERP.WORKFLOW.{Module}**(nếu chưa có)
 - Tạo thư mục **PopupSelectData** (nếu chưa có)
 - Tạo thư mục **SearchData** nằm trong thư mục **PopupSelectData** (nếu chưa có)
 - Tạo file class **{Màn hình chọn } + MainExecute**
 - 
 - Cấu trúc của class custom

```
public sealed class CRMF9001MainExecute : CodeActivity
{
    // Define an activity input argument of type string
    public InOutArgument<Dictionary<string, Object>> InOut { get; set; }
    public InOutArgument<string> Area { get; set; }
    public InOutArgument<string> ScreenID { get; set; }

    // If your activity returns a value, derive from CodeActivity<TResult>
    // and return the value from the Execute method.
    protected override void Execute(CodeActivityContext context)
    {
        // Obtain the runtime value of the Text input argument
        // Obtain the runtime value of the Text input argument
        Dictionary<string, List<string>> args = (Dictionary<string,
List<string>>)InOut.Get(context)["args"];
        ASOFTSysGrid ds = new ASOFTSysGrid();
        List<string> list = new List<string>();
        List<string> list1 = new List<string>();

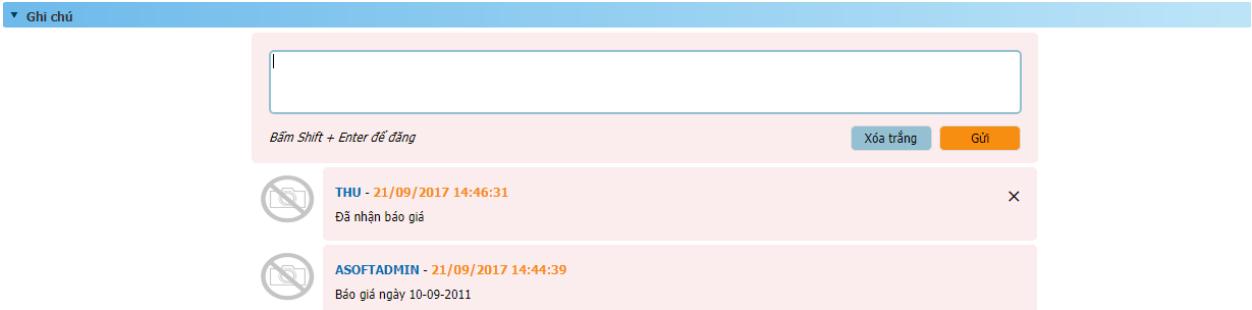
        //Code tại đây, tham khảo các class khác

        Dictionary<string, Object> Ouput = InOut.Get(context);
        Ouput.Add("kq", ds);
    }
}
```

- }

VI) Các tính năng dùng chung

1) Ghi chú



- Group ghi chú nằm trong màn hình xem chi tiết
- Bảng mặc định của ghi chú là CRMT90031, đã mặc định có trong DB Admin nên không cần khai báo.
- Các bước tạo group ghi chú trong màn hình xem chi tiết:

- o Bước 1: Khai báo group ghi chú cho màn hình xem chi tiết trong bảng sysGroup

```
set @GroupName= '{Mã màn hình xem chi tiết}.TabCRMT90031'
set @Description=N'Ghi chú'
set @ModuleID=N'Asoft' + {Module của màn hình xem chi tiết}
set @ScreenID=N'{Mã màn hình xem chi tiết}'
set @TabIndex=N'{Vị trí của group trong màn hình xem chi tiết}'
set @GroupID=N'{Mã màn hình xem chi tiết}.TabCRMT90031'
set @sysTable=N'CRMT90031'
set @PartialView='Notes'
If not exists(select top 1 1 from [dbo].[sysGroup] where [GroupID] = N'{Mã màn
hình xem chi tiết}.TabCRMT90031')Begin
insert into
sysGroup(GroupName,Description,ModuleID,ScreenID,TabIndex,GroupID,sysTable,Parti
alView)values(@GroupName,@Description,@ModuleID,@ScreenID,@TabIndex,@GroupID,@sy
sTable,@PartialView)
End
```

- o Bước 2: Khai báo bảng cha cho bảng ghi chú (CRMT90031).

```
update sysTable set ParentTable = ParentTable + ',{bảng của màn hình xem chi
tiết}' where TableName = 'CRMT90031'
```

- o Bước 3: Khai báo trường TypeREL trong bảng sysTable cho bảng của màn hình xem chi tiết. Trường TypeREL tương ứng với trường RelatedTypeID trong của DB khách hàng. Trường này do BA cung cấp.

```
update sysTable set TypeREL = '{RelatedTypeID}' where TableName = '{Màn hình
xem chi tiết}'
```

2) Lịch sử

Lịch sử					
STT	Trạng thái	Nội dung chỉnh sửa	Người tạo	Ngày tạo	
1	Chỉnh sửa	Xóa dữ liệu - Ghi chú: hehe	HOANG	21/09/2017 15:10:20	
2	Chỉnh sửa	Tạo mới Ghi chú: hehe	HOANG	21/09/2017 15:10:14	
3	Chỉnh sửa	Xóa dữ liệu - Ghi chú: hehe	HOANG	21/09/2017 15:10:04	
4	Chỉnh sửa	Tạo mới Ghi chú: hehe	HOANG	21/09/2017 15:10:02	
		Thay đổi dữ liệu Cá nhân tự đánh giá - Điểm đánh giá lại: 0.00 -> 79.84			

- Group lịch sử nằm trong màn hình xem chi tiết
- Bảng mặc định của lịch sử là CRMT00003, đã mặc định có trong DB Admin nên không cần khai báo.
- Các bước tạo group lịch sử trong màn hình xem chi tiết:
 - o Bước 1: Khai báo group lịch sử cho màn hình xem chi tiết trong bảng sysGroup

```
set @GroupName='Mã màn hình xem chi tiết'.TabCRMT00003'
set @Description=N'Lịch sử'
set @ModuleID=N'Asoft' + Module của màn hình xem chi tiết
set @ScreenID=N'Mã màn hình xem chi tiết'
set @TabIndex=N'Vị trí của group trong màn hình xem chi tiết'
set @GroupID=N'Mã màn hình xem chi tiết'.TabCRMT00003'
set @sysTable=N'CRMT00003'
set @PartialView=null
If not exists(select top 1 1 from [dbo].[sysGroup] where [GroupID] = N'Mã màn
hình xem chi tiết'.TabCRMT00003')Begin
insert into
sysGroup(GroupName,Description,ModuleID,ScreenID,TabIndex,GroupID,sysTable,Parti
alView)values(@GroupName,@Description,@ModuleID,@ScreenID,@TabIndex,@GroupID,@sy
sTable,@PartialView)
End
```

- o Bước 2: Khai báo bảng cha cho bảng lịch sử (CRMT00003).

```
update sysTable set ParentTable = ParentTable + ',{bảng của màn hình xem chi
tiết}' where TableName = 'CRMT00003'
```

- o Bước 3: Khai báo trường TypeREL trong bảng sysTable cho bảng của màn hình xem chi tiết. Trường TypeREL tương ứng với trường RelatedTypeID trong của DB khách hàng. Trường này do BA cung cấp.

```
update sysTable set TypeREL = '{RelatedTypeID}' where TableName = '{Màn hình
xem chi tiết}'
```

3) Gửi mail

a) Group gửi mail

▼ Email					
STT	Tiêu đề	Người gửi	Người tạo	Ngày tạo	
Không có dữ liệu					

- Group email nằm trong màn hình xem chi tiết
- Bảng mặc định của email là CMNT90051, đã mặc định có trong DB Admin nên không cần khai báo.
- Các bước tạo group email trong màn hình xem chi tiết:
 - o Bước 1: Khai báo group email cho màn hình xem chi tiết trong bảng sysGroup

```
set @GroupName='{Mã màn hình xem chi tiết}.TabCMNT90051'
set @Description=N'Email'
set @ModuleID=N'Asoft' + {Module của màn hình xem chi tiết}
set @ScreenID=N'{Mã màn hình xem chi tiết}'
set @TabIndex=N'{Vị trí của group trong màn hình xem chi tiết}'
set @GroupID=N'{Mã màn hình xem chi tiết}.TabCMNT90051'
set @sysTable=N'CMNT90051'
set @PartialView=null
If not exists(select top 1 1 from [dbo].[sysGroup] where [GroupID] = N'{Mã màn
hình xem chi tiết}.TabCMNT90051')Begin
insert into
sysGroup(GroupName,Description,ModuleID,ScreenID,TabIndex,GroupID,sysTable,Parti
alView)values(@GroupName,@Description,@ModuleID,@ScreenID,@TabIndex,@GroupID,@sy
sTable,@PartialView)
End
```

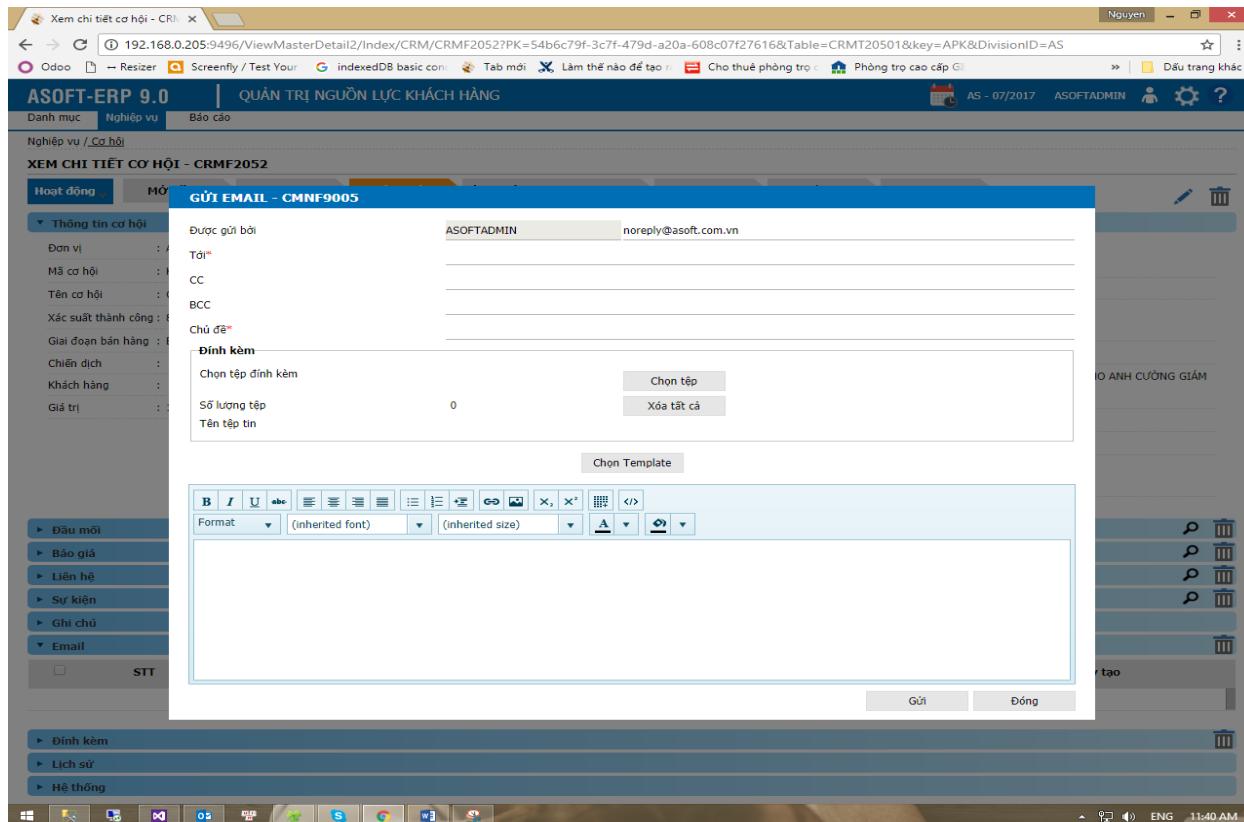
- o Bước 2: Khai báo bảng cha cho bảng email (CMNT90051).

```
update sysTable set ParentTable = ParentTable + ',{bảng của màn hình xem chi
tiết}' where TableName = 'CMNT90051'
```

- o Bước 3: Khai báo trường TypeREL trong bảng sysTable cho bảng của màn hình xem chi tiết. Trường TypeREL tương ứng với trường RelatedTypeID trong của DB khách hàng. Trường này do BA cung cấp.

```
update sysTable set TypeREL = '{RelatedTypeID}' where TableName = '{Màn hình
xem chi tiết}'
```

b) Màn hình gửi mail (CMNF9005)



Sử dụng màn hình gửi mail trong màn hình xem chi tiết.

- Khai báo sysAction trong bảng sysScreen.

```
update sysScreen set sysAction = (select top 1 convert(varchar(50),sysActionID) from sysAction where ActionID = 'SendMail') where ScreenID = 'Mã màn hình xem chi tiết'
```

- Sau khi khai báo sẽ hiển thị:

Danh mục [Nghiệp vụ](#) [Báo cáo](#)

Nghiệp vụ / [Cơ hội](#)

XEM CHI TIẾT CƠ HỘI - CRMF2052

Hoạt động **MỞ ĐẦU** **DEMO** **BÁO GIÁ** **ĐÀM PHÁN - THƯƠNG LƯƠNG** **Hủy** **Thắng** **Thua**

Chuyển đổi sang khách hàng

Gửi mail 1-0012020

Đính kèm NHÂN SÁNG TẠO

Thêm sự kiện

Tạo liên hệ

Tạo đầu mối .000

Gửi báo giá

Độ ưu tiên : Thấp

Lý do kết thúc :

Ghi chú : 0918445251 - ANH CƯỜNG GIÁM ĐỐC

Người phụ trách : Trần Thanh Duy

Nguồn đầu mối : NHÂN VIÊN ĐIỆN THOẠI

Ngày bắt đầu : 23/08/2017

Ngày dự kiến kết thúc : 30/08/2017

Hành động : Gọi điện demo

Ngày thực hiện hành động : 23/08/2017

Từ khóa :

Hành động thêm vào : Cố vào lịch

Không hiển thị : Không

Nội dung thêm vào : DEMO ONLINE POS - CHO ANH CƯỜNG GIÁM ĐỐC 0918445251

Dùng chung : Không

Vùng/Khu vực : Khu vực miền nam

Ngành nghề kinh doanh : Dịch vụ doanh

Đầu mối **Báo giá** **Liên hệ** **Sự kiện** **Ghi chú** **Email** **Đính kèm**

Sử dụng màn hình gửi mail ở chỗ khác:

Sử dụng javascript:

```
var urlLink = "/SendMail?RelatedToID=" + {khóa chính của đối tượng gửi mail} +
"&RelatedToTypeID_REL=" + {RelatedToTypeID của đối tượng gửi mail};  

ASOFT.asoftPopup.showIframe(urlLink, {});
```

4) Đính kèm

a) Group đính kèm

Đính kèm					
	STT	Tên file	Người tạo	Ngày tạo	
	1	CRMF1020-Danh mục nguồn đầu mối 01.png	VU	22/08/2017 09:03:43	
	2	CRMF1032-Xem chi tiết nhóm nhân viên 02.png	VU	22/08/2017 09:04:30	
	3	CRMF1032-Xem chi tiết nhóm nhân viên 02.png	VU	22/08/2017 09:03:35	
	4	CRMF2051-Cập nhật cơ hội 01.png	VU	22/08/2017 09:05:11	
	5	FrameWorkERP9.0.pptx	ASOFTADMIN	11/08/2017 10:34:52	
	6	KPI 2017 - Sales Dept - Inopak,Inoco,ArtDre - R.xls	NAM	19/09/2017 13:47:05	
	7	SOR300-Report.png	VU	22/08/2017 09:06:17	

- Group đính kèm nằm trong màn hình xem chi tiết
- Bảng mặc định của đính kèm là CRMT00002, đã mặc định có trong DB Admin nên không cần khai báo.
- Các bước tạo group đính kèm trong màn hình xem chi tiết:
 - o Bước 1: Khai báo group đính kèm cho màn hình xem chi tiết trong bảng sysGroup

```

set @GroupName= '{Mã màn hình xem chi tiết}.TabCRMT00002'
set @Description=N'Dính kèm'
set @ModuleID=N'Asoft' + {Module của màn hình xem chi tiết}
set @ScreenID=N'{Mã màn hình xem chi tiết}'
set @TabIndex=N'{Vị trí của group trong màn hình xem chi tiết}'
set @GroupID=N'{Mã màn hình xem chi tiết}.TabCRMT00002'
set @sysTable=N'CRMT00002'
set @PartialView=null
If not exists(select top 1 1 from [dbo].[sysGroup] where [GroupID] = N'{Mã màn
hình xem chi tiết}.TabCRMT00002')Begin
insert into
sysGroup(GroupName,Description,ModuleID,ScreenID,TabIndex,GroupID,sysTable,Parti
alView)values(@GroupName,@Description,@ModuleID,@ScreenID,@TabIndex,@GroupID,@sy
sTable,@PartialView)
End

```

- Bước 2: Khai báo bảng cha cho bảng đính kèm (CRMT00002).

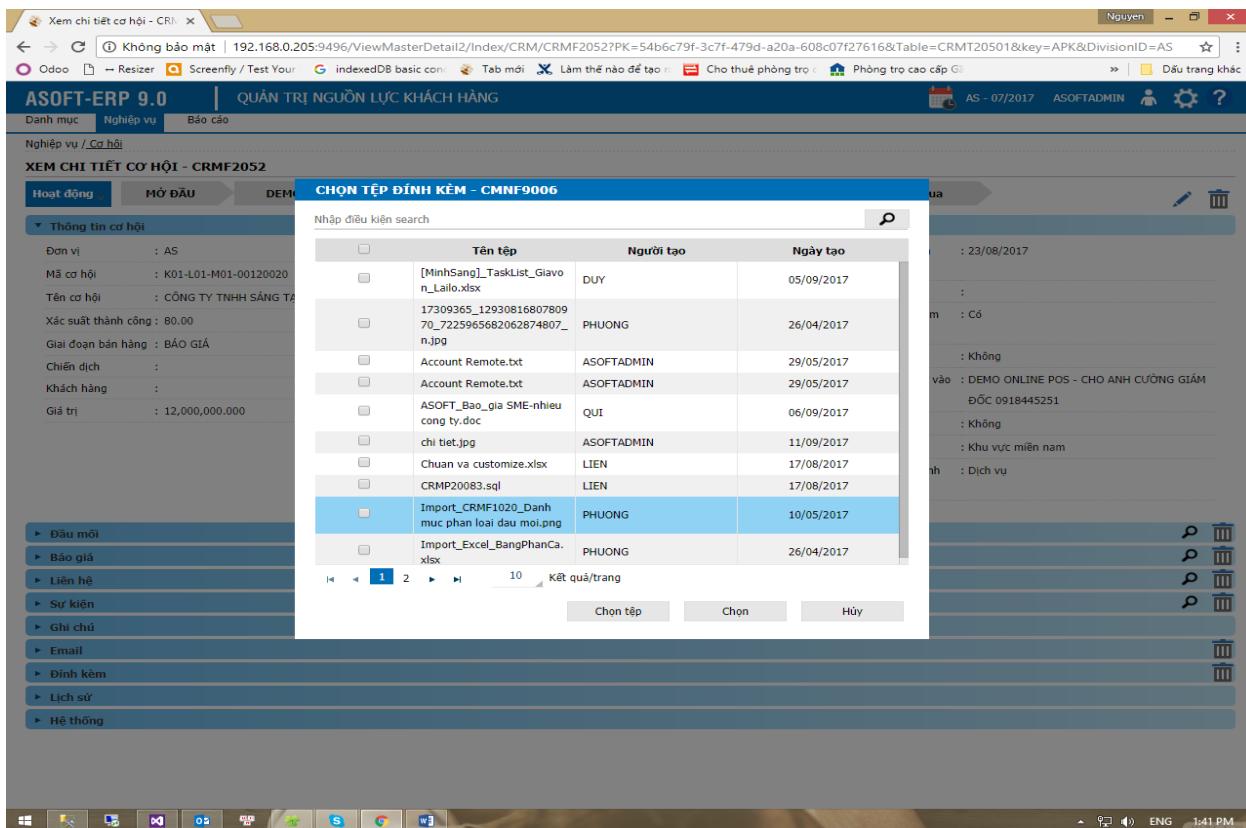
```
update sysTable set ParentTable = ParentTable + ',{bảng của màn hình xem chi
tiết}' where TableName = 'CRMT00002'
```

- Bước 3: Khai báo trường TypeREL trong bảng sysTable cho bảng của màn hình xem chi
tiết. Trường TypeREL tương ứng với trường RelatedTypeID trong của DB khách
hàng. Trường này do BA cung cấp.

```
update sysTable set TypeREL = '{RelatedTypeID}' where TableName = '{Màn hình
xem chi tiết}'
```

b) Màn hình đính kèm

Sử dụng màn hình đính kèm trong màn hình xem chi tiết:



- Khai báo sysAction trong bảng sysScreen.

```
update sysScreen set sysAction = (select top 1 convert(varchar(50),sysActionID) from sysAction where ActionID = 'Attach') where ScreenID = 'Mã màn hình xem chi tiết'
```

- Sau khi khai báo sẽ hiển thị:

Danh mục [Nghiệp vụ](#) [Báo cáo](#)

Nghiệp vụ / [Cơ hội](#)

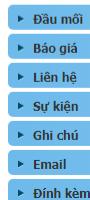
XEM CHI TIẾT CƠ HỘI - CRMF2052

Hoạt động **MỞ ĐẦU** **DEMO** **BÁO GIÁ** **ĐÀM PHÁN - THƯƠNG LƯƠNG** **Hủy** **Thắng** **Thua**



Chuyển đổi sang khách hàng	
Gửi mail	01-00120020
Đính kèm	NHH SÁNG TẠO
Thêm sự kiện	
Tạo liên hệ	
Tạo đầu mối	.000
Gửi báo giá	

Độ ưu tiên : Thấp
Lý do kết thúc :
Ghi chú : 0918445251 - ANH CƯỜNG GIÁM ĐỐC
Người phụ trách : Trần Thanh Duy
Nguồn đầu mối : NHÂN VIÊN ĐIỆN THOẠI
Ngày bắt đầu : 23/08/2017
Ngày dự kiến kết thúc : 30/08/2017
Hành động : Gọi điện demo
Ngày thực hiện hành động : 23/08/2017
Từ khóa :
Hành động thêm vào : CÓ
vào lịch
Không hiển thị : Không
Nội dung thêm vào : DEMO ONLINE POS - CHO ANH CƯỜNG GIÁM ĐỐC 0918445251
Dùng chung : Không
Vùng/Khu vực : Khu vực miền nam
Ngành nghề kinh doanh : Dịch vụ



Sử dụng màn hình đính kèm ở chỗ khác:

Sử dụng javascript:

```
var urlLink = "/PopupLayout/Index/CRM/CRMF9003?DivisionID=" + {DivisionID của đối tượng gửi mail} + "&RelatedTypeID=" + {RelatedTypeID của đối tượng gửi mail} + "&RelatedToID=" + {Khóa chính của đối tượng gửi mail}
ASOFT.asoftPopup.showIframe(urlLink, {});
```

VII) Khai báo ngôn ngữ

1) Khai báo ngôn ngữ trong DBAdmin

- Đối với các field khi khai báo ngôn ngữ cần thêm ngôn ngữ vào bảng sysLanguage

```
DECLARE
@FormID VARCHAR(200),
@ColumnName VARCHAR(200)
SET @FormID = 'CMNF9003';
SET @ColumnName = N'EmployeeID';
EXEC ADDLanguageAdmin 'CMNF9003.EmployeeID' , @ColumnName, @FormID;
```

- Sau đó tạo dữ liệu ngôn ngữ trong DB khách hàng với key đã được thêm ở trên

2) Tạo dữ liệu trong DataBase Khách hàng

a) Thêm ngôn ngữ mới cho các [control chung]

- Control chung : Button, Menu, khác, ...
- Thêm vào A0001.
- **Module : 00, FormID: "A00", ID : Theo phân tích**

- Tạo script execute store **ERP9AddLanguage** Insert/Update ngôn ngữ.

```
DECLARE
@ModuleID VARCHAR(10),
@FormID VARCHAR(200),
@Language VARCHAR(10),
@ControlName NVARCHAR(4000),
SET @Language = 'vi-VN'
SET @ModuleID = '00';
SET @FormID = 'A00';
SET @ControlName = N'Lưu & nhập tiếp';
EXEC ERP9AddLanguage @ModuleID, 'A00.btnSaveNew' , @FormID, @ControlName,
@Language;
```

b) Thêm ngôn ngữ mới cho một màn hình

- Thêm khóa cho tất cả các label trên màn hình
- Module: Phân hệ của màn hình, FormID: “Tên màn hình”, ID: Theo phân tích, Name: Tên
- Tạo script execute store **ERP9AddLanguage** Insert/Update ngôn ngữ.

```
DECLARE
@ModuleID VARCHAR(5),
@FormID VARCHAR(50),
@Language VARCHAR(10),
@LanguageValue NVARCHAR(4000),
@LanguageCustomValue NVARCHAR(4000),
SET @Language = 'vi-VN';
SET @ModuleID = 'MT';
SET @FormID = 'MTF1000';
SET @LanguageValue = N'Từ ngày';
SET @LanguageCustomValue = N'';
EXEC ERP9AddLanguage @ModuleID, 'MTF1000.BeginDate' , @FormID, @LanguageValue,
@Language, @LanguageCustomValue;
```

c) Thêm message mới

- Thêm một khóa vào bảng A00002
- Tạo script execute store **ERP9AddMessage** Insert/Update message.

```
DECLARE
@ModuleID VARCHAR(10),
@FormID VARCHAR(200),
@Language VARCHAR(10),
@MessageValue NVARCHAR(400),
SET @Language = 'vi-VN'
SET @ModuleID = '00';
SET @FormID = 'A00';
SET @MessageValue = N'Bạn có muốn lưu?';
```

```
EXEC ERP9AddMessage @ModuleID, 'A00ML000001' , @MessageValue, @Language;
```

d) Thêm ngôn ngữ hay message dùng Storeprocedure

- Sử dụng storeprocedure **ERP9AddLanguage** để thêm ngôn ngữ hay message.
- Sử dụng :

```
SET @LanguageValue = N'Hệ thống';
EXEC ERP9AddLanguage @ModuleID, 'POSF0000.System' , @FormID,
@LanguageValue, @Language;
```

e) Qui định đặt các file message và language trên SVN

- Message : 10_DOCUMENT\14_DATABASE\7.DATA_MASTER\02.Message\{Module}
- Language: 10_DOCUMENT\14_DATABASE\7.DATA_MASTER\01.Language\{Module}
- Các script của DBAdmin để trong thư mục: 10_DOCUMENT\14_DATABASE\#ADMIN

3) Khai báo trong Project

a) Message JavaScript

- Tự động gen ra file javascript khi khởi động IIS
- Có kiểm tra version khi thay đổi trong A00001
- Có tự chuyển file message theo ngôn ngữ
- Sử dụng trong javascript:

```
asoftNamespace.helper.getMessage(Id);
Ví dụ : var message = ASOFT.helper.getMessage('AFML000001');
alert(message);
```

b) Hiển thị ngôn ngữ cho control chung, menu (trong .cshtml)

- Lưu trong cache khi khởi động IIS
- Chưa kiểm tra version khi thay đổi A00001 (TODO)
- Màn hình sinh động khi ngôn ngữ đã được thêm vào database.

c) Hiển thị ngôn ngữ khi cho một màn hình (trong ViewModel và .cshtml)

- Lưu trong cache khi chọn một màn hình.
- Nếu update script khi release thì phải restart lại IIS.
- Màn hình sinh động khi ngôn ngữ đã được thêm vào database.

VIII) Sử dụng Custom JavaScript và Custom CSS

1) Javascript custom

- Khai báo javascript trong thư mục Scripts/JavaCustomize/{Module}/{Tên File}
- Tên file có dạng ScreenID + {chuỗi tự chọn} + .js (Ví dụ CRMF2023_1.js, CRMF2023_2.js, trong đó _1 và _2 là chuỗi tự chọn)

a) Bind và Unbind sự kiện các button chuẩn

- Sử dụng bind và unbind của kendo để tạo lại sự kiện cho các button chuẩn

```
$("#BtnHide").unbind();  
  
$("#BtnHide").kendoButton({  
    "click": CustomBtnHide_Click,  
});
```

2) CSS custom

- Khai báo trong Content/CSSCustomize/{Module}/{Tên File}
- Tên file tương tự javascript custom