



asoft

Growing Together

CODING GUIDELINE

Người tạo: Lê Minh Lâm

Theo dõi sửa đổi

[illegible]

TÀI LIỆU MÔ TẢ CODING GUIDELINE

---oOo---

I)	Gới thiệu	5
1)	Tổng quan	5
2)	Môi trường phát triển	5
3)	Cấu trúc framework	6
4)	Cấu trúc solution	6
II)	Các quy định khi phát triển chương trình	14
1)	Các module trong chương trình	14
2)	Quy định cách đặt Namespace	14
1.1)	Namespace cho Project Web	14
1.2)	Namespace cho Project Entities	14
1.3)	Namespace cho Project DataAccess	14
1.4)	Namespace cho Project Business	14
3)	Quy định cách đặt tên	15
3.1)	Project Web	15
3.2)	Project Entities	16
3.3)	Project DataAccess	16
3.4)	Project Business	17
III)	Các bước khai báo ngôn ngữ	17
1)	Cấu trúc bảng lưu ngôn ngữ	17
2)	Tạo dữ liệu cho database	18
1.1)	Thêm ngôn ngữ mới cho các [control chung]	19
1.2)	Thêm ngôn ngữ mới cho các một màn hình	19
1.3)	Thêm message mới	19
3)	Khai báo trong project	19
1.1)	Message javascript	19
1.2)	Hiển thị ngôn ngữ cho control chung, menu (trong .cshtml)	20
1.3)	Hiển thị ngôn ngữ khi chọn một màn hình (trong ViewModel và .cshtml)	20
IV)	Quy định chuẩn Encoding khi lưu file	20
V)	Hướng dẫn xử lý code	20
1)	Xử lý SQL	20
1.1)	SQL liên quan đến một Table	20
1.2)	SQL liên quan đến một Store Procedure	21
1.3)	SQL kết hợp bởi nhiều Table	21
1.4)	Xử lý SQL cộng chuỗi trong code :	22
1.5)	Xử lý parameter trong SQL:	23
1.6)	Một số trường hợp đặc biệt:	26
1.7)	Xử lý ngày trong SQL	27
2)	Cách xử lý try/ catch	27
2.1)	Project Web	27
2.2)	Project Business	27
2.3)	Project DataAccess	28
2.4)	Project Entities	28
3)	Cách sử dụng biến hệ thống	28
4)	Danh sách các hàm hỗ trợ layout và cách dùng:	28
4.1)	Cách dùng	28

4.2) Hỗ trợ tạo các control thông thường:	29
4.3) Hỗ trợ các control đặc biệt:	32
4.4) Control GridView	33
4.5) Menu	36
4.6) MultiSelectBoxItem - Control chọn nhiều Item	39
4) Các hàm hỗ trợ javascript	41
5) Sử dụng Toolbar	45
5.1) File *.CSHTML	45
5.2) *ViewModel.cs	45
5.3) Thêm button cho toolbar	45
6) Cách sử dụng AutoMapper	45
6.1) Chuyển từ ViewModel sang Entity	46
6.2) Chuyển từ Entity sang ViewModel	46
7) Cách sử dụng Validate	46
7.1) Code ViewModel	46
7.2) Code javascript	47
8) Thiết lập cột ẩn trong View	48
9) Quản lý session	49
VI) Project Tool gen code	50
1) Mục đích	50
2) Mô tả xử lý	50
a. Thiết lập	50
b. Generate code cho stored procedure	50
c. Generate code cho table	51
d. Generate code cho màn hình	51




asoft
Growing Together

I) Giới thiệu

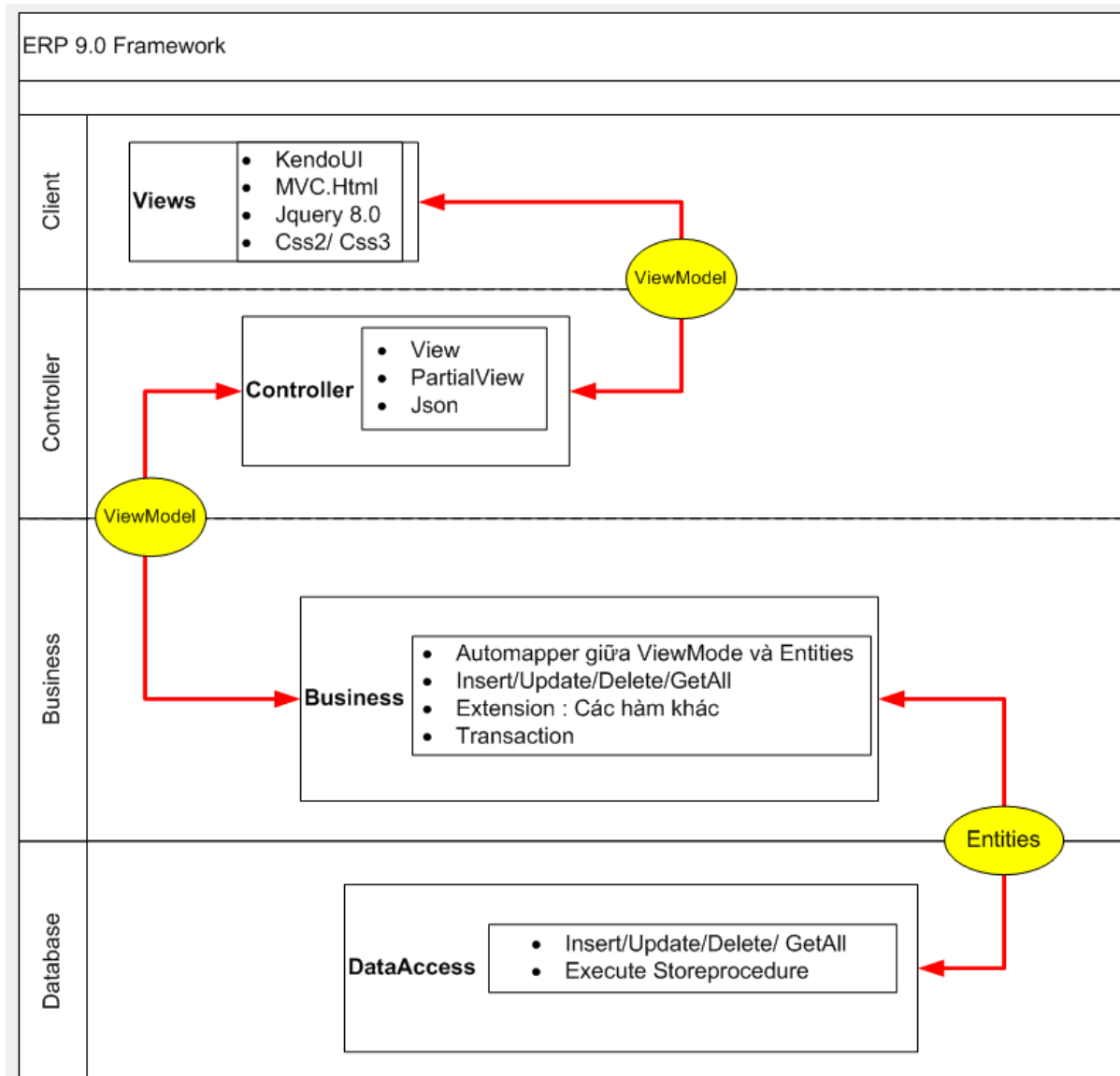
1) Tổng quan

- ❖ Tài liệu cho người phát triển có cái nhìn chung về Framework 9.0
 - Cấu trúc, ý nghĩa thư mục
 - Quy tắc đặt tên
 - Tool gen database
- ❖ 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, JQuery-1.8.3
Database	SQL Server 2012
Mô hình sử dụng	MVC4

3) Cấu trúc framework



4) Cấu trúc solution



A00

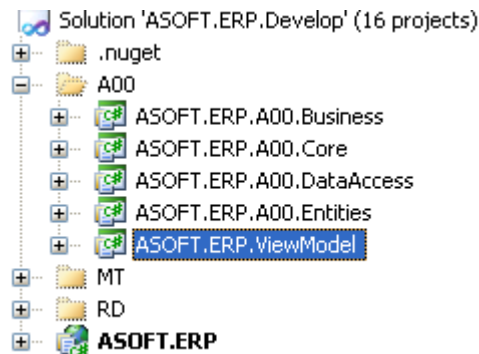
Các thư viện dành cho toàn hệ thống

MT, RD

Thư viện dành riêng cho phân hệ

ASOFT.ERP

Website chính của hệ thống



ASOFT.ERP.A00.Business

Chứa các hàm thêm/xóa/sửa của một bảng

ASOFT.ERP.A00.DataAccess

Chứa các hàm thêm/xóa/sửa của một bảng

ASOFT.ERP.A00.Core

Các thiết định chung của hệ thống

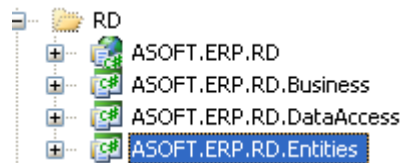
ASOFT.ERP.A00.Entities

Chứa class tham chiếu cho các column của một bảng

ASOFT.ERP.ViewModel

Chứa class tham chiếu cho các item trên màn hình

Các class chuẩn của một phân hệ (bắt buộc phải có các project này)



ASOFT.ERP.RD.DataAccess

Chứa cách hàm thêm/xóa/sửa theo nghiệp vụ

ASOFT.ERP.RD.Business

Chứa cách hàm thêm/xóa/sửa theo nghiệp vụ

Thường là nhiều bảng, transaction

ASOFT.ERP.RD.Entities

Chứa các entities phát sinh theo nghiệp vụ (thường là từ Storeprocedure)

ASOFT.ERP.RD

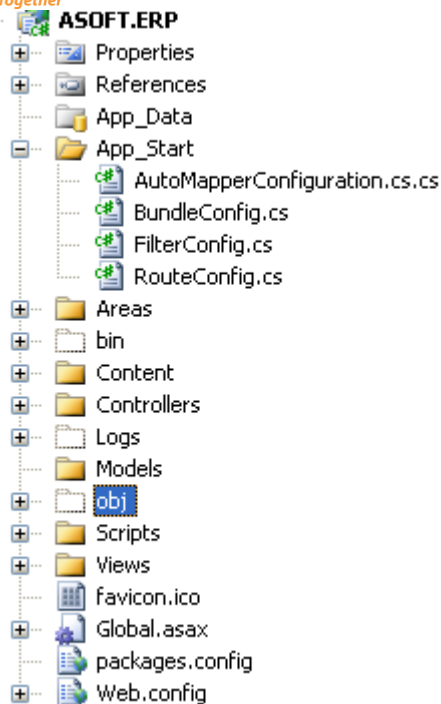
Chứa HTML, CSS, JS của riêng phân hệ

(Đang ví dụ là project RD)

Chú ý khi tạo module mới:

- Project mới phải nằm trong thư mục **AREA** của ASOFT.ERP
- Chỉ build ra ngoài thư mục BIN của ASOFT.ERP
- Đặt tên project theo nguyên tắc : ASOFT.ERP.{Phân Hệ}.{Thư viện}
- Phân hệ mới phải bao gồm tất cả các project trên

asoft
Growing Together



App_Start

Class chung đăng ký resources, controller, module (theo chuẩn của MVC4)

Areas

Chứa source website của các phân hệ (có thể execute khỏi solution)

Là 1 alias của ASOFT.ERP.RD

Content

Chứa Images, CSS

Controllers

Chứa các controller chung của hệ thống như Menu, Combo, ...

Scripts

Chứa thư mục JS của hệ thống

Views

Chứa các view chung của hệ thống như Layout chung, PartialView, Menu,...

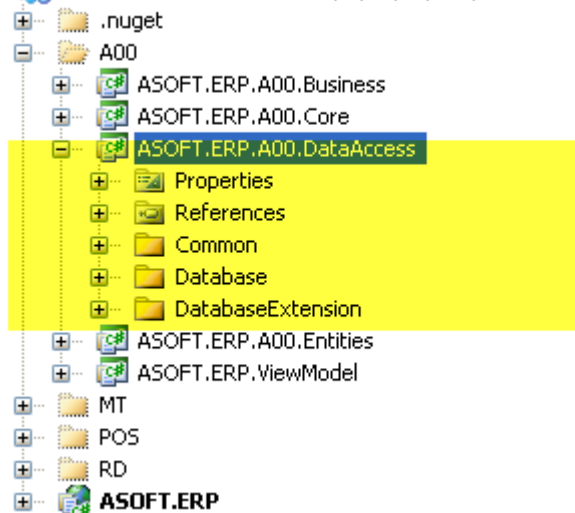
Common

Các xử lý chung của hệ thống như dữ liệu
GetDivisionID(), GetPeriod(**string** divisionId)

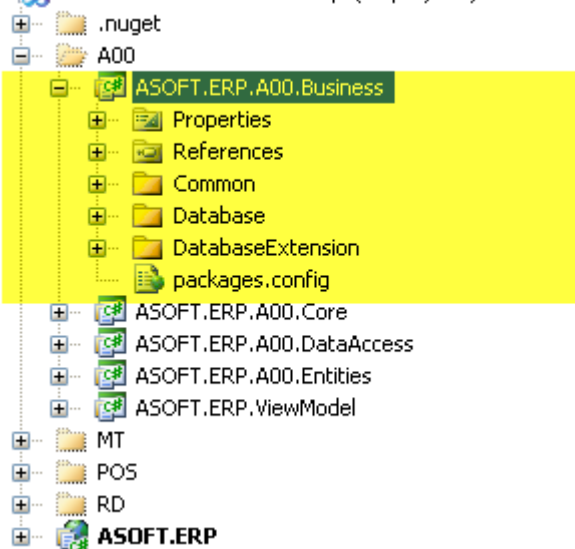
Database

Các class thao tác trực tiếp với database với 1 bảng, class được gen từ tool **VI.Project Tool gen code** → **sẽ bị reset lại khi chạy lại tool**

Solution ASOFT.ERP.Develop (17 projects)



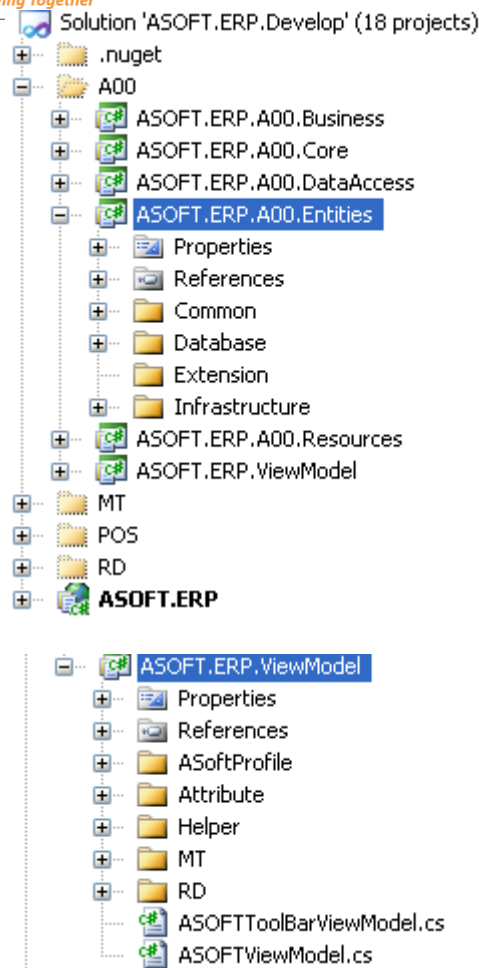
Solution 'ASOFT.ERP.Develop' (17 projects)



DatabaseExtension

Các class thao tác 1 bảng mà **không được** gen từ tool
VI.Project Tool gen code

asoft
Growing Together



Common

Các model phát sinh sử dụng chung cho hệ thống

Database

Các Entity có thông tin giống DB → sử dụng tool
VI.Project Tool gen code

Infrastructure

Các class cơ bản chugn dành cho hệ thống

ASoftProfile

Đăng ký AutoMapper của hệ thống, sẽ chạy khi server được start

Attribute

Các Attribute phát sinh thêm dùng cho ViewModel

MT, RD,...

Các ViewModel của các màn hình trong các phân hệ

ASOFTToolBarViewModel.cs

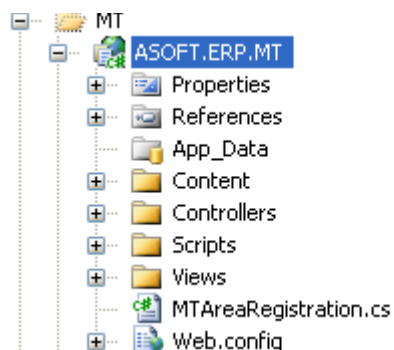
ViewModel của toolbar

ASOFTViewModel.cs

Class cơ sở của ViewModel

Khi thực hiện một phân hệ mới, thì cần phải tạo cấu trúc project có đầy đủ thư mục như sau:

1. Tạo Folder MT
2. Tạo project có tên là ASOFT.ERP.MT
3. Location trong thư mục AREA của ASOFT.ERP
4. Lưu Solution → đóng toàn bộ solution
5. Đổi tên ASOFT.ERP.MT → MT
6. Mở lại Solution và add lại project ASOFT.ERP.MT



Content

Chứa các file css dành riêng cho từng màn hình

Controllers

Các controller cho từng màn hình của phân hệ

Scripts

Các file javascript cho từng màn hình của phân hệ

Views

Các file .cshtml cho từng màn hình của phân hệ

MTAreaRegistration.cs

(Khai báo y hàm như sau)

```
public class MTAreaRegistration : AreaRegistration
{
    public override string AreaName
    {
        get
        {
            return "MT";
        }
    }

    public override void RegisterArea(AreaRegistrationContext context)
    {
        context.MapRoute(
            name: "MT_Default",
            url: "MT/{controller}/{action}/{id}",
            defaults: new
            {
                area = "MT",
                controller = "MTF1000",
                action = "Index",
                id = UrlParameter.Optional
            }
        );
    }
}
```

Thay MT thẳng phân hệ mới

Web.config

Copy từ phân hệ cũ qua

asoft
Growing Together

II) Các quy định khi phát triển chương trình

1) Các module trong chương trình

- **BI** : Thông tin điều hành doanh nghiệp
- **PS** : Giải pháp cảng
- **00** : Module dùng chung trong chương trình
-

2) Quy định cách đặt Namespace

1.1) Namespace cho Project Web

- Namespace chung : **ASOFT.ERP.{Module}**
- Namespace của Controller : **ASOFT.ERP.{Module}.Controllers**

1.2) Namespace cho Project Entities

- Namespace chung : **ASOFT.ERP.{Module}.Entities**
- Namespace của Common (chỉ có ở A00):
ASOFT.ERP.A00.Entities.Common
- Namespace của Database (chỉ có ở A00):
ASOFT.ERP.A00.Entities.Database
- Namespace của Extension (chỉ có ở A00):
 - Là một partial class của entities trong thư mục Database.
 - Mục đích tạo ra: Thêm những properties không có trong file Entities của tool gen ra hoặc cấu trúc table thêm/bớt một số field cho một số khách hàng khác với chuẩn.
 - Tạo ra file Extension với namespace **ASOFT.ERP.A00.Entities.Database** với các properties cần bổ sung.
 - Khởi tạo class: Ví dụ:

```
public partial class AT0001
{
}
```

1.3) Namespace cho Project DataAccess

- Namespace chung : **ASOFT.ERP.{Module}.DataAccess**
- Namespace của Common (chỉ có ở A00):
ASOFT.ERP.A00.DataAccess.Common
- Namespace của Database (chỉ có ở A00):
ASOFT.ERP.A00.DataAccess.Database
- Namespace của Extension (chỉ có ở A00):
ASOFT.ERP.A00.DataAccess.Database

1.4) Namespace cho Project Business

- Namespace chung : **ASOFT.ERP.{Module}.Business**
- Namespace của Common (chỉ có ở A00):
ASOFT.ERP.A00.Business.Common
- Namespace của Database (chỉ có ở A00):
ASOFT.ERP.A00.Business.Database

- Namespace của Extension (chỉ có ở A00):

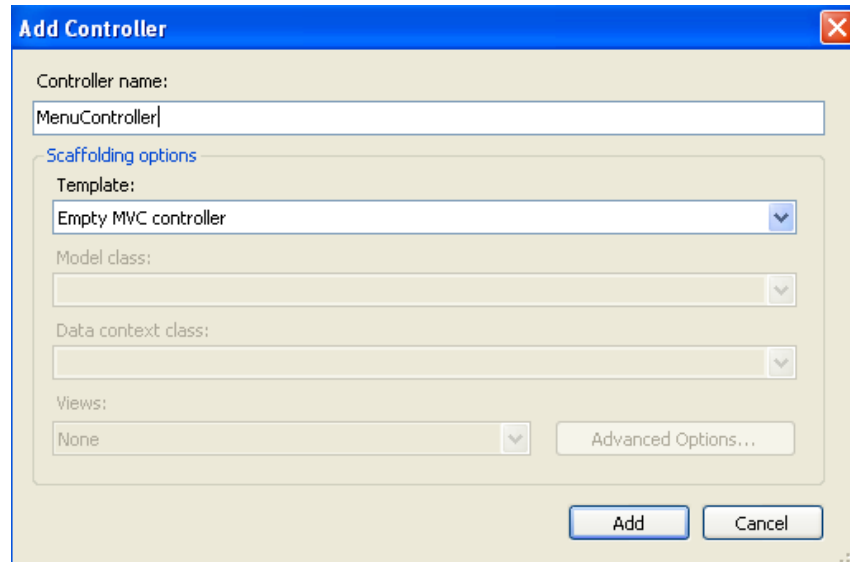
ASOFT.ERP.A00.Business.Database

3) Quy định cách đặt tên

3.1) Project Web

a) Controllers

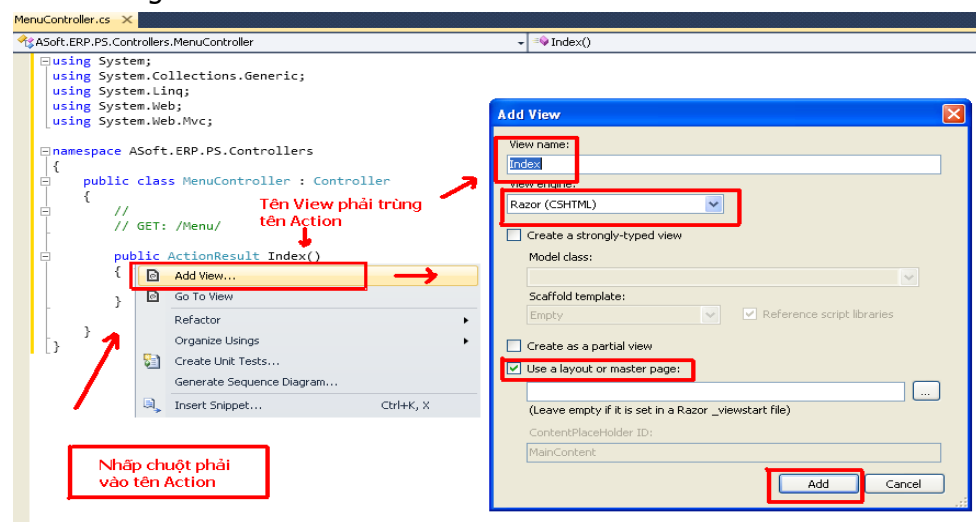
{Tên Controller} + Controller



Ví dụ: MenuController

b) View

- Để tạo View, ta nên tạo Controller trước rồi tạo View sau theo như trong hình bên dưới.



- Đối với những hàm không phải là Action thì ta không tạo View.

Ví dụ: Tạo View cho MenuController như hình vẽ. Sau khi nhấn nút "Add" thì Visual sẽ gen ra cho chúng ta một thư mục là Menu trong thư mục Views mặc định. Trong thư mục Menu sẽ có các file

(* .cshtml) có tên tương ứng với từng Action trong Controller. Ví dụ trong MenuController chỉ có Action "Index" thì sẽ có file "Index.cshtml" tương ứng.

3.2) Project Entities

a) Trong Project dùng chung (A00)

- Trường hợp đặt tên file trong thư mục Common:
{Tên người dùng đặt} + Model
Ví dụ: MenuModel.css
- Trường hợp đặt tên file trong thư mục Database:
Người dùng không cần đặt tên vì tên các file sẽ được gen từ CSDL
Ví dụ: A00001.cs

b) Trong Project khác

- Trường hợp đặt tên file của màn hình thì sẽ đặt tên file theo tên màn hình đang xử lý.
Ví dụ: BF0001.cs
- Trường hợp đặt tên class của control trong màn hình đang xử lý
{Tên màn hình} + {Tên control}
Ví dụ: BF0110ComboBox.cs

3.3) Project DataAccess

a) Trong Project dùng chung (A00)

- Trường hợp đặt tên trong thư mục Common:
{Tên người dùng đặt} + DAL
Ví dụ: ComboBoxDAL.cs
- Trường hợp đặt tên trong thư mục Database:
Người dùng không cần đặt tên vì tên các file sẽ được gen từ CSDL
Ví dụ: A0001DAL.cs
- Trường hợp đặt tên trong thư mục Extension:
{Tên Table} + DAL.extension
Ví dụ: A0001DAL.extension.cs
- Trường hợp đặt tên trong thư mục StoredProcedure:
{Tên Store} + DAL
Ví dụ: AP1409DAL.cs
- Đối với các câu truy vấn sử dụng cho các class ở thư mục Common:
{Tên Class ở thư mục Common} + _{Tên hàm người dùng đặt}
Ví dụ: ComboxDAL_GetByName

- Đối với các câu truy vấn sử dụng cho các class ở thư mục Database:
{Tên Class ở thư mục Database} + _{Tên hàm do người dùng đặt}
Ví dụ: A0001DAL_GetByName
- Đối với các câu truy vấn sử dụng cho các màn hình:
{Tên màn hình} + {Loại control cần gán dữ liệu} + _{Tên hàm do người dùng đặt}
Ví dụ: BF0110ComboBox_GetByDate

b) Trong Project khác

- Đặt tên các file theo màn hình đang xử lý
{Tên màn hình} + DAL
Ví dụ: BF0110DAL.cs

3.4) Project Business

a) Trong Project dùng chung (A00)

- Trường hợp đặt tên trong thư mục Common:
{Tên người dùng đặt} + BL
Ví dụ: ComboBoxBL.cs
- Trường hợp đặt tên trong thư mục Database:
Người dùng không cần đặt tên vì tên các file sẽ được gen từ CSDL
Ví dụ: A0001BL.cs
- Trường hợp đặt tên trong thư mục Extension:
{Tên Table} + BL.extension
Ví dụ: A0001BL.extension.cs
- Trường hợp đặt tên trong thư mục StoredProcedure:
{Tên Store} + BL
Ví dụ: AP1409BL.cs

b) Trong Project khác

- Đặt tên các file theo màn hình đang xử lý
{Tên màn hình} + BL
Ví dụ: BF0110BL.cs

III) Các bước khai báo ngôn ngữ

1) Cấu trúc bảng lưu ngôn ngữ

A00000	
LanguageID	
LanguageName	
InsertDate	
UpdateDate	
Version	
Module	
Type	

A00001	
ID	
LanguageID	
Name	
InsertDate	
UpdateDate	
Module	
Deleted	
CustomName	
FormID	

A00002	
ID	
LanguageID	
Name	
InsertDate	
UpdateDate	
Module	
Deleted	

Tên bảng

Mô tả

A00000

Lưu version của ngôn ngữ theo module

LanguageID :

en-US : Tiếng Anh

ja-JP : Tiếng Nhật

vi-VN : Tiếng Việt

zh-CN : Tiếng Trung

A00001

Lưu ngôn ngữ của tất cả các phân hệ

A00002

Lưu message của tất cả phân hệ

Do db cũ chưa có cột FormID nên dùng script sau để thêm cột

```
ifnotexists(select*fromsyscolumns col innerjoinsysobjects tab
On col.id = tab.id where tab.name ='A00001'and col.name ='FormID')
BEGIN
    ALTERTABLE A00001 ADD FormID nvarchar(100)NULL
END
```

2) Tạo dữ liệu cho database

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

- ❖ Control chung : Button, Menu, khác, ...
- ❖ Thêm vào A0001.
- ❖ **Module** : 00, **FormID**: "A00", **ID** : Theo phân tích
- ❖ Sử dụng script sau để thêm khóa

```
IFNOTEXISTS(SELECT 1 FROM [A00001] WHERE [ID] ='btnSaveNew'and
LanguageID ='vi-VN')
INSERTINTO [A00001]([ID],[LanguageID],[Name],[Module], [FormID])
VALUES ('btnSaveNew','vi-VN','N'Lưu & nhập tiếp','00','A00')
```

1.2) 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
- ❖ Sử dụng script sau để thêm khóa

```
IFNOTEXISTS(SELECT 1 FROM [A00001] WHERE [ID]
='MTF1001.Disabled'and LanguageID ='vi-VN')
INSERTINTO [A00001]([ID],[LanguageID],[Name],[Module], [FormID])
VALUES ('MTF1001.Disabled','vi-VN','N'Disabled','RD','MTF1001')
```

1.3) Thêm message mới

- ❖ Thêm một khóa vào bảng A00002
- ❖ Sử dụng script sau để thêm khóa

```
IFNOTEXISTS(SELECT 1 FROM [A00002] WHERE [ID] ='AFML000001'and
LanguageID ='en-US')
INSERTINTO [A00002]([ID],[LanguageID],[Name],[Module], [Deleted])
VALUES ('AFML000001','en-US','N'Absolute value is so much','RD', 0)
```

1.4) 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;
```

1.5) Quy 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}

3) Khai báo trong project

1.1) 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);

1.2) Hiện thị ngôn ngữ cho control chung, menu (trong .cshtml)

- ❖ Lưu trong cache khi khởi đội IIS
- ❖ Chưa kiểm tra version khi thay đổi A0001 (TODO)
- ❖ Sử dụng trong code :

```
ASOFTLanguage.FindWordA00("btnSaveNew")
```

Ví dụ :

```
@Html.DisplayName(ASOFTLanguage.FindWordA00("A00.CreateUserID"))
```

1.3) Hiện thị ngôn ngữ khi chọn 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
- ❖ Khai báo Attribute LocalizedDisplayName cho properties
- ❖ Sử dụng trong code:

Khai báo trong ViewModel

```
[LocalizedDisplayName("MTF1001.DivisionID", "MTF1001")]
```

```
publicstring DivisionID { set; get; }
```

➔ **MTF1001.DivisionID** : Khóa (tương ứng là ID trong A00001)

➔ **MTF10001** : Tên màn hình (tương ứng là FormID trong A0001)

Khai báo trong .cshtml

```
@Html.LabelFor(model => model.DivisionID)
```

IV) Quy định chuẩn Encoding khi lưu file

Tất cả các file trong chương trình phải lưu với chuẩn Encoding là **Unicode (UTF-8 with signature) – Codepage 65001**

V) Hướng dẫn xử lý code

1) Xử lý SQL

1.1) SQL liên quan đến một Table

- Các file source code và SQL liên quan đến một table sẽ được sinh code tự động bằng tool, các file được sinh ra:

+ **TableName.cs**

+ **TableNameDAL.cs**

+ **TableNameBL.cs**

- Các method được tạo sẵn tương ứng:

+ **GetAll** : select tất cả dữ liệu của table

+ **GetByKey** : select dữ liệu theo Key của table

+ **Exist** : kiểm tra dữ liệu có tồn tại chưa

+ **Insert** : thực hiện Insert dữ liệu

+ **InsertRange** : thực hiện Insert danh sách dữ liệu

- + **Update** : thực hiện Update dữ liệu
- + **UpdateRange** : thực hiện Update danh sách dữ liệu
- + **Delete** : thực hiện Delete dữ liệu
- + **AddParameter** : thực hiện gán biến vào câu truy vấn

1.2) SQL liên quan đến một Store Procedure

- Các file xử lý liên quan đến Store sẽ được người phát triển định nghĩa:
 - + **StoreNameDAL.cs**
 - + **StoreNameBL.cs**
- Các method được tạo tương ứng:
 - + **Execute**
- Một số lưu ý đối với người phát triển:
 - + Do tool sinh code không thể kiểm tra được store dùng để lấy dữ liệu hay để thực thi SQL nên người phát triển phải xác định trường hợp xử lý của Store để implement method cho phù hợp và chính xác.
 - + Trường hợp Store dùng để thực thi SQL thì chỉ cần implement method **Execute** ở tầng DataAccess.

1.3) SQL kết hợp bởi nhiều Table

- Trường hợp này, sẽ không tự động sinh code mà người phát triển phải thực hiện sinh code bằng tay
- Các bước thực hiện:
 - + **B1:** Khai báo thêm câu SQL, chú ý cách đặt tên
 TD: Thêm vào một số câu SQL có liên quan đến table AT1004
 - + **B1': Thực hiện sinh file Entities** (nếu là câu select trên nhiều bảng)
 (Tham khảo tài liệu hướng dẫn generator)
 - + **B2: File **ScreenNameDAL.cs** (trong Project DataAccess)**
 Tạo method theo chức năng yêu cầu của người phát triển

TD:

```
public List<AV9999> BF0113_GetMonth()
{
    List<AV9999> result = newList<AV9999>();
    DbCommand command = null;

    try
    {
        // Xử lý câu sql
        string sql =
            ASOFTDatabase.GetSql(Sql.BF0113_FillMonthComboBox);
        // Tạo đối tượng sql command
    }
}
```

```
using (command = ASOFTDatabase.GetSqlCommand(sql))
{
    // Tạo reader
    using (var reader = ASOFTDatabase.ExecuteReader(command,
        this))
    {
        // Đọc dữ liệu
        result =
        ASOFTDatabase.ToList<AV9999>(reader);
    }
}
catch (Exception ex)
{
    throw ASOFTException.FromCommand(command, ex);
}

return result;
}
catch (Exception ex)
{
    throw new ASOFTException(ex.Message,
        LayerException.DataAccess);
}
}
```

+ **B3: File ScreenNameBL.cs**

Thực hiện override lại các method theo yêu cầu của người phát triển.

TD: BF0113_GetMonth

```
public List<AV9999> BF0113_GetMonth()
{
    try
    {
        var dal = new BF0113DAL();
        var result = dal.BF0113_GetMonth();
        return result;
    }
    catch (ASOFTException) { throw; }
    catch (Exception ex)
    {
        throw new ASOFTException(ASOFTLayer.Business, ex.Message, ex);
    }
}
```

1.4) Xử lý SQL cộng chuỗi trong code :

- Cách xử lý : Chuyển các câu SQL như thế này sẽ thực hiện như sau :

```
public DataTable GetDataByClauses(string whereclause, string orderby)
{
    DataTable result = new DataTable();
    DbCommand command = null;

    try
    {
        // Xử lý câu sql
        string sql =
        ASOFTDatabase.GetSql(ExtensionsSql.AV7501_GetDataByClauses);
        // Tạo đối tượng sql command
        using (command = ASOFTDatabase.GetSqlCommand(sql))
        {
            //Thêm tham số
            ASOFTDatabase.ReplaceParameter(command, "@Where", whereclause);
            ASOFTDatabase.ReplaceParameter(command, "@OrderClause",
            orderby);
            // Tạo reader
            using (var reader = ASOFTDatabase.ExecuteReader(command, this))
            {
                // Đọc dữ liệu
                result.Load(reader);
            }
        }
    }
    catch (Exception ex)
    {
        throw ASOFTException.FromCommand(command, ex);
    }

    return result;
}
```

1.5) Xử lý parameter trong SQL:

- Sẽ có 2 loại hàm được sử dụng để truyền vào parameter trong SQL.
- Một là ASOFTDatabase.AddInParameter(Dbcommand command,string pname,DbType ptype,object value)

Ví dụ:

```
public DataTable GetByGroupID(string groupid)
{
    DataTable result = new DataTable();

    DbCommand command = null;
```

```
try
{
    // Xử lý câu sql
    string sql = ASOFTDatabase.GetSql(Sql.BF0004_FillDivisionID);
    // Tạo đối tượng sql command
    using (command = ASOFTDatabase.GetSqlCommand(sql))
    {
        // Thêm tham số
        ASOFTDatabase.AddInParameter(command, "@GroupID",
        DbType.String, groupid);
        // Tạo reader
        using (var reader = ASOFTDatabase.ExecuteReader(command, this))
        {
            result.Load(reader);
        }
    }
}
catch (Exception ex)
{
    throw ASOFTException.FromCommand(command, ex);
}
return result;
}
```

- Hai là sử dụng `ASOFTDatabase.ReplaceParameter(Dbcommand command, string pname, object value)`

Ví dụ:


```
public List<BF0116ReportID> FillReportID(string fieldname, string
fieldtitle)
{
    var result = newList<BF0116ReportID>();

    DbCommand command = null;

    try
    {
        // Xử lý câu sql
        string sql = ASOFTDatabase.GetSql(Sql.BF0116_FillReportID);
        // Tạo đối tượng sql command
        using (command = ASOFTDatabase.GetSqlCommand(sql))
        {
            // Thêm tham số
            ASOFTDatabase.ReplaceParameter(command, "@ReportName",
            fieldname);
            ASOFTDatabase.ReplaceParameter(command, "@Title", fieldtitle);
            // Tạo reader
            using (var reader = ASOFTDatabase.ExecuteReader(command, this))
            {
                result =
                ASOFTDatabase.ToList<BF0116ReportID>(reader);
            }
        }
    }
    catch (Exception ex)
```

```

    {
        throw ASOFTException.FromCommand(command, ex);
    }

    return result;
}

```

Trong hàm **ReplaceParameter** sẽ được chia ra 3 trường hợp thêm vào parameter:

- Nếu là **danh sách chuỗi** thì chuỗi thay thế có dạng N'abc', N'def', ..., N'xyz'.
- Nếu **không là danh sách chuỗi** thì mặc định xem như là danh sách số khi đó chuỗi thay thế có dạng 9999, 43, 67, ..., 99.
- Nếu là **1 đối tượng** thì thay tên biến bằng chính giá trị truyền vào.

1.6) Một số trường hợp đặc biệt:

- Một số parameter lấy thông tin chung của hệ thống sẽ được thêm tự động khi gọi hàm **ASOFTDatabase.GetSqlCommand**. Khi code không cần truyền giá trị, chỉ cần đặt đúng tên parameter theo danh sách bên dưới. Danh sách tham số mặc định và giá trị:

Tên parameter	Giá trị truyền vào
@DivisionID	ASOFTEnvironment.DivisionID
@DefDivisionID	ASOFTEnvironment.DivisionID
@CreateUserID	ASOFTEnvironment.UserID
@LastModifyUserID	ASOFTEnvironment.UserID
@ConditionOB	ASOFTEnvironment.ConditionVT
@IsUsedConditionOB	ASOFTEnvironment.IsUsedConditionVT
@ConditionIV	ASOFTEnvironment.ConditionOB
@IsUsedConditionIV	ASOFTEnvironment.IsUsedConditionOB
@ConditionVT	ASOFTEnvironment.ConditionWA
@IsUsedConditionVT	ASOFTEnvironment.IsUsedConditionWA

@ConditionWA	ASOFTEnvironment.ConditionIV
@IsUsedConditionWA	ASOFTEnvironment.IsUsedConditionIV
@ConditionAC	ASOFTEnvironment.ConditionAC
@IsUsedConditionAC	ASOFTEnvironment.IsUsedConditionAC
@ConditionDE	ASOFTEnvironment.ConditionDE
@IsUsedConditionDE	ASOFTEnvironment.IsUsedConditionDE

Trong một số trường hợp đặc biệt không muốn 2 parameter @DivisionID và @DefDivisionID được truyền tự động thì dùng như bên dưới. Khi đó giá trị tham số truyền như bình thường. Trong trường hợp này được truyền thông qua hàm

`ASOFTDatabase.GetSqlCommand(sqlstring, false);`

1.7) Xử lý ngày trong SQL

- Hiện trạng:
 - + Xử lý sql liên quan đến ngày khi tìm kiếm hoặc thao tác bất cứ nghiệp vụ gì thì đều có đính kèm giờ/phút/giây.
 - + Vì thế khi tìm kiếm sẽ không ra dữ liệu nếu ngày có thêm giờ/phút/giây.

```
(CONVERT(VARCHAR(10), VoucherDate, 101) BETWEEN '' +  
CONVERT(VARCHAR(10), @FromDate, 101) + '' AND '' +  
CONVERT(VARCHAR(10), @ToDate, 101) + '')
```

- Cách xử lý:
 - + Trước tiên là LTRIM, RTRIM giá trị tham số ngày
 - + Định dạng giá trị tham số ngày về kiểu là 101. Tham khảo ví dụ

```
(CONVERT(DATETIME, CONVERT(varchar(10), ' + LTRIM(RTRIM(@TypeDate)) +  
, 101), 101) BETWEEN '' + CONVERT(VARCHAR(10), @FromDate, 101) +  
'' AND '' + CONVERT(VARCHAR(10), @ToDate, 101) + '')
```

2) Cách xử lý try/catch

2.1) Project Web

Không đặt try/catch ở bất cứ method hay event nào của Controller hoặc

View.

2.2) Project Business

Sử dụng đối tượng `ASOFTException` của hệ thống để throw error.

TD:

```
try  
{  
    //Do something
```

```
}
catch (ASOFTException) { throw; }
catch (Exception ex)
{
    throw new ASOFTException(ASOFTLayer.Business, ex.Message, ex);
}
```

2.3) Project DataAccess

Sử dụng đối tượng **ASOFTException** của hệ thống để throw error.

TD:

```
try
{
    //Do something
}
catch (Exception ex)
{
    throw ASOFTException.FromCommand(dbCommand, ex);
}
```

2.4) Project Entities

Không đặt try/catch ở bất cứ method xử lý nào.

3) Cách sử dụng biến hệ thống

- Dùng các biến hệ thống trong class **ASOFTEnvironment**.

.NET
ASOFTEnvironment.Month
ASOFTEnvironment.Year
ASOFTEnvironment.DivisionID
ASOFTEnvironment.UserID
ASOFTEnvironment.GroupID
...

4) Danh sách các hàm hỗ trợ layout và cách dùng:

4.1) Cách dùng

- Dùng trong view với cú pháp với **@Html.<tên hàm>**
- Chi tiết về tham số tham khảo bằng chức năng intellisense của Visual Studio.
- Đối với control nhập số, khi dùng **GridViewSpinEditColumn** và **ASOFTSpinEdit**, cần thêm tham số **dataType**. Tham số này dùng để xác định MaxValue, MinValue cho phép nhập trên control để tránh lỗi tràn số. Các giá trị có thể truyền vào của tham số và kiểu dữ liệu tương ứng trong DB:

- **typeof(decimal)** tương ứng với kiểu **decimal** (mặc định)
- **typeof(int)** tương ứng với kiểu **int**
- **typeof(byte)** tương ứng với kiểu **tinyint**

- Xử lý hiển thị dòng [% - Tất cả] cho combo InventoryTypeID:
Khi cần hiển thị dòng [% - Tất cả] chỉ cần set ShowRowAll = true, không cần sửa câu sql.

4.2) Hỗ trợ tạo các control thông thường:

Tên	Hình	Cách sử dụng	Dạng Control	Tham khảo
ASOFTRaw		Không encode HTML Html.ASOFTPageTitle(m => m.MTF1000Title, "MTF1000");	MVC	
ASOFTPageTitle		Hiển thị tiêu đề của trang Html.ASOFTRaw(m => m.MTF1000Title)	MVC	
ASOFTButton		*.cshtml	Kendo	Link
ASOFTButtonIcon		@(Html.ASOFTSearchButton()	Kendo	
ASOFTSearchButton		.Name("btnFilterMaster")	Kendo	
ASOFTSearchResetButton		.Events(ev =>	Kendo	
ASOFTAddButton		ev.Click("FilterData"))	Kendo	
ASOFTEditButton		*.js	Kendo	
ASOFTDeleteButton		<script>	Kendo	
ASOFTDeleteDetailButton		function FilterData(event) { //TODO : event	Kendo	
ASOFTPrinterButton		}	Kendo	
ASOFTExportExcelButton		</script>	Kendo	
ASOFTExportExcelDetailButton			Kendo	
ASOFTTextBox		@Html.ASOFTTextBox(model => model.CourseIDFilter)	MVC4	
ASOFTCheckbox		@Html.ASOFTCheckbox(model => model.IsCommonFilter)	MVC4	
ASOFTRadioButton		@Html.ASOFTRadioButton(model => model.IsCommonFilter, "1")	MVC4	

		@Html.ASOFTRadioButton(mode l => model.IsCommonFilter,"2")		
ASOFTRadioButtontList		@Html. ASOFTRadioButtontList (model => model.IsCommonFilter,new List<SelectListItem> () {})		
ASOFTLabel		@Html.ASOFTLabel(model => model.DivisionIDFilter) Nếu có required thì đăng ký Attribute trong *.ViewModel	MVC4	
ASOFTPassWord		@Html.ASOFTPassWord(model => model.DivisionIDFilter)	MVC4	
ASOFTPanel		Html.AsoftPanel("").ToHtmlStrin g()	Kendo	Link
ASOFTHidden		@Html.ASOFTHidden("currentCo ntroller",Html.ASOFTGetControll er())	MVC4	
ASOFTCombobox		var comboBox = (Html.ASOFTCombobox(model => model.CurrencyID, htmlStyle: Model.Style) .Name(Model.Name) .DataValueField(AT1004.COL_ CURRENCYID) .DataTextField(AT1004.COL_C URRENCYNAME) .Enable(Model.Enabled));	Kendo	Link
ASOFTPopup		@(Html.ASOFTPopup() .Name("MTF1001Popup") .Width(500) .Title(string.Format("{0} - MTF1001", Html.DisplayNameFor(m=>m.M	Kendo	Link

		TF1001_Title).ToHtmlString()))		
ASOFTSpinEdit		@Html.ASOFTSpinEdit(model => model.Quantity)	Kendo	Link
ASOFTDateEdit		@Html. ASOFTDateEdit (model => model.Quantity)	Kendo	Link
ASOFTTimeEdit		Html. ASOFTTimeEdit (model => model.Quantity)	Kendo	Link
ASOFTTextArea		@Html.ASOFTTextArea(model= >model.Notes)	MVC4	
Các hàm hỗ trợ cho MVCshtmlString				
DisabledIf (Thêm attribute disable = "disabled" vào thuộc tính của control)		MVC : @Html.ASOFTTextArea(model= >model.Notes).DisableIf() KenDo: @Html. ASOFTDateEdit (model => model.Quantity). ToClientTemplate().DisableIf () Chú ý: Hàm này đặt cuối cùng của phần khai báo	Mvcsh tmlStri ng	
ReadOnlyIf (Thêm attribute readonly vào thuộc tính của control)		MVC : @Html.ASOFTTextArea(model= >model.Notes). ReadOnlyIf() KenDo: @Html. ASOFTDateEdit (model	Mvcsh tmlStri ng	

		=> model.Quantity) .ToClientTemplate() .ReadOnlyIf() Chú ý: Hàm này đặt cuối cùng của phần khai báo		
HideIf (Không hiển thị control)		MVC : @Html.LabelFor(model => model.IsCommon). HideIf (true) KenDo: @Html. ASOFTDateEdit (model => model.Quantity). HideIf ()		
CheckedIf (Check radio/ checkbox)		MVC : @Html.LabelFor(model => model.IsCommon). CheckIf (true)		

4.3) Hỗ trợ các control đặc biệt:

Tạo control nhập dữ liệu số, các control này đã được gán sự kiện tự động format trong quá trình nhập liệu. Đồng thời, giá trị sẽ được làm tròn theo thiết lập trong ERP 8

- **ASOFTConvertedDecimal:** Hiển thị textbox nhập dữ liệu số **tiền quy đổi**, sau khi tính toán, ví dụ: thành tiền, tiền chiết khấu, tiền giảm giá, tiền thuế...
- **ASOFTPercentDecimal:** Hiển thị textbox nhập dữ liệu **phần trăm**, ví dụ: phần trăm thuế, phần trăm chiết khấu.
- **ASOFTQuantityDecimal:** Hiển thị textbox nhập dữ liệu **số lượng** sản phẩm, có thể nhập số nguyên, hoặc số thập phân tùy vào thiết lập trong ERP.
- **ASOFTUnitCostDecimal:** Hiển thị textbox chuyên dùng cho nhập dữ liệu **đơn giá**.

VD: @Html.ASOFTPercentDecimal(model => model.TotalDiscountRate)

Tạo control đồng thời lấy dữ liệu cho control.

- **ASOFTDivisionID** : Đơn vị
- **ASOFTPeriodID** : Kỳ kế toán
- **ASOFTAccountID** : Tài khoản
- **ASOFTCurrencyID** : Loại tiền

- **ASOFTEmployeeID** : Nhân viên
- **ASOFTObjectTypeID** : Loại đối tượng
- **ASOFTObjectID** : Đối tượng
- **ASOFTInventoryTypeID** : Loại hàng
- **ASOFTInventoryID** : Mặt hàng
- **ASOFTWareHouseID** : Kho
- **ASOFTUnitID** : Đơn vị tính
- **ASOFTVoucherTypeID** : Loại chứng từ
- **ASOFTChart** : Biểu đồ
- **ASOFTPaymentTermID** : Điều kiện thanh toán
- **ASOFTPaymentID** : Phương thức thanh toán
- **ASOFTPortID** : Danh sách cảng
- **ASOFTDepartmentID** : Danh sách phòng ban
- **ASOFTTeamIDByDepartmentID** : Danh sách tổ nhóm theo phòng ban
- **ASOFTWorkID** : Danh sách công việc
- **ASOFTVATGroupID** : Danh sách nhóm thuế
- **ASOFTServiceTypeID** : Loại dịch vụ
- **ASOFTStatusID** : Tình trạng
- **ASOFTPlanID** : Danh mục nhóm phương án
- **ASOFTSubPlanID** : Danh mục phương án
- **ASOFTShiftID** : Danh mục công việc

4.4) ControlGridView

❖ Các thuộc tính trong class **ASOFTGridColumnSetting**

Tên thuộc tính	Diễn giải	Ghi chú
IDColumn	Dùng để map từ dữ liệu vào cột	Yêu cầu
TitleColumn	Caption của cột	
Format	Định dạng dữ liệu của cột	
Width	Chiều rộng của mỗi cột	
ClientTemplate	Custom template: VD: <a style='text-decoration:underline;cursor:pointer'	

	onclick='ClickLinkOnGrid()'>#:FieldID#	
VisibleColumn	Thuộc tính hiển thị cột.	Mặc định là true
EditorTemplateName	Định nghĩa TemplateName khi sửa Cell. Template đặt trong thư mục Shared/EditorTemplates	Chỉ sử dụng cho lưới Edit
EditColumn	Định nghĩa cho phép cột có được phép sửa.	Mặc định là true(được phép sửa).Chỉ sử dụng cho lưới Edit
FormatType	Kiểu dữ liệu EnumFormatType , dùng để xác định format số và editor template cho cột. EnumFormatType.QuantityDecimals: để định dạng cột số lượng EnumFormatType.UnitCostDecimals: để định dạng cột đơn giá EnumFormatType.PercentDecimals: để định dạng cột phần trăm EnumFormatType.ConvertedDecimals: để định dạng cột tiền quy đổi, tiền sau khi tính toán	Mặc định nếu không xác định FormatType , thì cột sẽ được định dạng theo kiểu dữ liệu của Property trong ViewModel , editor template mặc định là textbox thông thường

❖ Các thuộc tính trong class **ASOFTGridSetting**

Tên thuộc tính	Diễn giải	Ghi chú
NameGrid	ID Của Lưới.	Yêu cầu
Scrollable	Cho phép cuộn trên lưới.	Mặc định là false

HtmlAttribute	Định nghĩa các thuộc tính html cho lưới. VD: <code>HtmlAttribute = new {style = "height:590px;", @class = "asf-grid"}</code>	
ListGridColumn	Định nghĩa danh sách coloumns cho lưới	
ControllerName	Tên Controller để lấy dữ liệu về lưới	Yêu cầu
ActionName	Tên ActionName để lấy dữ liệu về lưới	Yêu Cầu
NameEventPost	Tên hàm trong javascript. Thêm tham số khi Post lên ControllerName	
NameEventDataBound	Tên hàm trong javascript sau khi dữ liệu đã được bind vào lưới.	Không dùng cho lưới Edit
NameEventDataBind	Tên hàm trong javascript khi dữ liệu trước khi bind vào lưới	

❖ JavaScript

Tham khảo thêm Kendo Grid.

Link: <http://docs.kendoui.com/api/web/grid>

Biến và hàm có sẵn trong javascript:

Biến **currentRecordGrid**: Trả về dòng được chọn trên lưới.

Hàm **ASOFT.asoftGrid. ReturnGrid(name)**: trả về đối tượng lưới với tham số ID của lưới.

Lấy **dataSource** của lưới: **grid.dataSource**.

Load lại lưới dùng hàm **read()** của đối tượng **dataSource**.

❖ Gridview có 3 loại:

- Loại phân trang: dùng hàm **ASOFTCreateGrid**

TD: @Html.ASOFTCreateGrid<object>(ASOFTGridSetting)

- Loại không phân trang dùng hàm:
ASOFTCreateGridNotPaging
TD:@Html.ASOFTCreateGridNotPaging<object>
(ASOFTGridSetting)
- Loại cho phép Edit trên lưới: dùng
hàm:**ASOFTCreateGridEdit**
TD:@Html.ASOFTCreateGridEdit <object>
(ASOFTGridSetting)

4.5) Menu

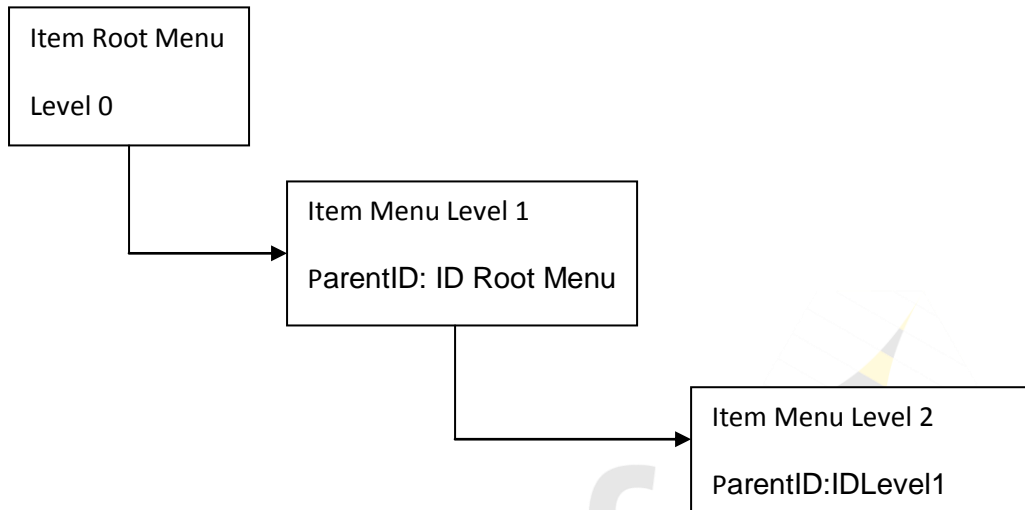
Chú ý: Menu được sinh tự động từ Controller.

- Các thuộc tính trong class **ASOFTMenuItem**

Tên thuộc tính	Diễn giải	Ghi chú
Name	Id của menu. Chú ý: đối với menu level 1 và level 2 cần phải tuân thủ đặt theo quy tắc sau: IDLevel0_IDLevel1_IDLevel2	Yêu cầu
Level	Cấp độ item của menu	Yêu cầu
ParentID	Id cha của menu.	Root menu thì không yêu cầu
Text	Text của item menu	
Url	Link của item menu	
Controller	Tên Controller của item menu	Item Menu Root không cần phải nhập. Yêu cầu nhập đối với menu cấp 1 và menu cấp 2

Action	Tên Action của item menu	Item Menu Root không cần phải nhập. Yêu cầu nhập đối với menu cấp 1 và menu cấp 2
--------	--------------------------	---

Cấu trúc cây menu:



TD:Add Menu

Menu Level0:

```

asoftMenu.Add(new ASOFTMenuItem
{
    Level = 0,
    Name = "ItemQLHV",
    Text = "QLHV",
    Url = "",
    Controller = "",
    Action = ""
});
  
```

Menu Level1:

```

asoftMenu.Add(new ASOFTMenuItem
{
    Level = 1,
    ParentID = "ItemQLHV",
    Name = "ItemQLHV_DanhMuc",
    Text = "Hệ thống",
    Url = "/MT/MTF0010",
    Controller = "MTF0010",
    Action = "Index"
});
  
```

Menu Level 2:

```
asoftMenu.Add(new ASOFTMenuItem  
{  
    Level = 2,  
    ParentID = "ItemQLHV_DanhMuc",  
    Name = "ItemQLHV_DanhMuc_KhoaHoc",  
    Text = "Khóa học",  
    Controller = "MTF1000",  
    Action = "Index",  
    Url = "/MT/MTF1000/"  
});
```

asoft
Growing Together

4.6) MultiSelectBoxItem - Control chọn nhiều Item

a. Mục đích:

- Cho phép chọn 1 một hoặc nhiều item dung để Lọc, lưu nhiều dữ liệu
- Kế thừa `DropDownListBuilder` của `KenDo`

b. Cách sử dụng:

```
@Html.ASOFTMultiSelectBox(
    mode=>mode.DivisionID,
    newList<MultiSelectBoxItem>()
)
```

- `MultiSelectBoxItem` : là một đối tượng gồm 3 thuộc tính:
 - **Text** : tên của checkbox
 - **Value**: giá trị của checkbox
 - **Checked** : Checkbox có được chọn hay không?
- `MultiSelectBoxItem` kế thừa từ `DropDownListItem`, mặc dù có thuộc tính là `Selected` nhưng sẽ không được sử dụng (Lý do : Không gen ra trên html nên không xử lý được phần check Tất cả)
- Cách sử dụng như các control khác mô tả tại **4.2 Hỗ trợ tạo các control thông thường.**
- Không được định nghĩa lại các sự kiện, phương thức:
 - `Events(m => m.DataBound())`
 - `Events(m => m.Select())`
 - `Template()`
- Các text "(Tất cả), {0} Chọn" được lưu trong database

c. Format sau khi chọn các item trong javascript

- Sau khi chọn các item thì value của control được lấy như sau:

```
$( '#DivisionID' ).val()  
// 0,1,2,3,4,5,6
```

- Lên server xử lý chuỗi này để sử dụng trong nghiệp vụ

d. Sử dụng dữ liệu khác loại List< MultiSelectBoxItem>

- Trong vài trường hợp, nếu không sử dụng MultiSelectBoxItem là kiểu của một Item thì có thể xử lý như sau:

```
@Html.ASOFTMultiSelectBox(mode=>mode.DivisionID)  
    .DataValueField(AT1101.COL_DIVISIONID)  
    .DataTextField(AT1101.COL_DIVISIONNAME)
```

- Thêm thuộc tính vào kiểu dữ liệu item của data

```
public class AT1101  
{  
    .....  
    public bool Checked { set; get; }  
}
```

asoft
Growing Together

4) Các hàm hỗ trợ javascript

Namespace	Hàm	return	Nội dung
ASOFT.form	getMaxHeightByClass(className)	Int	
	setAutoHeight (heightOfClass, horzClass)		
	setSameHeight (className)		
	setSameWidth (className)		
	showDiv (ctrl, display)		
	setValidator (value)		
	checkRequired (formId)		
	createMessageBox (message, skipItems)		
	displayMessageBox (elementID,message,skipMessageItems)		
	selectedTab (tabStripId, tabPageSelectedId)		
	enterReplaceTab (event)		
	displayWarning (elementID, message)		ASOFT.form.displayWarning("#FormFilter", "Test info");
	displayInfo (elementID, message)		ASOFT.form.displayInfo("#FormFilter", "Info info");

ASOFT.dialog	oKClick ()		
	yesClick ()		
	noClick ()		
	templateMessageDialog ((message, name)		
	createDialog (name)		
	messageDialog (message)		
	confirmDialog (message, yesFunction, noFunction)		
ASOFT.asofLoadingPanel	show()		
	hide()		
	createPanel()		
ASOFT.asoftPopup	create(name, title, modal, visible, draggable)		
	castName(name)		
	show(popup, url, data)		
	refresh(popup, url, data)		
	center(popup)		
	hide(popup)		
	activate(popup, handler)		

	hidePopupAction(e)		
ASOFT.comboBox	castName(name)		
	setDataSource(url, data)		
	callback (name, url, data)		
ASOFT.asoftGrid	pageSizeRight()		
	returnGrid(name)		
	checkAll()		
	rertunListDelete(name)		
	recordSelect()		
	dataBindGridPanging()		
	setValueTextbox(name, e, cellindex, rowindex)		
	checkEditCell(model, fieldId)		
	openCellEdit(model, fieldId)		
	closeCellEdit (model, fielded)		
	addRecord (name)		
	removeRecord (name)		
	movePrevCell(elm, name)		

	moveNextCell(elm, name)		
	processKeyForGrid(e, elm, name)		
	dataBoundGridEdit()		
	removeEventKeyTab(gridobject)		
ASOFT.helper	getMessage(id)		
	getLabelText(id, messageId)		
	debug(data)		
	getFormData(includeListName, formID)		
	post(url, data, success, error)		
	defaultErrorHandler()		
ASOFT.format	stringToDate (value)		
	dateToString (value, format)		
	jsonToDateInList(list)		
	jsonToDate(value)		
	trim(value, charater)		
	numberToString (value, decimal)		

5) Sử dụng Toolbar

5.1) File *.CSHTML

```
@Html.Partial("~/Views/Shared/EditorTemplates/Toolbar.cshtml",
@Model.Toolbar1)
```

5.2) *ViewModel.cs

```
public MTF1000ViewModel()
{
    Toolbar1 = new ASOFTToolBarViewModel
    {
        DisplayAdd = true, //Hiển thị
        EventAdd = "", //Sự kiện
        DisplayDelete = true, // Hiển thị
        DisplayExport = true,
        DisplayPrint = true
    };
    Toolbar2 = new ASOFTToolBarViewModel
    {
        DisplayExport = true,
        DisplayPrint = true
    };
}
```

5.3) Thêm button cho toolbar

❖ Trong file ASOFTToolBarViewModel.cs

```
public bool DisplayAdd { get; set; } // Thêm thuộc tính
public string EventAdd { get; set; } //Thêm tên sự kiện
public string NameAdd { get; set; } //Id của button
```

❖ Trong file ASOFT.ERP/ Views/ Shared/ EditorTemplates/ Toolbar.cshtml

```
@if (Model.DisplayAdd)
{
    <li>@Html.ASOFTExportExcelDetailButton(Model.EventAdd)
        .Name(Model.NameAdd?? "btnExportDetail")
    </li>
}
```

6) Cách sử dụng AutoMapper

AutoMapper là một thư viện hỗ trợ chuyển dữ liệu qua lại giữa ViewModel và Entity. Thông thường chúng ta thường phải khai báo qua lại giữa các thuộc tính của ViewModel và Entity.

Ví dụ:

- mtt1000.CreateDate = MTF1000ViewModel.CreateDate;
- mtt1000.UserID = MTF1000ViewModel.UserID;
- ⇒ Khi sử dụng AutoMapper thì cách code như sau:

```
//Khai báo profile
CreateMap<MTF1000ViewModel, MTT1000>();
//Sử dụng
var mtt1000=ASOFTMapper.ToViewModel<MTF1000ViewModel,MTT1000>(args);
hoặc
ASOFTMapper.ToEntity< MTT1000, MTF1000ViewModel>(args);
```

Tham khảo thêm tại : <https://github.com/AutoMapper/AutoMapper/wiki>

Một vài qui định khi dùng AutoMapper:

- Sử dụng automapper trong class ***BL.cs (Business)**. Trừ trường hợp Automapper dành cho lưới (Tham khảo AsoftController.cs)
- Trong controller chỉ sử dụng các class *ViewModel.cs, không sử dụng Entity.
- Sử dụng hàm chung

6.1) Chuyển từ ViewModel sang Entity

- **ASOFTMapper.ToEntity**<MTF1000ViewModel, MTT1000>(args);
 - **Nguồn** : Class *.ViewModel
 - **Đích** : Class Entity (đại diện 1 bảng)

6.2) Chuyển từ Entity sang ViewModel

- **ASOFTMapper.ToViewModel**<MTF1000ViewModel, MTT1000>(args);
 - **Nguồn** : Class Entity (Đại diện cho một bảng)
 - **Đích** : Class *.ViewModel

7) Cách sử dụng Validate

- ❖ Sử dụng Kendo.validator để check validate của form
- ❖ Message các check chung sẽ khai báo trong script.js - tham khảo hàm createValidator
- ❖ Sử dụng rules và message để check
- ❖ Sử dụng Data Annotation của ViewMode và Javascript để check

7.1) Code ViewModel

Data Annotation	Ví dụ
Required	[Required] → để trống ErrorMessage
DataType	[DataType(DataType.DateTime)]
StringLength	[StringLength(12, ErrorMessage = "FFML000103")]

RegularExpression	[RegularExpression(ViewModelConstant.REX_EMAIL, ErrorMessage = "HFML000014")]
Range	[Range(0.01, 100.00, ErrorMessage = "HFML000015")]

7.2) Code javascript

```

rules: { // custom rules
    mvcdater: function (input) {
var dp = input.data("kendoDatePicker") ||
input.data("kendoDateTimePicker");
if (dp != undefined) {
value = input.val();
var result = false;
try {
var t = kendo.parseDate(value, 'dd/MM/yyyy', 'vi-VN');
if (t != null) {
return true;
}
result = false;
} catch (err) {
result = false;
}
}
return result;
}

return true;
},
mvcregex: function (input) {
//check for the rule attribute
if (input.filter("[data-val-regex]").length && input.val()) {
var regex = new RegExp(input.attr("data-val-regex-pattern"));
return regex.test(input.val());
}
return true;
}
},
messages: {
mvcrequired: function (input) {
var id = $(input).attr('id');
return ASOFT.helper.getLabelText(id, "ASML000082");
}
}
}

```

```

        mvdate: function (input) {
var id = $(input).attr('id');
returnASOFT.helper.getLabelText(id, "AFML000077");
        },
        mvregex: function (input) {
var id = $(input).attr('id');
var messageID = input.attr("data-val-regex");
returnASOFT.helper.getLabelText(id, messageID);
        },
        mvlength: function (input) {
        },
        mvcmx: function (input) {
        },
        mvcmn: function (input) {
        },
        mvnumber: function (input) {
        },
        mvcrange: function (input) {
        }
    }

```

8) Thiết lập cột ẩn trong View

- Cách thiết lập cũ:

```

settings.Columns.Add(
    Html.GridviewSpinEditColumn(newSpinEditColumnOptions(
        settings, PSP1020.COL_ORDERS, Language.STT, width: 40, readOnly:
true)
    {
        HorizontalAlign = System.Web.UI.WebControls.HorizontalAlign.Center,
        HiddenFieldNames = newstring[] {
PSP1020.COL_APK,
PSP1020.COL_INVENTORYTYPEID,
PSP1020.COL_INVENTORYID,
PSP1020.COL_PLANID,
PSP1020.COL_WORKID,
PSP1020.COL_SUBPLANID,
PSP1020.COL_SUBPLANNAME,
PSP1020.COL_UNITID,
        },
    }));

```

- Cách thiết lập mới: chú ý tránh nhầm lẫn giữa **HiddenFieldNames** và **FieldNames**


```
settings.Columns.Add(
    Html.GridViewSpinEditColumn(newSpinEditColumnOptions(
        settings, PSP1020.COL_ORDERS, Language.STT, width: 40, readOnly:
true)
    {
        HorizontalAlign = System.Web.UI.WebControls.HorizontalAlign.Center,
    }));

settings.Columns.Add(
    Html.GridViewHiddenColumn(newHiddenColumnOptions(settings)
    {
        FieldNames = newstring[] {
PSP1020.COL_APK,
PSP1020.COL_INVENTORYTYPEID,
PSP1020.COL_INVENTORYID,
PSP1020.COL_PLANID,
PSP1020.COL_WORKID,
PSP1020.COL_SUBPLANID,
PSP1020.COL_SUBPLANNNAME,
PSP1020.COL_UNITID,
        },
    }));
```

9) Quản lý session

Các biến Session: sẽ được quản lý thông qua AsoftEnvironment bằng các property và method. Đảm bảo khi truy xuất sẽ trả về đúng giá trị tương ứng với thông tin đăng nhập (UserID, GroupID, DivisionID...).

VI) Project Tool gen code

1) Mục đích

- ❖ Hỗ trợ người phát triển: giảm thời gian, giảm sai sót, dễ chỉnh sửa hệ thống trên diện rộng.

2) Mô tả xử lý

a. Thiết lập

The screenshot shows the 'MainForm' window with two tabs: 'Database information' and 'Solution information'. The 'Database information' tab is active, showing fields for Server (192.168.0.205\SQL2008R2), Database (CSG_NB), User (asofterp), and Password (asoft123456). The 'Solution information' tab shows a Path field with the value 'D:\ASOFTERP_NET\30_SOURCECODE\ASOFT-BI-MVC\ASoft.ERP\'. A red box highlights the database fields, and a yellow box highlights the path field. A red box also highlights the 'Form' tab, and a yellow box highlights the 'Table' tab.

Thông tin database

Đường dẫn đến source code

- ❖ Thiết lập thông tin database và đường dẫn tới thư mục chứa source code.
- ❖ Nếu không cần user/pass để đăng nhập thì bỏ check [User]

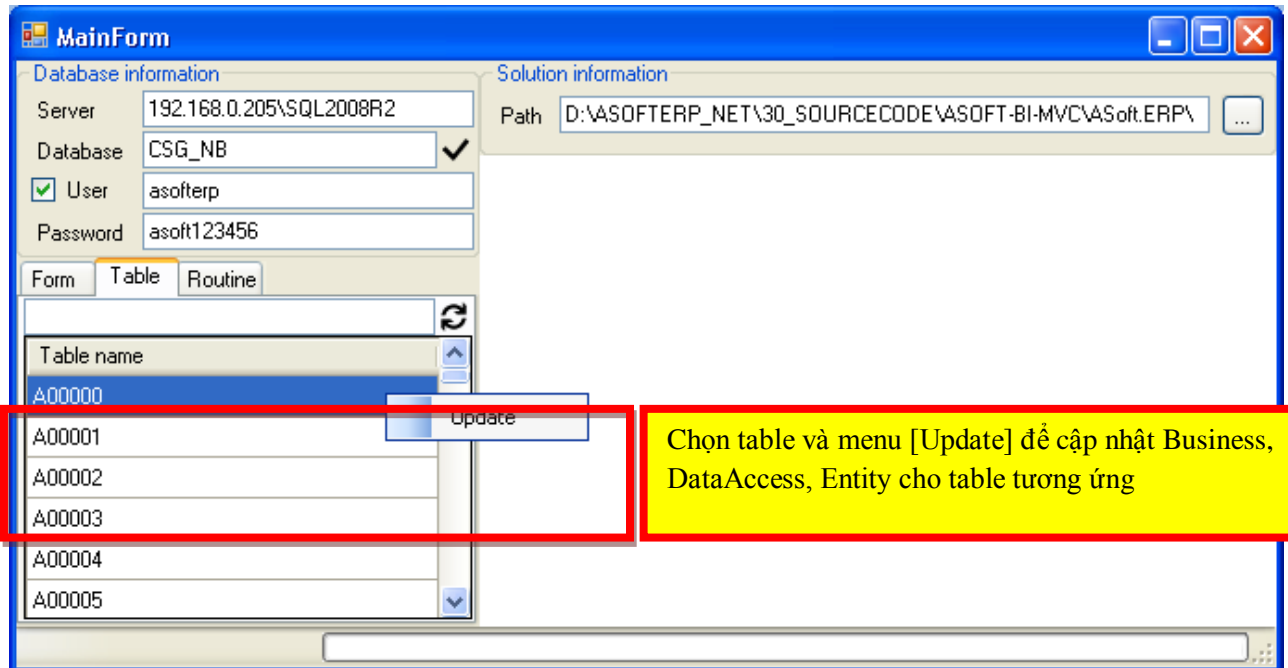
b. Generate code cho stored procedure

The screenshot shows the 'MainForm' window with the 'Routine' tab selected. The 'Routine name' list shows several routines, including AP0000, AP0001, AP0002, AP0003, AP0004, and AP0005. A red box highlights the 'Routine name' list, and a yellow box highlights the 'Update (GetData)', 'Update (GetFirst)', and 'Update (Execute)' options. A red box also highlights the 'Form' tab, and a yellow box highlights the 'Table' tab.

Chọn procedure và menu [Update] tương ứng để cập nhật Business, DataAccess, Entity cho procedure

- ❖ GetData: trả về danh sách List<T>, tương ứng với câu SELECT
- ❖ GetFirst: trả về một đối tượng T, tương ứng với câu SELECT TOP 1
- ❖ Execute: chỉ xử lý không trả về giá trị.

c. Generate code cho table



- ❖ Các hàm sẽ được tạo
 - **GetAll:** lấy toàn bộ dữ liệu của table.
 - **GetAllEnabled:** lấy toàn bộ dữ liệu của table với điều kiện Disable = 0.
 - **GetByKey:** lấy một record theo khoá.
 - **Exist:** kiểm tra dữ liệu với khoá tương ứng có tồn tại hay không.
 - **Insert:** thêm mới một record.
 - **InsertRange:** thêm mới nhiều record.
 - **Update:** cập nhật một record.
 - **UpdateRange:** cập nhật nhiều record.
 - **Delete:** xoá dữ liệu.
 - **DeleteLogic:** cập nhật DeleteFlag = 1.

d. Generate code cho màn hình

- ❖ Các thuộc tính của câu truy vấn:
 - **EntitySql:** câu truy vấn dùng để tạo đối tượng chứa dữ liệu. Dùng khi GenerateEntity = True.
 - **GenerateEntity:** có tạo đối tượng chứa dữ liệu hay không.
 - **Name:** tên câu truy vấn cũng là tên hàm sẽ được generate.
 - **ReturnEntity:** đối tượng chứa dữ liệu.
 - **ReturnList:** giá trị trả về là một đối tượng hay một List<T>.
 - **Sql:** câu truy vấn.
 - **SqlType:** kiểu của câu truy vấn: SELECT, UPDATE, DELETE, EXECUTE
- ❖ Các thuộc tính của tham số:

- DBType: kiểu dữ liệu database của tham số.
- Hidden: True thì không tạo tham số ở khai báo hàm, người phát triển phải cập nhật sau.
- IsReplace: dùng hàm [ReplaceParameter] hay [AddInParameter] để truyền tham số.
- Name: tên tham số.
- Type: kiểu dữ liệu .NET của tham số.

MainForm

Database information

Server: 192.168.0.205\SQL2008R2

Database: CSG_ND

☒ User: asofterp

Password: asoft123456

Form | Table | Routine

FormID: RDF0000

ModuleID: RD

Query name

- GetData_001
- GetData_002
- Delete_003
- GetData_004

EntitySql

GenerateEntity: False

Name: GetData_001

ReturnEntity: PST2010

ReturnList: True

Sql: SELECT Serial, Invoice

SqlType: Select

Parameter name

- DivisionID
- VoucherID

DBType: DbType.String

Hidden: True

IsReplace: False

Name: DivisionID

Type: string

Thông tin màn hình

Danh sách các câu truy vấn của màn hình

Danh sách các tham số của câu truy vấn