

## **PPE 4.2**

GSB - Northwind App



Auteur : Maxime Lalange – Laura Sampaolo  
Date de création : 19/04/2019  
Version : 1.0.0

# **SOMMAIRE**

## 1. Contexte de la situation professionnelle

- Le besoin
- Les objectifs
- L'existant

## 2. Environnement de développement

- IDE mis en œuvre
- Langage(s) utilisé(s)
- Nature du projet
- Environnement de travail collaboratif
- Framework utilisés

## 3. Modélisation fonctionnelle

- Méthodologie mis en œuvre
- Modélisation fonctionnelle graphique (Use Case Diagram UML)
- Modélisation textuelle du cas d'utilisation "Obtenir les commandes d'une zone géographique"

## 4 . IHM

- Capture d'écran des interfaces de l'application

## 5 . Architecture applicative en couches

- Schéma
- Répartition des rôles
- Intérêt de cette architecture

## 6 . Persistance des données

- Schéma de données (MCD ou MLD)
- Extrait de script

## 7 . Entity Framework

- Principes d'un ORM
- Mécanisme des classes partielles (extrait de code)
- Mise à jour de la BDD et du modèle (création vue, maj modèle)
- Extraits de code (instanciation du modèle, requêtes LINQ, persister/annuler les changements, actualiser les données)

## 8 . DataBinding

- Principes
- Extrait de code, mise en œuvre dans les classes de formulaires

## 9 . Faire évoluer le modèle : les Classes Partielles

- Mécanisme de classes partielles
- Doter les entités concernées par l'ajout de nouvelles propriété (Amount, FullName, GetAgeFromBirthDay)

## 10 . Tests unitaires

- Principes des tests unitaires
- Projet de tests unitaires avec test des propriétés FullName et GetAgeFromBirthDay

## 12. Conclusion

- Difficultés rencontrées
- Améliorations possibles

## Contexte de la situation professionnelle

### **Le besoin**

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.

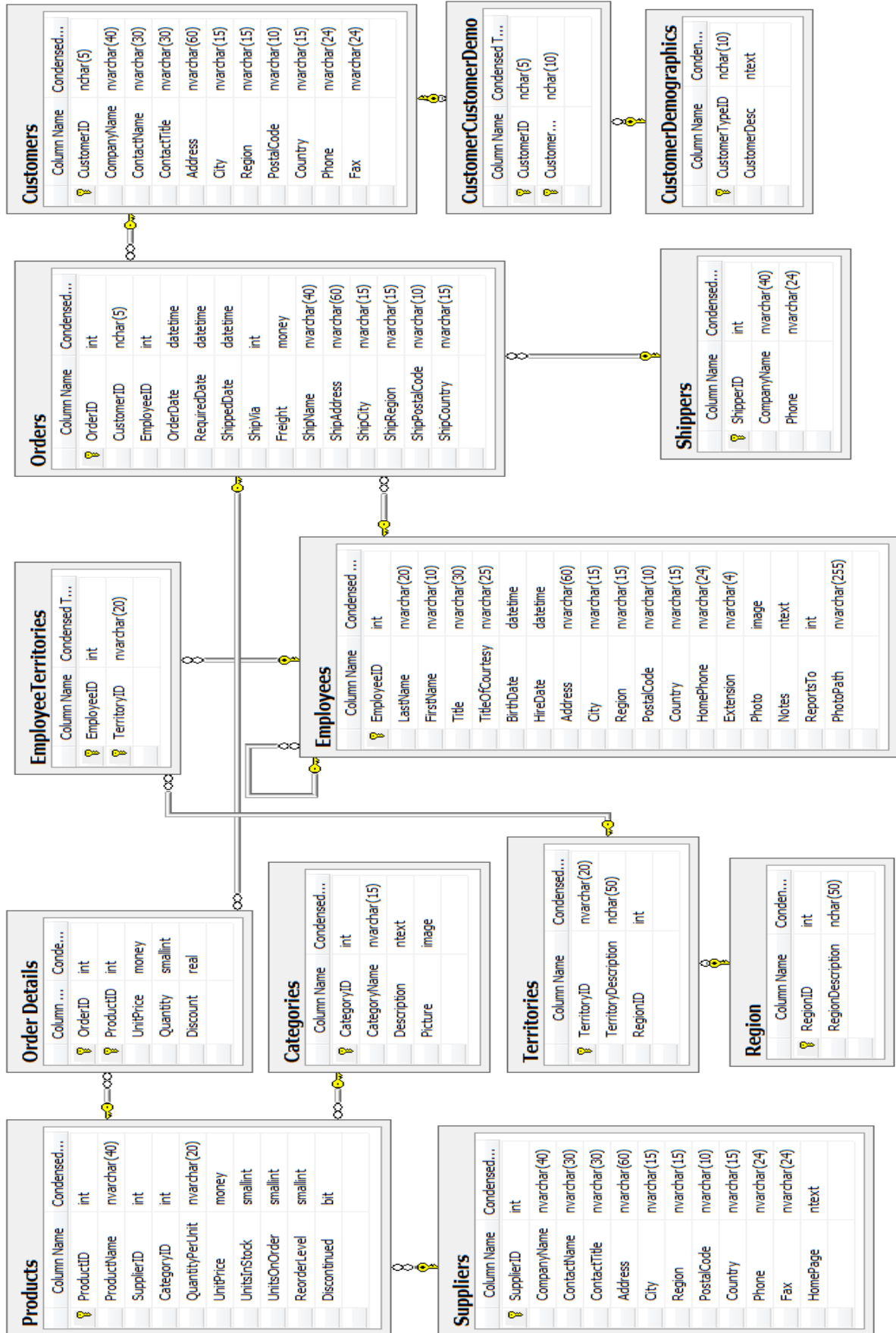
### **Les objectifs**

A cet effet, le comité d'entreprise GSB nous demande de développer une application pour gérer les commandes des collaborateurs du groupe GSB.

### **L'existant**

L'équipe projet a déjà procédé à l'analyse de l'expression des besoins. Cette analyse a permis d'élaborer la modélisation des données. Le résultat de ce travail vous est fourni ci-après :

## Modélisation des données



# Environnement de développement

## IDE mis en œuvre

### Microsoft Visual Studio Entreprise 2017

**Microsoft Visual Studio** est une suite de logiciels de développement pour Windows et Mac OS conçue par Microsoft. La dernière version s'appelle **Visual Studio 2017**.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP.

## Langage(s) utilisé(s)

### C#

**C#** (*C sharp* en anglais) est un langage de programmation orienté objet, commercialisé par Microsoft depuis 2002 et destiné à développer sur la plateforme Microsoft .NET.

Il est dérivé du C++ et très proche du Java dont il reprend la syntaxe générale ainsi que les concepts, y ajoutant des notions telles que la surcharge des opérateurs. Il est utilisé notamment pour développer des applications Web sur la plateforme ASP.NET.

### LINQ (Sugar Syntax – Méthodes extension – Expression Lambda)

**LINQ** (Language-Integrated Query) est le nom d'un ensemble de technologies basées sur l'intégration de fonctions de requête directement dans le langage C#. En règle générale, les requêtes de données sont exprimées comme de simples chaînes, sans vérification de type au moment de l'exécution.

**Sugar Syntax** est une expression imaginée par Peter J. Landin pour désigner les extensions à la syntaxe d'un langage de programmation qui :

- ne modifient pas son expressivité,
- le rendent plus agréable à écrire comme à lire.

Le Sugar Syntax exprime le fait de donner au programmeur des possibilités d'écriture plus succinctes ou plus proches d'une notation usuelle.

**L'expression Lambda** de requête est écrite selon une syntaxe de requête déclarative. En utilisant la syntaxe de requête, il est possible d'effectuer des opérations de filtrage, de classement et de regroupement sur des sources de données avec un minimum de code.

### SQL

**SQL** (Structured Query Language, en français langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

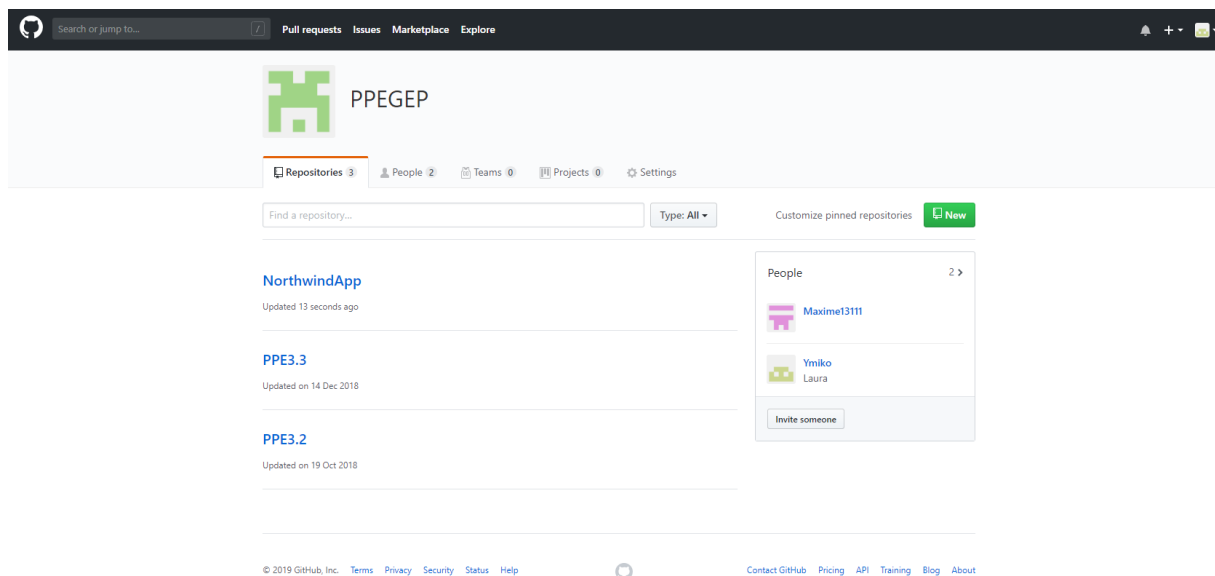
## Environnement d'exploitation

### Windows 10.x

**Windows 10** est un système d'exploitation de la famille Windows développé par la société américaine Microsoft. Officiellement présenté le 30 septembre 2014, il est disponible publiquement depuis le 29 juillet 2015

## Environnement de travail collaboratif

Ce projet a été conçu en mode collaboratif, partagé sur la plateforme GitHub. Github est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de version Git. Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet.



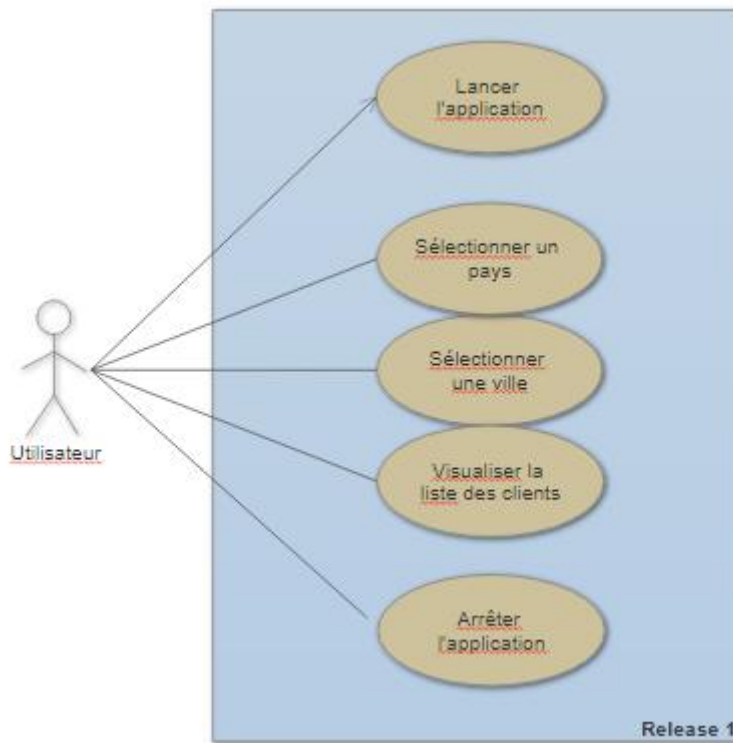
## Framework utilisé(s)

### Application ASP.NET

**ASP.NET** est un Framework permettant de générer à la demande des pages web, lancée par Microsoft en juillet 2000, et utilisée pour mettre en œuvre des applications Web. Il s'agit d'une évolution majeure d'Active Server Pages (ASP, alias Classic ASP), par laquelle cette technique a été incorporée dans la plateforme Microsoft .NET.

## Modélisation fonctionnelle


### Modélisation fonctionnelle graphique (Use Case Diagram UML)



### Modélisation textuelle du cas d'utilisation "Obtenir les commandes d'une zone géographique"

N°	Action	Résultat attendu
1	L'utilisateur lance l'application	Le système affiche le menu principal
2	L'utilisateur clique sur le bouton correspondant au formulaire « Obtenir les commandes d'une zone géographique »	Le système affiche le formulaire « Obtenir les commandes d'une zone géographique » et invite l'utilisateur à choisir un pays
3	L'utilisateur sélectionne un pays dans la liste déroulante	Le système affiche la liste des villes correspondantes au pays sélectionné et invite l'utilisateur à en sélectionner une.
4	L'utilisateur sélectionne une ville dans la liste déroulante	Le système affiche la liste détaillée des clients correspondants à la ville sélectionnée.
5	L'utilisateur visualise la liste détaillée.	

## Capture d'écran des interfaces de l'application



**Entreprise Committee**  
Northwind App

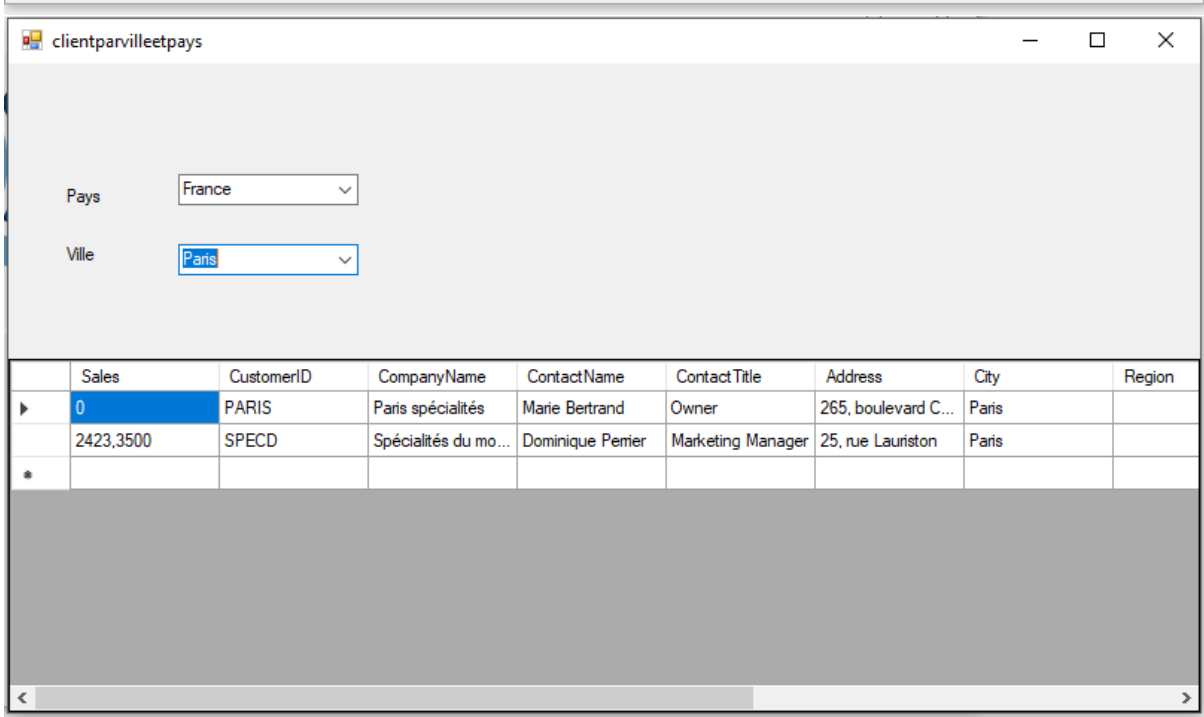
Client par Pays et par Ville      CA par Transporteurs

CA client et commandes      Calendrier

Prix minimum par Produits      Orders

Liste des Produits

Quitter

clientparvilleetpays

Pays: France

Ville: Paris

	Sales	CustomerID	CompanyName	ContactName	Contact Title	Address	City	Region
▶	0	PARIS	Paris spécialités	Marie Bertrand	Owner	265, boulevard C...	Paris	
	2423,3500	SPECD	Spécialités du mo...	Dominique Perrier	Marketing Manager	25, rue Lauriston	Paris	
*								



Form2

1 of 91

Save Undo Refresh

Company Name

Country

Sales

	Amount	OrderID	CustomerID	EmployeeID	OrderDate	Re
▶	814,50 €	10643	ALFKI	6	25/08/2018	22/
	878,00 €	10692	ALFKI	4	03/10/2018	31/
	330,00 €	10702	ALFKI	4	13/10/2018	24/
	845,80 €	10835	ALFKI	1	15/01/2019	12/
	471,20 €	10952	ALFKI	1	16/03/2019	27/

Form3

1 of 830

Save Undo Refresh

Order Date

Customer

Employee

Amount

Details

	Products	Amount	UnitPrice	Quantity	Discount
▶	Queso Cabral... ▼	168,0000	14,0000	12	0
	Singaporean ... ▼	98,0000	9,8000	10	0
	Mozzarella di ... ▼	174,0000	34,8000	5	0

SAVE CANCEL REFRESH

Form4

Minimum UnitPrice

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit
▶	10	Ikura	4	8	12 - 12 oz bags
	26	Gumbär Gummib...	11	3	100 - 100 g bags
	32	Mascarpone Fabioli	14	4	24 - 12 oz jars
<	64	Wimmers aute Se...	12	5	20 - 12 oz jars

SAVE CANCEL REFRESH

menu

	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	

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice
▶	1	Chai	1	1	10 boxes x 20 bags	18.00
	2	Chang	1	1	24 - 12 oz bottles	19.00
	3	Aniseed Syrup	1	2	12 - 550 ml bottles	10.00
	4	Chef Anton's Caj...	2	2	48 - 6 oz jars	22.00

SAVE CANCEL REFRESH

Form5

StartDate      EndDate

<      avril 2019      >

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
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

 Aujourd'hui : 19/04/2019

<      avril 2019      >

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
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

 Aujourd'hui : 19/04/2019

GetOrders

	Amount	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	Shipper
▶	440,0000	10248	VINET	5	04/07/2017	01/08/2017	16/07/2
	1863,4000	10249	TOMSP	6	05/07/2017	16/08/2017	10/07/2
	1552,600000	10250	HANAR	4	08/07/2017	05/08/2017	12/07/2
<	654,000000	10251	VINET	5	09/07/2017	05/08/2017	15/07/2

Form7

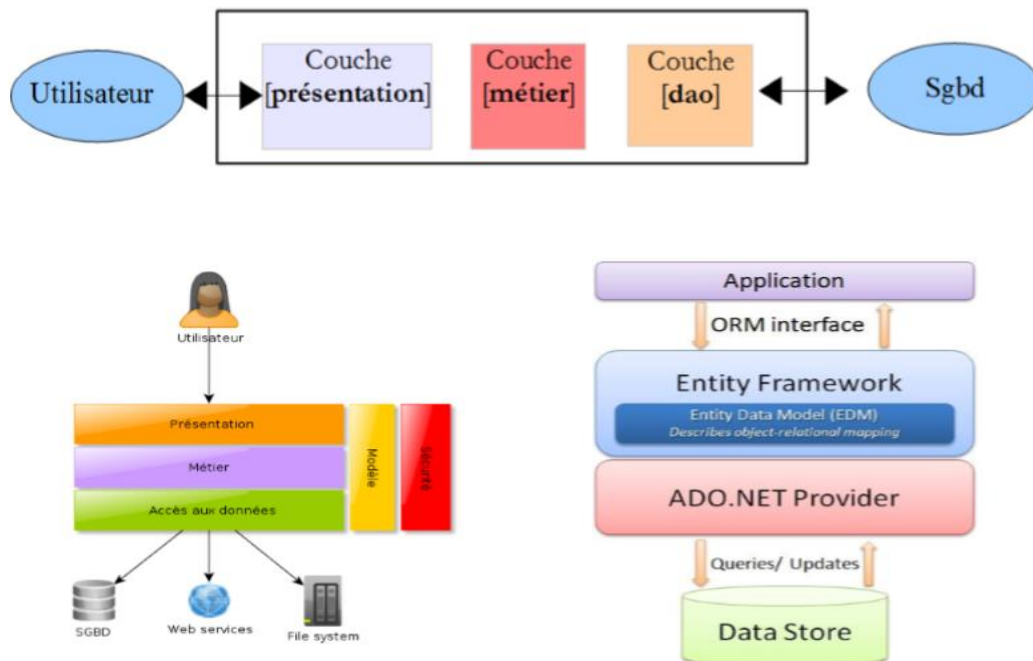
Legend:

- Federal Shipping
- Speedy Express
- United Package

	Transporteur	CA
▶	Federal Shipping	20 512,51 €
	Speedy Express	16 185,33 €
	United Package	28 244,85 €
*		

# Architecture applicative en couches

## Schéma



## Répartition des rôles

### Couche de persistance des données : SQL Server 2016

Elle correspond à la gestion de la persistance des données réfère au mécanisme responsable de la sauvegarde et de la restauration des données. Ces mécanismes font en sorte qu'un programme puisse se terminer sans que ses données et son état d'exécution ne soient perdus.

Ces informations de reprise peuvent être enregistrées sur disque, éventuellement sur un serveur distant (un serveur de bases de données relationnelles, par exemple).

Du fait de la différence de modèles entre les bases de données et les langages de programmation (notamment les langages objet) la notion de correspondance entre modèles (en anglais : *mapping*) est centrale.

### Couche d'accès aux données et couche métier : Entity Framework 6.x

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.

Les différentes règles de gestion et de contrôle du système sont mises en œuvre dans cette couche.

La couche d'accès aux données offre des services applicatifs et métier à la couche de présentation. Pour fournir ces services, elle s'appuie, le cas échéant, sur les données du système, accessibles au travers des services de la couche supérieure. En retour, elle renvoie à la couche de présentation les résultats qu'elle a calculés.

## **Couches présentation : Windows Forms**

Elle correspond à la partie visible et interactive de l'application pour les utilisateurs. On parle d'interface homme-machine. Elle peut être réalisée par une application graphique ou textuelle. Elle peut aussi être représentée en HTML pour être exploitée par un navigateur Web.

La couche de présentation relaie les requêtes de l'utilisateur à destination de la couche de traitement, et en retour lui présente les informations renvoyées par les traitements de cette couche. Il s'agit donc ici d'un assemblage de services métiers et applicatifs offerts par la couche supérieure.

## **Intérêt de cette architecture (trois-tiers)**

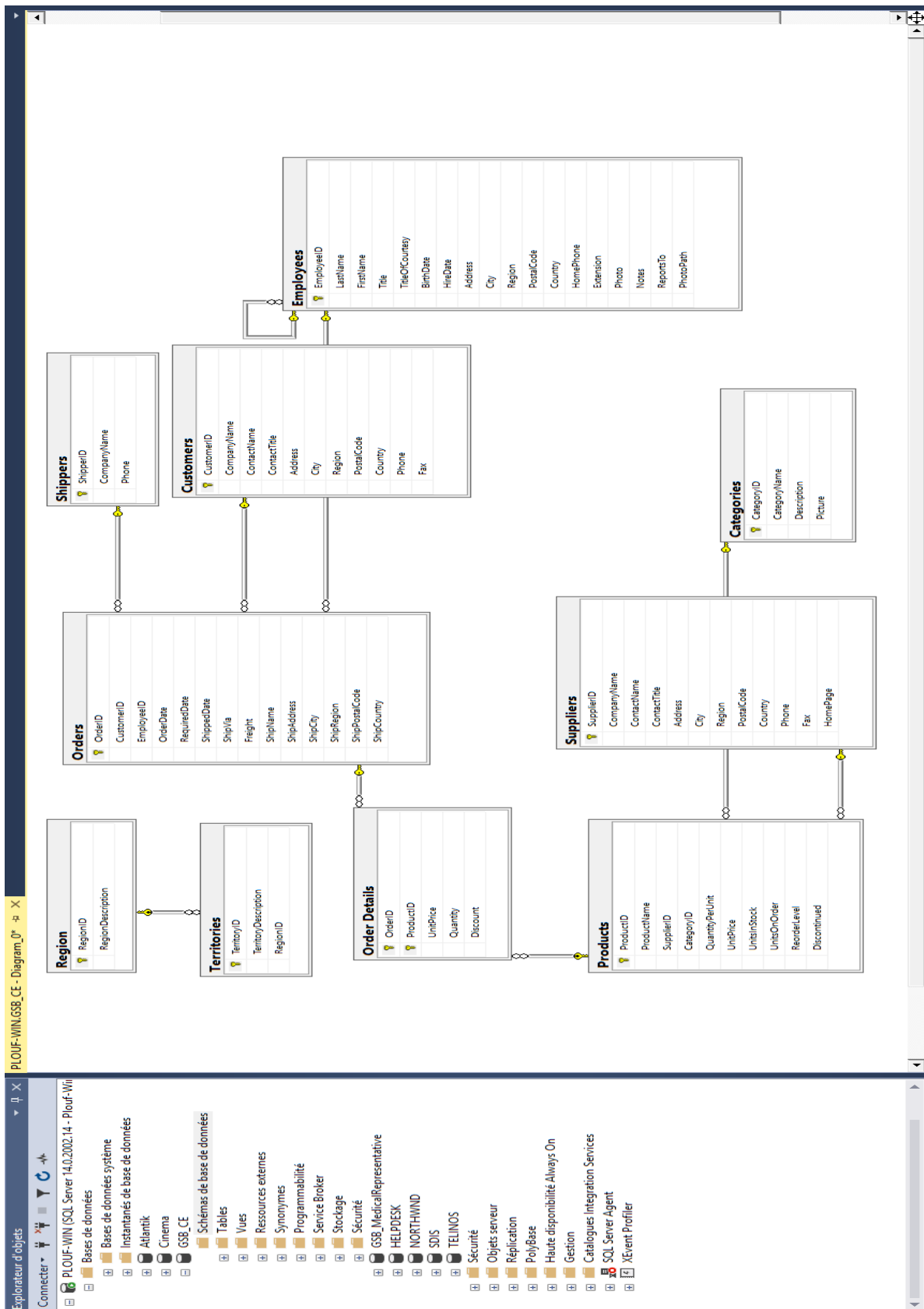
L'objectif premier d'un système d'information quel qu'il soit est de permettre à plusieurs utilisateurs d'accéder aux mêmes informations. Pour cela il faut donc regrouper les informations utilisées par l'entreprise.

En terme technique, cela se traduit par la centralisation des données au sein d'une base de données. L'évolution des systèmes d'information s'est donc basée sur une meilleure subdivision entre les tâches à réaliser pour permettre l'exploitation de ces données par les utilisateurs finaux. Ceci permet de structurer plus efficacement les informations ce qui entraîne à la fois une meilleure organisation de l'entreprise et une meilleure efficacité technique.

Cette subdivision a été facilitée par l'avènement des technologies orientées objets qui s'appliquent aussi bien au modèle client-serveur qu'au modèle Internet. Ces technologies permettent une séparation entre les différents composants du système. Il devient alors possible de réaliser de nouvelles architectures permettant la mise à disposition des informations sous différentes formes tout en diminuant les temps de développement.

# Persistence des données

## Schéma de données (MCD ou MLD)



## Extrait de script

```
script.sql - PLOUF-...louf-Win\Plouf (53)*
USE [GSB_CE]
GO
/***** Object: Table [dbo].[Customers]    Script Date: 22/02/2019 16:50:02 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Customers](
    [CustomerID] [nchar](5) NOT NULL,
    [CompanyName] [nvarchar](40) NOT NULL,
    [ContactName] [nvarchar](30) NULL,
    [ContactTitle] [nvarchar](30) NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [Phone] [nvarchar](24) NULL,
    [Fax] [nvarchar](24) NULL,
    CONSTRAINT [PK_Customers] PRIMARY KEY CLUSTERED
(
    [CustomerID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Suppliers]    Script Date: 22/02/2019 16:50:02 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Suppliers](
    [SupplierID] [int] IDENTITY(1,1) NOT NULL,
    [CompanyName] [nvarchar](40) NOT NULL,
    [ContactName] [nvarchar](30) NULL,
    [ContactTitle] [nvarchar](30) NULL,
    [Address] [nvarchar](60) NULL,
    [City] [nvarchar](15) NULL,
    [Region] [nvarchar](15) NULL,
    [PostalCode] [nvarchar](10) NULL,
    [Country] [nvarchar](15) NULL,
    [Phone] [nvarchar](24) NULL,
    [Fax] [nvarchar](24) NULL,
    [HomePage] [ntext] NULL,
    CONSTRAINT [PK_Suppliers] PRIMARY KEY CLUSTERED
(
    [SupplierID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Suppliers] ADD CONSTRAINT [FK_Order_Details_Suppliers] FOREIGN KEY([ProductID])
REFERENCES [dbo].[Products] ([ProductID])
```

## Entity Framework

### Principes d'un ORM

Un mapping objet-relationnel (en anglais object relational mapping ou ORM) est un type de programme informatique qui se place en interface entre un programme applicatif et une base de données relationnelle pour simuler une base de données orienté objet. Ce programme définit des correspondances entre les schémas de la base de données et les classes du programme applicatif. On pourrait le désigner par-là, « comme une couche d'abstraction entre le monde objet et monde relationnel ». Du fait de sa fonction, on retrouve ce type de programme dans un grand nombre de Framework sous la forme de composant ORM qui a été soit développé, soit intégré depuis une solution externe.

## Les Classes Partielles

### Mécanisme de classes partielles

Il est possible de fractionner la définition d'une classe, d'une interface ou d'une méthode entre plusieurs fichiers sources. Chaque fichier source contient une section de la définition de méthode ou de type, et toutes les parties sont combinées au moment où l'application est compilée.

### Doter les entités concernées par l'ajout de nouvelles propriété (Amount, FullName, GetAgeFromBirthDay)

8 références

```
public partial class Employees
```

```
{
```

1 référence

```
public string FullName
```

```
{
```

```
get
```

```
{
```

```
return string.Format("{0} {1}", FirstName, LastName);
```

```
}
```

```
}
```

0 références

```
public override string ToString()
```

```
{
```

```
return FullName;
```

```
}
```

```
}
```

7 références

```
public partial class Customers
```

```
{
```

0 références

```
public decimal Sales
```

```
{
```

```
get
```

```
{
```

```
//format sugar
```

```
var q = from cu in this.Orders select cu.Amount;
```

```
return q.Sum();
```

```
}
```

```
//return Order_Details.Sum(n => n.Amount);
```

```
}
```

```
}
```

```
}
```



```

10 références
public partial class Orders
{
    1 référence
    public decimal Amount
    {
        get
        {
            //format sugar
            var q = from od in this.Order_Details select od.Amount;
            return q.Sum();

            //return Order_Details.Sum(n => n.Amount);
        }
    }
}

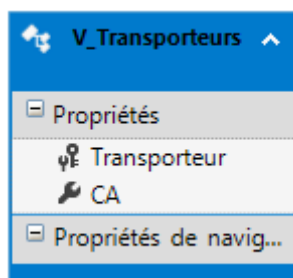
```

## Mise à jour de la BDD et du modèle (création vue, maj modèle)

```

CREATE VIEW V_Transporteurs
AS
SELECT CompanyName AS Transporteur, ROUND(SUM(Freight),2) AS CA FROM Orders INNER JOIN Shippers ON ShipperID = ShipVia GROUP BY CompanyName;

```



## Extraits de code (instanciation du modèle, requêtes LINQ, persister/annuler les changements, actualiser les données)

```

0 références
static class Program
{
    /// <summary>
    /// Point d'entrée principal de l'application.
    /// </summary>
    [STAThread]
    0 références
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new menu());
    }
}

```

1 référence

```
private void button1_Click(object sender, EventArgs e)
{
    Form1 f1 = new Form1();
    f1.ShowDialog();
}
```

1 référence

```
private void button2_Click(object sender, EventArgs e)
{
    Form2 f2 = new Form2();
    f2.ShowDialog();
}
```

1 référence

```
private void button3_Click(object sender, EventArgs e)
{
    Form3 f3 = new Form3();
    f3.ShowDialog();
}
```

1 référence

```
private void button4_Click(object sender, EventArgs e)
{
    Form4 f4 = new Form4();
    f4.ShowDialog();
}
```

1 référence

```
private void button5_Click(object sender, EventArgs e)
{
    Form5 f5 = new Form5();
    f5.ShowDialog();
}
```

```

1 référence
private void button6_Click(object sender, EventArgs e)
{
    clientparvilleetpays f6 = new clientparvilleetpays();
    f6.ShowDialog();
}

1 référence
private void button7_Click(object sender, EventArgs e)
{
    Form7 f7 = new Form7();
    f7.ShowDialog();
}

1 référence
private void button8_Click(object sender, EventArgs e)
{
    this.Close();
}
}

```

## DataBinding

### Principes

Le databinding se traduit en français par « liaison de données ». Il s'agit de la possibilité de lier un contrôle à des données. Le principe consiste à indiquer à un contrôle où il peut trouver sa valeur et celui-ci se débrouille pour l'afficher.

Le binding permet de positionner automatiquement des valeurs aux propriétés des contrôles en fonction du contenu de la source de données. En effet, il est très fréquent de mettre des valeurs dans des TextBox, dans des TextBlock ou dans des ListBox. Le binding est là pour faciliter tout ce qui peut être automatisable et risque d'erreurs. De plus, si la source de données change, il est possible de faire en sorte que le contrôle soit automatiquement mis à jour. Inversement, si des modifications sont faites depuis l'interface, alors on peut être notifié automatiquement des changements.

## Extrait de code, mise en œuvre dans les classes de formulaires

4 références

```
public partial class Form4 : Form
```

```
{
```

1 référence

```
public Form4()
```

```
{
```

```
    InitializeComponent();
```

```
}
```

2 références

```
void ApplyFilter()
```

```
{
```

```
    //never show discontinued products
```

```
    var filter = "(Not Discontinued)";
```

```
    //apply minimum price condition
```

```
    var minPrice = _txtMinPrice.Text.Trim();
```

```
    if (!string.IsNullOrEmpty(minPrice))
```

```
    {
```

```
        double d;
```

```
        if (!double.TryParse(minPrice, out d))
```

```
        {
```

```
            MessageBox.Show("Invalid Minimum Unit Price, please try again");
```

```
        }
```

```
        else
```

```
        {
```

```
            filter += string.Format(" and (UnitPrice >= {0})", d);
```

```
        }
```

```
    }
```

```
    //set the filter
```

```
    chartBindingSource.Filter = filter;
```

```
}
```

1 référence

```
private void _txtMinPrice_Validated(object sender, EventArgs e)
```

```
{
```

```
    ApplyFilter();
```

```
}
```

1 référence

```
private void _txtMinPrice_KeyPress_1(object sender, KeyPressEventArgs e)
```

```
{
```

```
    if (e.KeyChar == 13)
```

```
    {
```

```
        ApplyFilter();
```

```
        e.Handled = true;
```

```
    }
```

```
}
```

1 référence

```
private void chartBindingSource_ListChanged(object sender, ListChangedEventArgs e)
```

```
{
```

```
    chart1.DataBind();
```

```
}
```

1 référence

```
private void btnSave_Click(object sender, EventArgs e)
```

```
{
```

```
    entityDataSource1.SaveChanges();
```

```
}
```

1 référence

```
private void btnCancel_Click(object sender, EventArgs e)
```

```
{
```

```
    entityDataSource1.CancelChanges();
```

```
}
```

1 référence

```
private void btnRefresh_Click(object sender, EventArgs e)
```

```
{
```

```
    entityDataSource1.Refresh();
```

```
}
```

# Tests unitaires

## Principes des tests unitaires

En programmation, le test unitaire (ou « T.U. », ou « U.T. » en anglais) ou test de composants est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme.

## Projet de tests unitaires avec test des propriétés FullName et GetAgeFromBirthDay

```
namespace GSB_CE_TEST
{
    [TestClass]
    0 références
    public class UnitTest1
    {
        [TestMethod]
        0 références
        public void TestFullName()
        {
            Employees employee = new Employees();
            employee.FirstName = "Julie" ;
            employee.LastName = "Yu";
            string expected = "Julie Yu";
            Assert.AreEqual(expected, employee.FullName, "erreur");
        }

        0 références
        public void TestGetDayBirth()
        {
            Employees employee = new Employees();
            employee.BirthDate = new DateTime(2000, 05, 06);
            int expected = 18;
            Assert.AreEqual(expected, employee.GetAgeFromBirthDay, "erreur");

            Employees employee1 = new Employees();
            employee1.BirthDate = new DateTime(2000, 04, 18);
            int expected1 = 19;
            Assert.AreEqual(expected1, employee1.GetAgeFromBirthDay, "erreur");
        }
    }
}
```

## Conclusion

### Difficultés rencontrées

Lors du développement de cette application, l'équipe technique a rencontré quelques difficultés à la conception des Tests unitaires ainsi qu'aux Filtrages dans les différentes interfaces.

### Améliorations possibles

Pouvoir utiliser les statistiques représentées dans l'application sur un intranet via une nouvelle couche web à la racine du projet.

Le WindowsForm utilisé dans ce projet, étant dépassé, il serait possible d'utiliser Windows Presentation Foundation (WPF) (nom de code Avalon) qui est la spécification graphique de Microsoft .NET 3.0. Il intègre le langage descriptif XAML qui permet de l'utiliser d'une manière proche d'une page HTML pour les développeurs.