

NORTHWIND APP, GSB CE



1. Contexte de la situation professionnel	p3
2. Environnement de développement	p4
3. Modélisation fonctionnelle	p5/6
4. IHM	p7/8
5. Architecture applicative en couches	p9
6. Persistance des données	p10
7. Entity Framework	p11/13
8. Le DataBinding	p14/15
9. Classe partiel	p16
10. Test unitaires	p17
11. Déploiement	p18/19
12. Problème, proposition, conclusion	p20

Contexte de la situation professionnel

Le comité d'entreprise de GSB permet à l'ensemble des collaborateurs du groupe de bénéficier de conditions attractives lors de l'achat de produits d'épicerie fine auprès de la Société Northwind.

A cet effet, il nous faut développer une application pour gérer les commandes des collaborateurs du groupe GSB

Nous démarrerons le projet avec les ressources de la base de données ainsi qu'un Mock up de l'application désiré.



Environnement de développement



Microsoft Visual Studio outil de développement pour Windows et mac conçue par Microsoft. Il permet de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles.

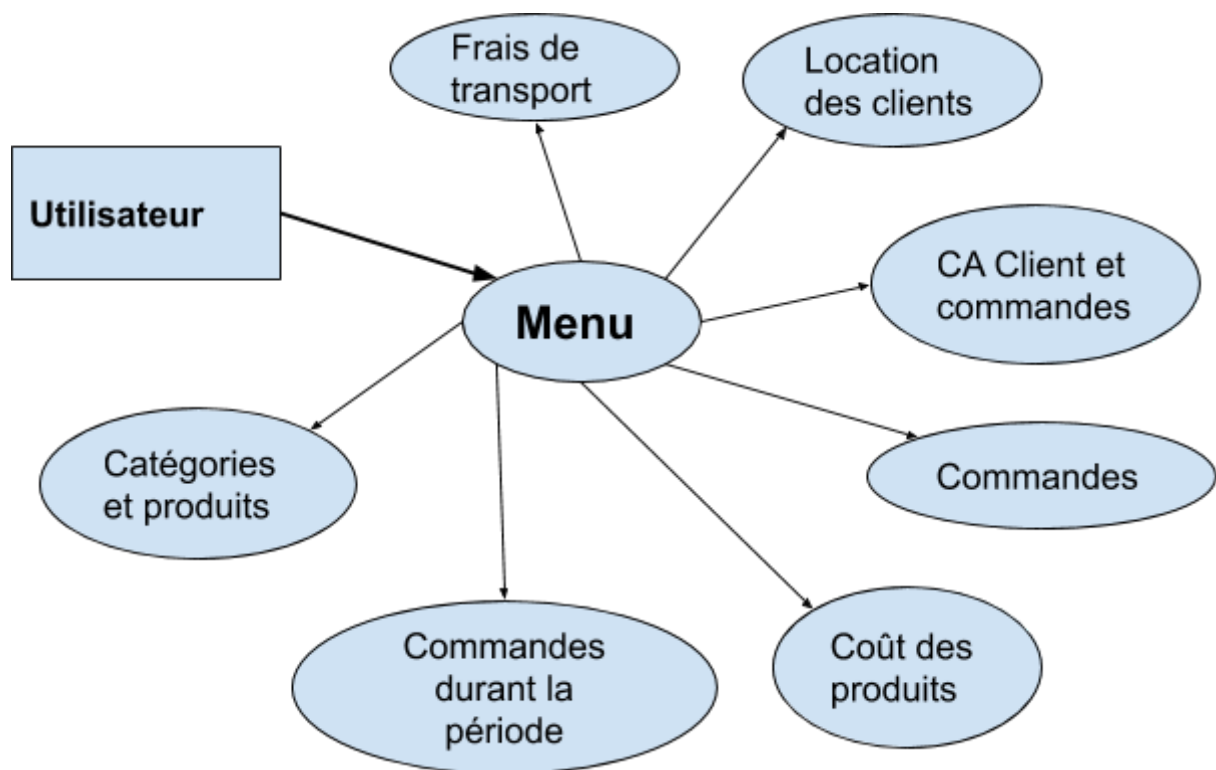


nous utiliserons Github comme outil de travail collaboratif.

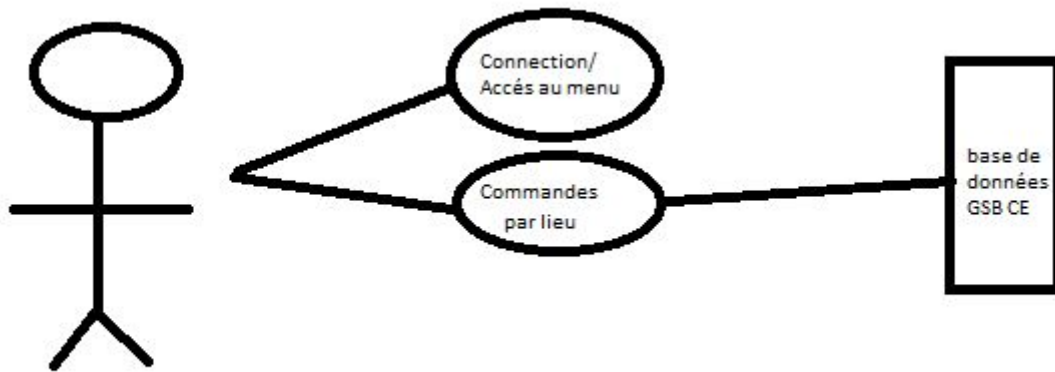
GitHub est un service web d'hébergement, de gestion de développement de logiciels. Ce site assure un contrôle d'accès et de fonctionnalités destinées à la collaboration.

Modélisation fonctionnelle

Lorsqu'un utilisateur lance l'application il arrive immédiatement au menu. À partir du menu il peut accéder à plusieurs fenêtres permettant de visionner ou modifier les données des commandes, des produits ou des clients.



Par exemple voici ce que doit faire l'utilisateur pour voir les commandes provenant d'une zone géographique précise.



Projet	GSB CE
pré condition	utilisateur possède l'application
utilisation	utilisateur va voir les commandes effectués dans un pays et une ville donnée.
Scénario nominal	1-utilisateur lance l'applicaion 2-System retourne le menu 3-utilisateur clique sur "Commandes géographique" 4- Système ferme la fenêtre du menu puis ouvre la fenetre des commandes géographiques 5- Système affiche les commandes, sans trier. 6- utilisateurs selectionne un pay 7-Systeme met à jour le dropdown ville et vide la liste des commandes. 8- Utilisateurs selectionne une ville 9- Système affiche les commandes dans la ville selectionné.
Exception	9-Système n'affiche aucune donnée car aucune commande n'à eu lieu dans cette ville

IHM

IHM définit les moyens et outils mis en œuvre afin qu'un utilisateur puisse contrôler et communiquer avec une machine. Les ingénieurs en ce domaine étudient la façon dont les humains interagissent avec les ordinateurs ou entre eux à l'aide d'ordinateur. Ils étudient aussi la façon de concevoir des systèmes ergonomiques et faciles à utiliser.



Dans notre situation nous communiquons avec l'application en utilisant des champs de caractères ou des dropdown, et elle nous réponds en affichant les données par le biais du datagridview.

Pays	<input type="text" value="Brazil"/>
Ville	<div><div>Rio de Janeiro</div><div><div>Campinas</div><div>Resende</div><div>Rio de Janeiro</div><div>Sao Paulo</div></div></div>

	CustomerID	CompanyName	Sale	ContactName
▶	HANAR	Hanari Cames	32841,370000	Mario Pontes
	QUEDE	Que Delícia	6664,810000	Bernardo Batista
	RICAR	Ricardo Adocica...	12450,800000	Janete Limeira

Le menu



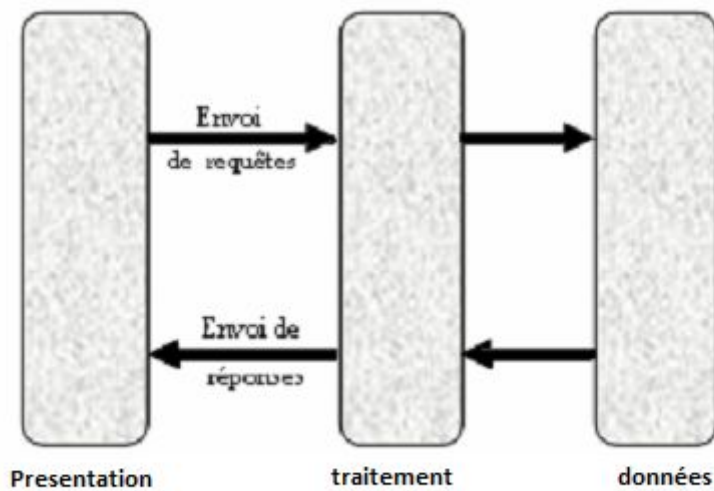
Catégorie et produits

CatandProduct

CategoryID	CategoryName	Description	Picture
1	Beverages	Soft drinks, coffe...	
2	Condiments	Sweet and savor...	
3	Confections	Desserts, candie...	
4	Dairy Products	Cheeses	
5	Grains/Cereals	Breads, crackers,...	
6	Meat/Poultry	Prepared meats	
7	Produce	Dried fruit and be...	
8	Seafood	Seaweed and fish	

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsO
1	Chai	1	1	10 boxes x 20 bags	18,0000	39	0
2	Chang	1	1	24 - 12 oz bottles	19,0000	17	40
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10,0000	13	70
4	Chef Anton's Caj...	2	2	48 - 6 oz jars	22,0000	53	0
5	Chef Anton's Gu...	2	2	36 boxes	21,3500	0	0
6	Grandma's Boyse...	3	2	12 - 8 oz jars	25,0000	120	0
7	Uncle Bob's Orga...	3	7	12 - 1 lb pkgs.	30,0000	15	0
8	Northwode Cran	2	2	12 - 12 oz jars	40,0000	6	0

Architecture applicative en couches



L'architecture en couche est un modèle basé sur un environnement client-serveur.

L'architecture logique du système est divisée en couches :

couche de présentation ;

Elle correspond à la partie visible et interactive de l'application pour les utilisateurs. On parle d'interface homme-machine IHM

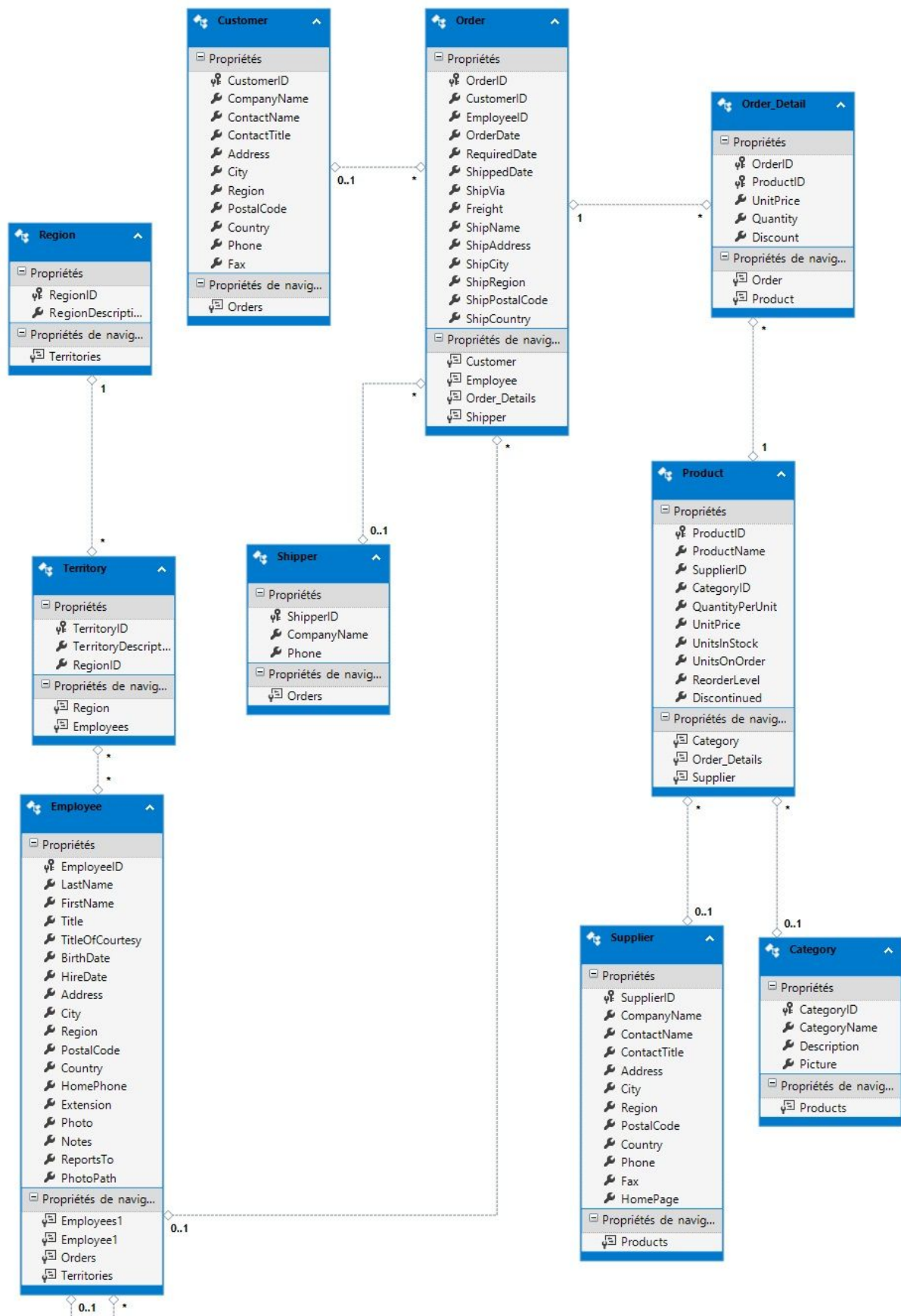
couche de traitement ;

Elle correspond à la partie fonctionnelle de l'application, celle qui implémente la logique métier, et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche de présentation.

couche d'accès aux données.

Elle correspond à la partie gérant l'accès aux données de l'application, dans cette situation elle gère les données provenant de la base de donnée de GSB CE.

Persistance des données



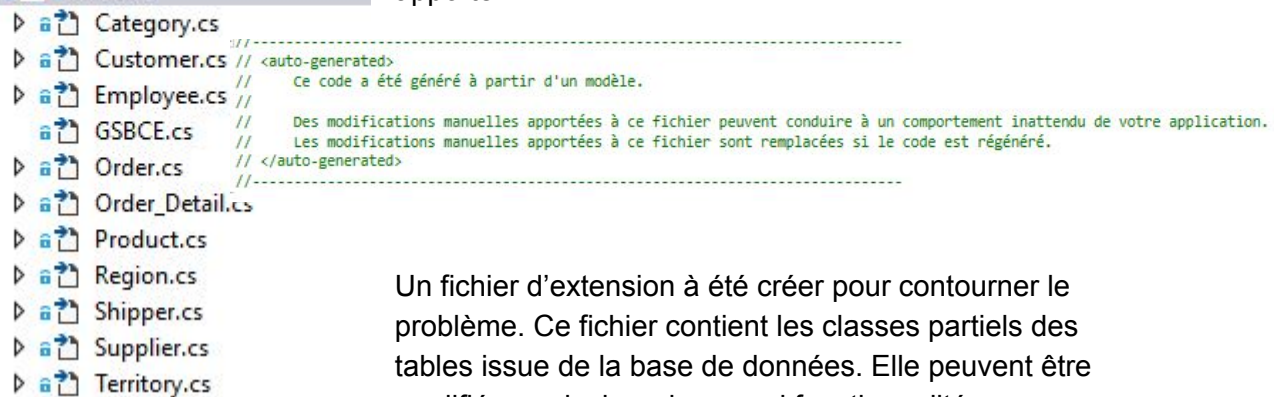
Entity Framework

Entity Framework est un ORM (object-relational mapping) utilisé par la technologie .NET. Un ORM créer automatiquement des classes selon les tables d'une base de données et inversement. peut créer des tables dans la base de données selon les classes.



Ci-dessous nous pouvons voir les classes issues de la base de données GSB CE qui ont été créer par l'entity framework.

Les fichiers classes étant généré automatiquement ne garderont pas les modifications que l'on puisse y apporté.

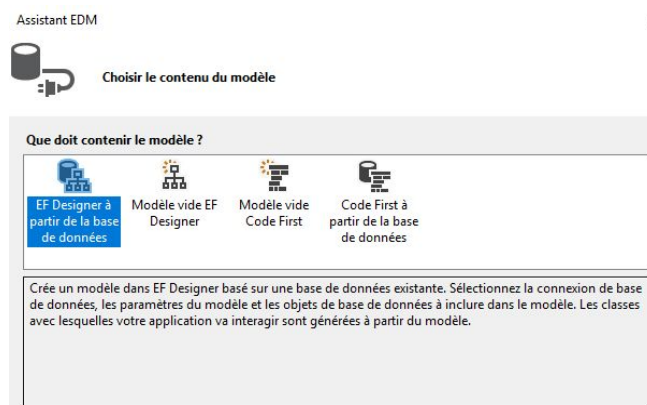


Un fichier d'extension à été créer pour contourner le problème. Ce fichier contient les classes partiels des tables issue de la base de données. Elle peuvent être modifié pour inclure de nouvel fonctionnalité.

L'entity framework est extremement utile pour rapidement établir un lien entre une base de données et l'application ainsi que pour garder les deux contiuellement à jour.

Comment généré le modèle de la base de donnée dans mon application ?

Selectionner le type de modèle à utilisé.



Sélectionner votre base de données puis les tables devant être utilisées.

Assistant EDM
Choisir votre connexion de données

Assistant EDM
Choisir vos paramètres et objets de base de données

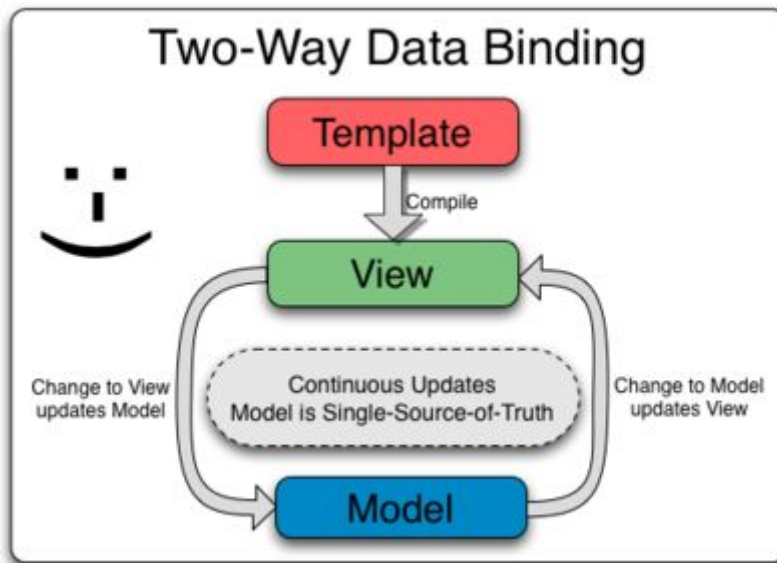
extrait de code auto généré

```
const string inputFile = @"Model1.edmx";
var textTransform = DynamicTextTransformation.Create(this);
var code = new CodeGenerationTools(this);
var ef = new MetadataTools(this);
var typeMapper = new TypeMapper(code, ef, textTransform.Errors);
var loader = new EdmMetadataLoader(textTransform.Host, textTransform);
var itemCollection = loader.CreateEdmItemCollection(inputFile);
var modelNamespace = loader.GetModelNamespace(inputFile);
var codeStringGenerator = new CodeStringGenerator(code, typeMapper);

var container = itemCollection.OfType<EntityContainer>().FirstOrDefault();
if (container == null)
{
    return string.Empty;
}
#>
//-----
// <auto-generated>
// <#=CodeGenerationTools.GetResourceString("Template_GeneratedCo
```

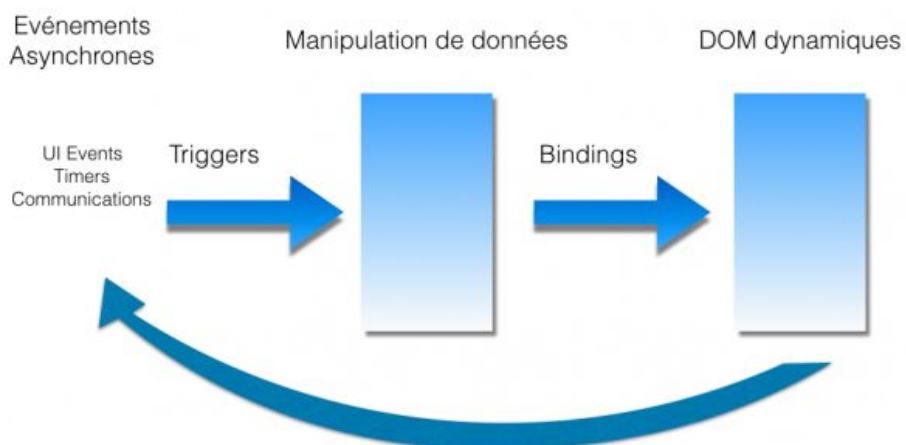

Le DataBinding

Le DataBinding (aussi nommé la liaison de données) est le processus qui établit une connexion entre l'Interface utilisateur de l'application et logique métier. Lorsque la liaison est paramétrée on constate que lorsque les données changent de valeur, les éléments qui sont liés à ces données reflètent automatiquement ces changements.



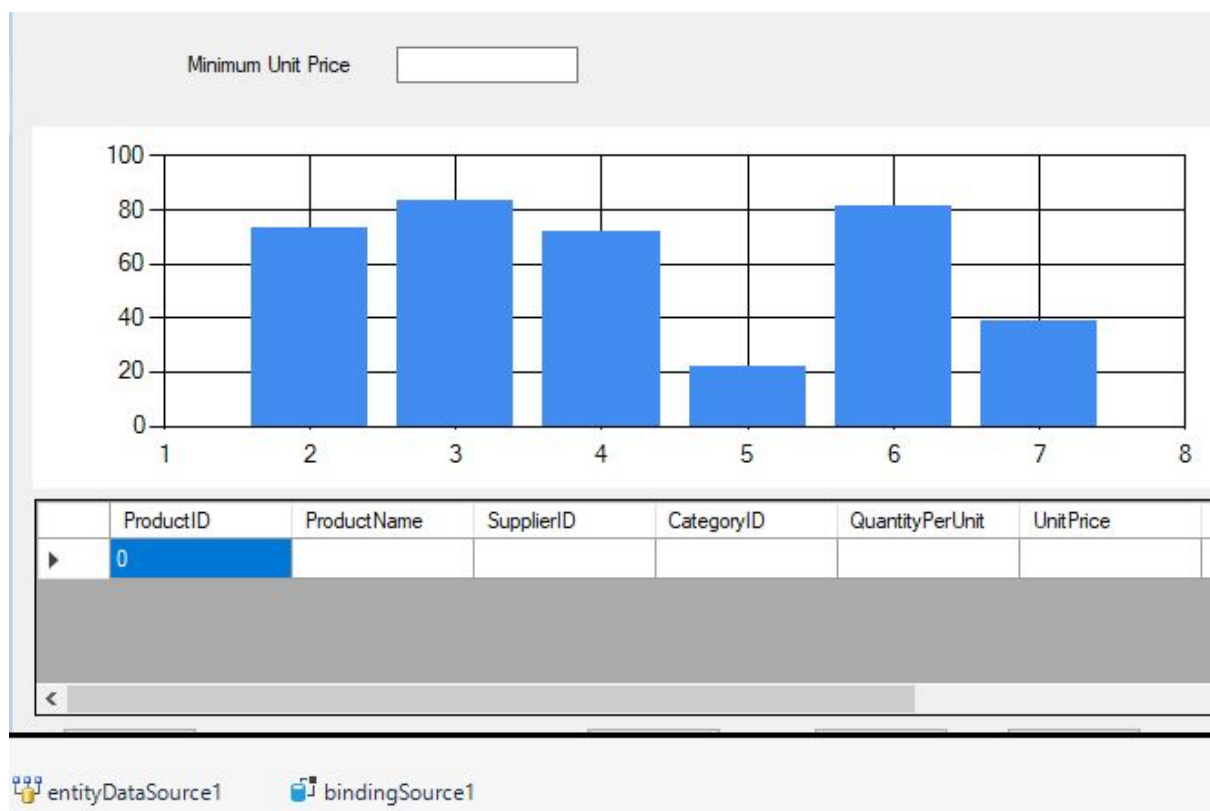
La liaison de données peut également signifier que si une représentation externe des données dans un élément change, les données sous-jacentes peuvent être automatiquement mises à jour pour refléter les modifications.

Par exemple, si l'utilisateur modifie une date de naissance dans un élément TextBox, la valeur de l'âge est automatiquement mis à jour pour refléter ce changement.



```
void ApplyFilter()
{
    var filter = "(Not Discontinued) ";
    //never show discontinued product
    double d;
    var MinPrice = TXT_MinUnitPrice.Text.Trim();
    if (!(double.TryParse(MinPrice, out d)))
    {
        MessageBox.Show("Invalid Minimum");
    }
    else { filter += String.Format(" And (UnitPrice >= {0})", d); }

    //Set filter
    bindingSource1.Filter = filter;
}
```



Le code ci-dessous récupère les données du prix minimum et les transfère à la binding source qui nourrit le graphique et le tableau en temps réel.

Les classes partielles

Les classes partielles nous permettent de diviser les classes en plusieurs parties et en plusieurs fichiers sources. Toutes les pièces sont combinées en une seule classe pendant la compilation.

Nos classes partielles sont regroupées dans la classe DataModelExtension.

Elles nous permettent entre autre de créer des fonctions ne risquant pas d'être supprimées par une régénération de l'application ainsi que de créer et nourrir de nouvelles propriétés tel que le chiffre d'affaire, la quantité ou l'âge.

```
public partial class Order_Detail
{
    public decimal CA()
    {
        return Quantity * UnitPrice * (1 - (decimal)Discount);
    }

    //calculate amount for this order detail
    public decimal Amount
    {
        get
        {
            return Quantity * UnitPrice * (1 - (decimal)Discount);
        }
    }
}

public partial class Order

public partial class Employee
{
    public string FullName
    {
        get
        {
            return string.Format("{0} {1}", FirstName, LastName);
        }
    }

    public override string ToString()
    {
        return FullName;
    }

    public int GetAgeFromBirth
    {
        get
        {
            return (DateTime.Today.Year - BirthDate.Value.Year);
        }
    }
}
```


Les test unitaires

Le test unitaire est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « unité » ou « module »).

Nous avons procédé à des tests unitaires afin de vérifier le fonctionnement des fonction GetAgeFromBirth et GetFullName.

```
namespace TestUnitaire
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void GetAgeFromBirth()
        {
            Employee emp = new Employee();
            var anniv = emp.BirthDate = DateTime.Parse("22/10/1996");

            var age = emp.GetAgeFromBirth;

            if (anniv > DateTime.Today.AddYears(-age))
            {
                age--;
            }

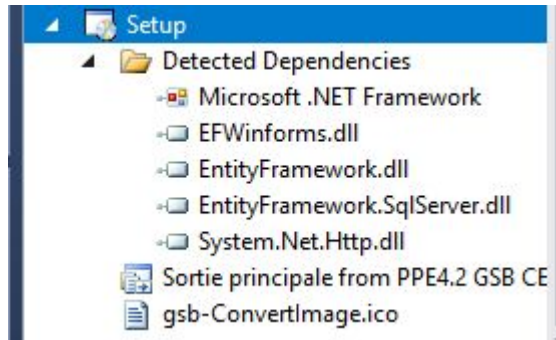
            Assert.AreEqual(22, age, "erreur dans l'age");
        }
    }
}
```



Par exemple le test unitaire suivant teste la fonction présente dans la classe partiel d'employé.

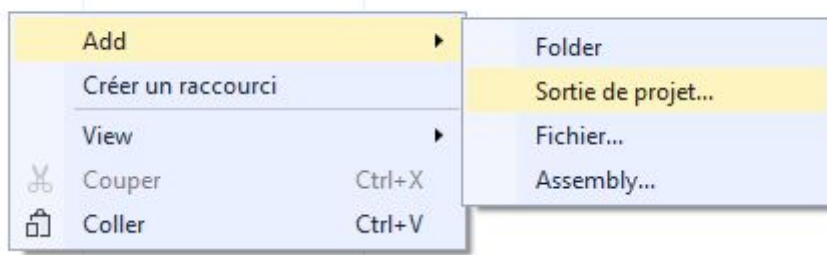
Déploiement

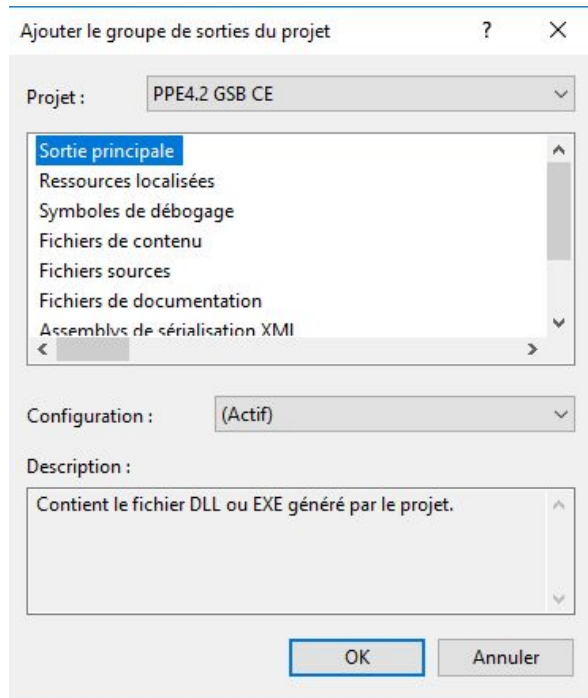
Nous avons utiliser visual studio installer 2013 pour nous servir d'installer.



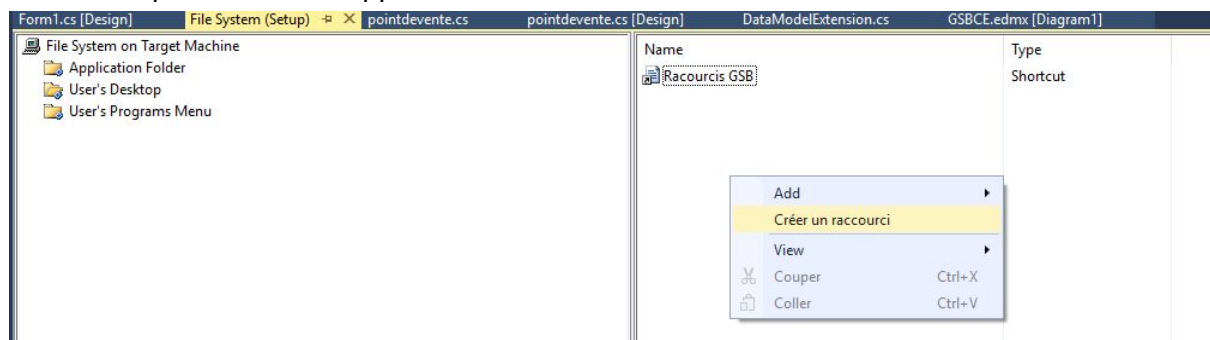
Trois menu devraient apparaitre: application Folder user desktop et user programs.

Dans Application Folder il faut ajouter notre projet comme sortie. Afin que celui-ci soit installé et puisse ainsi se lancer après l'instalation.

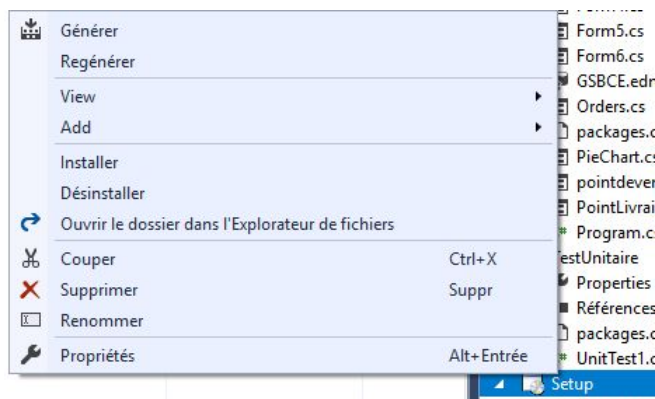




Par la suite dans User Desktop ou programs Menu, créer un raccourcis par lequel l'utilisateur puisse lancer l'application.



Maintenant il ne reste plus qu'à **générer** le projet puis à l'ancer l'executable (l'installateur) qui en résultera.



Problème, proposition, conclusion

Problème:

L'utilisateur ne peut retourner au menu après avoir choisi une fenêtre. Il lui est nécessaire de relancer l'application pour y arriver.

On remarque aussi que "fermer" ma fenetre ne stoppe pas l'application.