



École supérieure privée bac à bac+5
Centre de formation pour professionnels
Lyon - Villeurbanne



Conception et Développement d'une Application desktop et d'un site web pour un expert comptable

Rédigé par :
SAGHOUR Khalil

Entreprise :
Lc2l-informatique

Année :
2019-2020

Développement d'une Application desktop pour un expert comptable

Deuxième application :
Page 33.

Sommaire

1 - Introduction

1.1 - Synopsis	p.6
1.2 - Compétences du référentiel couvertes par le projet	p.7

2 - Présentation du projet

2.1 - Cahier des charges	p.8
2.1.1 - Contexte	p.8
2.1.2 - Présentation du client et du besoin	p.8
2.1.3 - Objectifs	p.8
2.1.4 - Mode d'accès de l'application	p.9
2.2 - Spécifications	p.9
2.2.1 - Fonctionnelles	p.9
2.2.2 - Techniques	p.10
2.3 - Gestion de projet	p.11
2.3.1 - Diagramme de GANTT	p.11
2.3.2 - Outils	p.12
2.3.3 - Cycle en V	p.13

3 - Réalisation

3.1 - Conception du projet	p.14
3.1.1 - Veille Technologique	p.14
3.1.2 - Diagramme de Séquence	p.15
3.2 - Maquettage des application	p.18
3.2.1 - Maquette desktop	p.19

3.3 -Développement des applications _____ p.20

3.3.1 - Mise en place de l'environnement _____ p.20

3.3.2 - Développement de l'application Desktop _____ p.24

4 - Conclusion

4 - Conclusion _____ p.32

Que tous ceux, qui de près ou de loin,
ont aidé ou encouragé mon efforts
trouvent ici le témoignage de ma plus sincère gratitude.

Introduction

1.1 Synopsis

Good morning everyone,

First of all, please excuse me for my accent and the mistakes I may make in English. I don't speak Shakespeare's language as well as Molière's. And on the other hand, I would like all those who, directly or indirectly, have encouraged my effort to find here the testimony of my most sincere gratitude.

My name is Khalil Saghour, I am 25 years old. So I have been coding daily for almost 3 years now.

At the end of 2019, thanks to the Region and Pole Emploi, I had the opportunity to start training as an Application Developer Designer.

I worked alone from initial conception to implementation. Even if there were no written specifications, I worked closely with my internship supervisor by using the Agile method to ensure that the platform, as the project progressed.

If I had to quickly summarize the skills I have acquired, I would tell you this:

I know HTML5, CSS3, BOOTSTRAP4, JAVASCRIPT5, JQUERY3, PHP7, SQL, CSHARP, JAVA, MERISE, UML and GIT. I know how to use SublimeText3, VS Code, Visual Studio, Blend, Eclipse, IntelliJ IDEA, Swing, JavaFx, Android Studio, Wampserver, MySQL, SQLServer, Oracle, Balsamiq, PowerAMC, ArgoUML.

1.2 Compétences du référentiel couvertes par le projet

Activité-type 1: Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

- Maquetter une application
- Mise en place de l'environnement
- Développer une interfaces de types desktop

Activité-type 2 : Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Activité-type 3 : Concevoir et développer une application multicouche répartie en intégrant les recommandation de sécurité

- Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement.
- Concevoir une application.

Présentation du projet

2.1 Cahier des charges

2.1.1 Contexte

J'ai eu l'opportunité d'effectuer mon stage chez lc2l-informatique/
La Batisse,
de avril à aout 2020 sous la direction de mon maître de stage,
M. DEPORT Franck.

Lc2l-informatique est une startup au service des petits producteurs,
des TPE et des PME du secteur agroalimentaire, sis 87
Rue de la Picardière 01300 Virignin France.

2.1.2 Présentation du client et du besoin

La Batisse est une Société Coopérative et Participative régie selon les règles de
l'Economie Sociale et Solidaire et membre de l'URSCOP .

Afin d'organiser ses projets et ses tâches, La Batisse souhaite créer une
application desktop de conversion de fichier Xml et Kml en Csv et qui
permettrait d'améliorer sa productivité.

Dans la continuité d'amélioration de productivité, l'application doit pouvoir
permettre aussi de visualiser un aperçu des données du fichier sélectionné.

L'application doit être simple d'utilisation et très intuitive pour les
utilisateurs.

2.1.3 Objectifs

L'objectif du projet est d'améliorer la productivité et l'efficacité du travail des
collaborateurs internes de l'agence.

L'application sera réservée dans un premier temps aux équipes internes.

2.1.4 Mode d'accès des applications

Application Desktop

L'application doit être accessible qu'en interne dans un premier temps. Elle doit être déployée sur un serveur local auquel les collaborateurs auront accès.

toutefois cette dernière partie sera développée par mon maitre de stage.

2.2 Spécifications

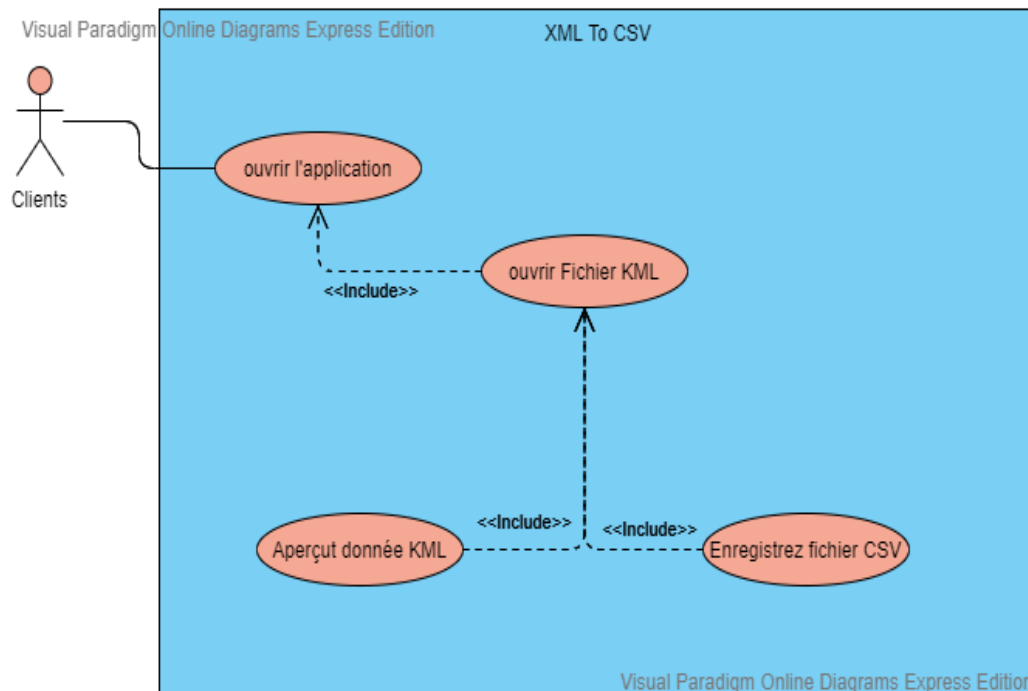
2.2.1 Fonctionnelles

Cette analyse démontre les différentes fonctions de l'application desktop.

a) Fonctionnalité de l'application:

- Sélectionner un fichier (KML ou XML).
- Avoir un aperçut des données.
- Enregistrez le fichier en format CSV.

Voici un diagramme des cas d'utilisation



2.2.2 Techniques

Application Desktop

- Language: .NET.
- Framework 4.7.2.

2.3 Gestion de projet

2.3.1 Diagramme de GANTT

Ce diagramme permet de lister les tâches à réaliser par semaine et est divisé en 2 étapes pour réaliser le projet en son entièreté.

XMLToCSV

5 mai 2020

<http://>

Chef de projet
Dates du projet

4 mai 2020 - 9 mai 2020

Avancée
Tâches
Ressources

0%
2
0

XMLToCSV

5 mai 2020

2

Tâches

Nom	Date de début	Date de fin
Analyse Technique du projet	04/05/20	05/05/20
coder programme de conversion	06/05/20	08/05/20



2.3.2 Cycle en V

Le développement de l'application desktop suit un processus séquentiel. L'utilisation du cycle en V permet de bien déterminer les étapes importantes et de respecter les objectifs qualités.

Il prend en compte 7 étapes:

1. L'analyse du besoin :

Il s'agit de l'expression du besoin client, c'est-à-dire la description de l'utilisation de l'application tel qu'il l'imagine. L'analyse du besoin doit s'accorder sur ce qui doit être fait dans le système.

2. Spécification fonctionnelle :

Elle correspond à tous les cas d'utilisations de l'application.

3. Spécification technique :

Traduction des spécifications fonctionnelles en termes techniques.

Sélection des technologies à mettre en oeuvre pour le développement de l'application.

4. Réalisation:

Phase de réalisation pendant laquelle sont développés les composants qui sont ensuite assemblés pour créer l'application finie.

5. Tests unitaire:

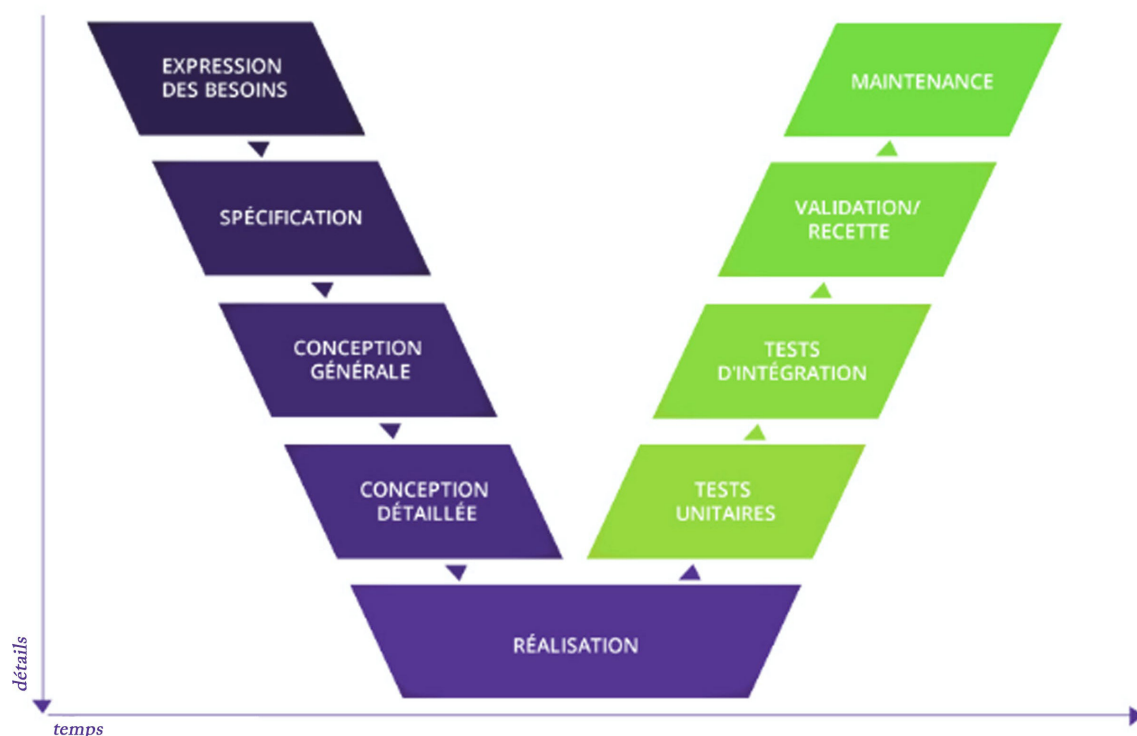
Réalisation de test sur les différentes parties codées : un test par fonctionnalité, pour se projet il à étai décider qu'il n'y aura pas de test.

6. Tests d'intégration:

Réalisation de test sur l'ensemble de l'application afin d'assurer qu'il suit les indications des spécifications techniques, pour se projet il à étai décider qu'il n'y aura pas de test.

7. Validation/Recette:

Avant mise en production, le client procède à la recette pour vérifier et valider que son expression de besoin est respectée.



Réalisation

3.1 Conception du projet

3.1.1 Veille Technologique

C'est dans le cadre de mon travail pour Lc2l-informatique, que j'ai pu développer avec le Framework .NET 4.7.2, en effet .NET Framework est, comme son nom l'indique à moitié, à la fois une framework et un environnement runtime. Une framework est une collection de code tout fait et prêt à l'usage pour les développeurs. Il leur permet donc de réutiliser du code existant, non seulement pour gagner du temps.

Mais aussi pour permettre une certaine standardisation entre les applications. Le codeur peut alors se concentrer sur le code qui rend réellement son application unique.

.NET Framework est aussi une environnement runtime.

Un peu comme les iFrame en HTLM, cet environnement est indépendant du système. Il crée un cadre dans lequel les applications créées via .NET Framework peuvent se lancer indépendamment du système d'exploitation. Ce qui représente un intérêt évident : on peut lancer les applications codées avec .NET Framework sur n'importe quel système d'exploitation où .NET Framework peut être installé.

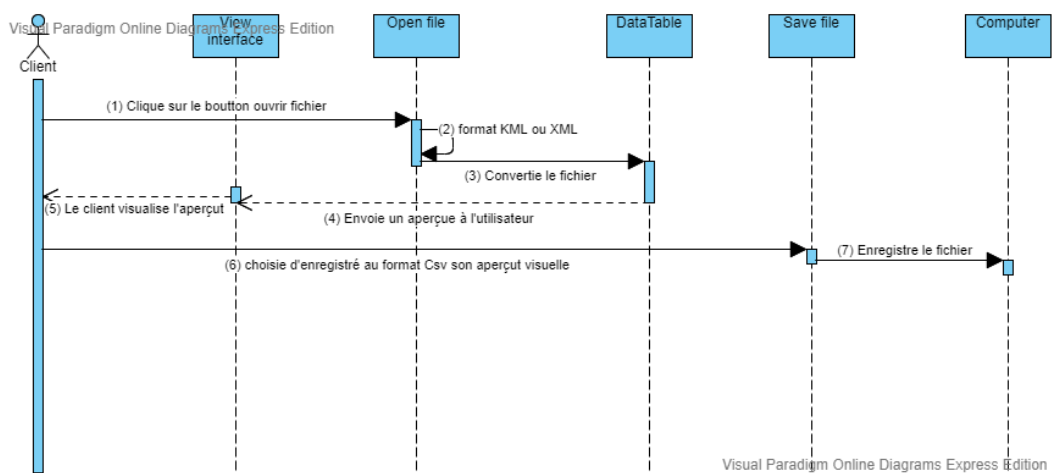
Avantages

deux des nombreux avantage d'utiliser .NET

- Installation plus facile d'applications commerciales.
De nombreuses applications commerciales de Microsoft et de tierces parties reposent sur le .NET Framework.
- Mises à jour plus simples.
Une fois le .NET Framework installé, il est automatiquement mis à jour par le service de mises à jour Microsoft.

3.1.2 Diagramme de Séquence

Pour représenter le fonctionnement de l'application, j'ai utilisé l'UML (Langage de modélisation unifié) afin de créer un schémas qui me permet d'apercevoir l'architecture logicielle de la future application.



1. L'utilisateur Clique sur le bouton ouvrir fichier.



2. Mon application filtre les formats afin d'être sur, que l'utilisateur Sélectionne un format XML ou Kml.

```

75  /// <summary>
76  /// Open File Dialog
77  /// </summary>
78  2 références
79  private void OpenXmlFile()
80  {
81  // open file dialog
82  using(OpenFileDialog open = new OpenFileDialog())
83  {
84  // filters
85  open.Title = "selectionner un fichier";
86  open.Filter = "Tout Les Fichiers (*.*)|*.*|Fichier XML(*.xml)|*.xml|Fichier KML(*.kml)|*.kml";
87  if (open.ShowDialog() == DialogResult.OK)
88  {
89  // xml file path
90  textBoxCheminFichierXml.Text = open.FileName;
91  }
92  }
93  }

```

3. Je récupère les données du fichier afin de l'insérer dans ma Datable.

```

148  /// <summary>
149  /// Add row in datatable
150  /// </summary>
151  /// <param name="dataTable"></param>
152  /// <param name="masterName"></param>
153  /// <param name="adress"></param>
154  /// <param name="dataName"></param>
155  /// <param name="dataValue"></param>
156  1 référence
157  private void FillDataTableRow(DataTable dataTable, DataRow datarow, string dataName, string dataValue)
158  {
159  if (!dataTable.Columns.Contains(dataName))
160  {
161  dataTable.Columns.Add(dataName);
162  }
163  datarow[dataName] = dataValue.Replace("\n", " ");
164  }
165  }

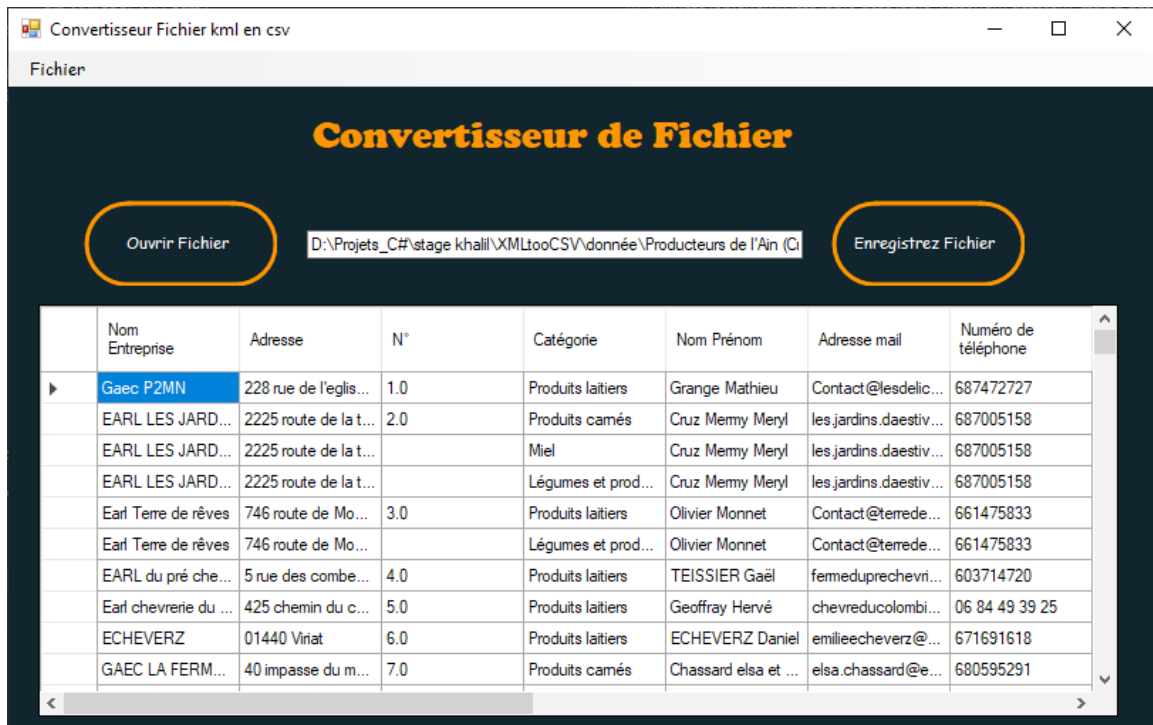
```

```

128  /// <summary>
129  /// add row entreprise and adress in datatable
130  /// </summary>
131  /// <param name="dataTable"></param>
132  /// <param name="masterName"></param>
133  /// <param name="adress"></param>
134  1 référence
135  private DataRow AddDataTableRow(DataTable dataTable, string masterName, string adress)
136  {
137  DataRow row = dataTable.NewRow();
138  if (!dataTable.Columns.Contains("Nom Entreprise") ||
139  !dataTable.Columns.Contains("Adresse"))
140  {
141  dataTable.Columns.Add("Nom Entreprise");
142  dataTable.Columns.Add("Adresse");
143  }
144  row[0] = masterName;
145  row[1] = adress;
146  dataTable.Rows.Add(row);
147  return row;
148  }

```


4. J'envoie un aperçut à l'utilisateur afin qu'ils puissent confirmer les données dans ma DataTable.



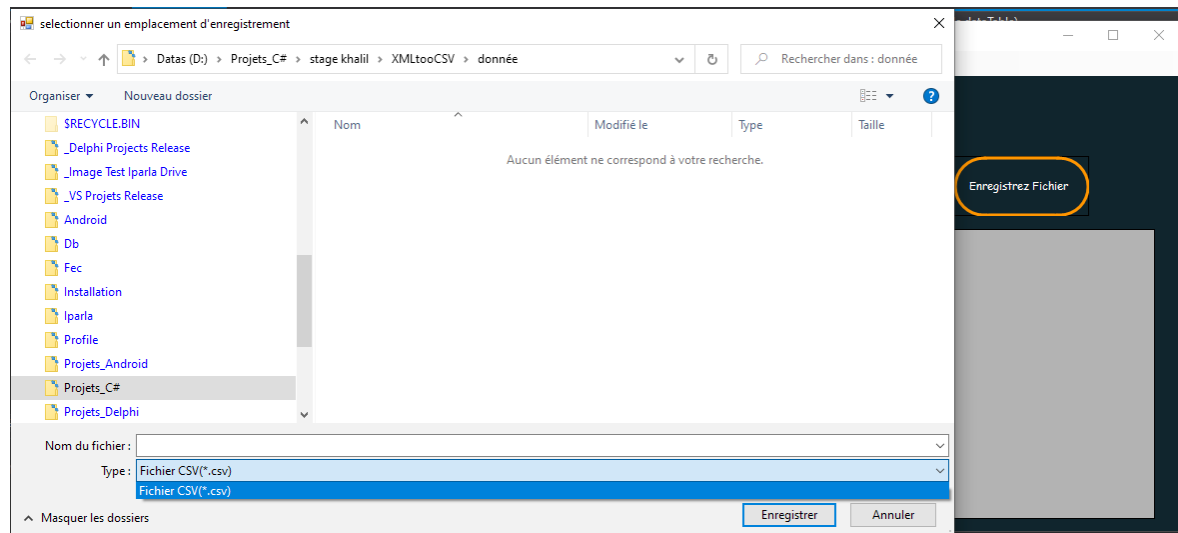
5. L'utilisateur visualise les données et décide d'enregistrer les Données au format Csv, on intègre un filtre afin d'être sur que le format sera en .CSV.

```

196     /// <summary>
197     /// Openl Location File for Csv
198     /// </summary>
199     1 référence
200     private void OpenLocationCsvFile()
201     {
202         // save file dialog
203         SaveFileDialog save = new SaveFileDialog();
204
205         // filters
206         save.Title = "selectionner un emplacement d'enregistrement";
207         save.Filter = "Fichier CSV(*.csv)|*.csv";
208         if (save.ShowDialog() == DialogResult.OK)
209         {
210             // csv file path
211             textBoxCheminFichierXml.Text = save.FileName;
212         }
213     }

```

6. Enfin l'utilisateur enregistre son fichier à l'emplacement qu'il souhaite sur son pc.



3.2 Maquettage des applications

Le maquettage est l'étape indispensable, car elle force à se poser de nombreuses questions pour affiner la structure de l'application, l'organisation interne des pages et leur maillage.

L'objectif de cette étape est de proposer un premier visuel d'interfaces conformes aux attentes et aux besoins.

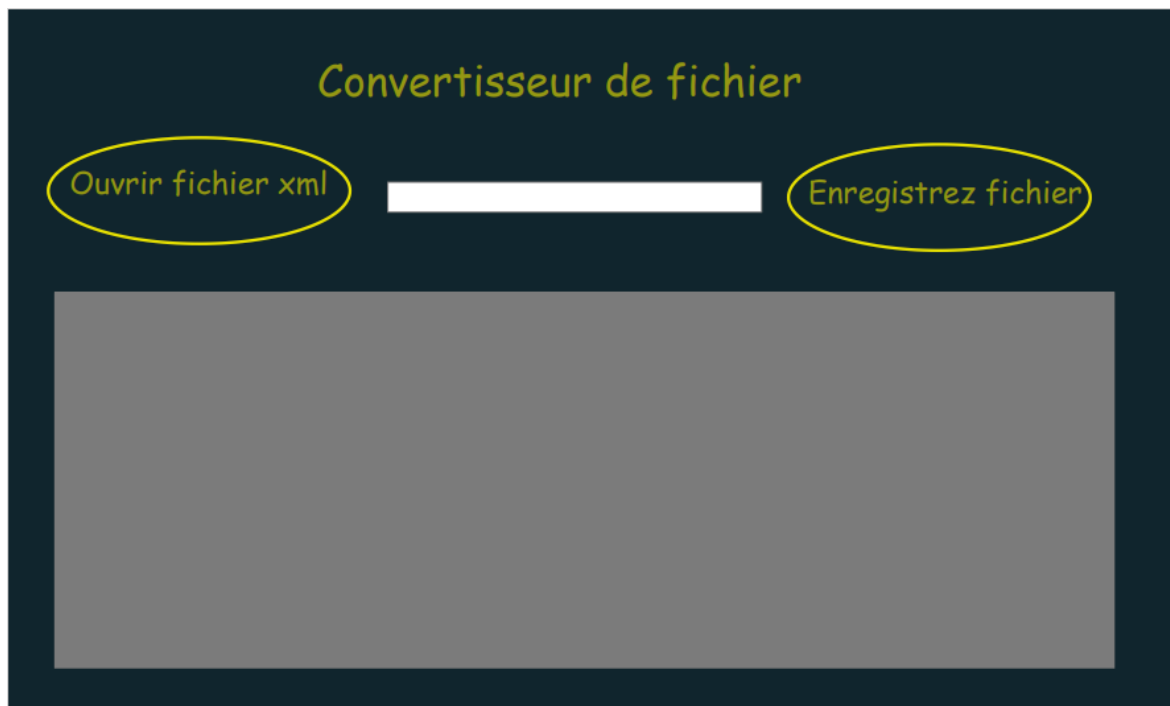
Il a fallu penser à la structure de l'application Desktop en hiérarchisant les différents contenus, afin de faciliter au maximum l'accès aux informations. En m'appuyant sur le cahier des charges, j'ai tout d'abord réalisé les maquettes des interfaces graphiques grâce au logiciel Adobe XD.

3.2.1 Maquette Application

Wiframe



Prototype



L'objectif de l'application étant de faciliter le travail de Comptable, l'application entière devait être simple visuellement, intuitif et facile d'utilisation.

3.3 Développement de l'application

3.3.1 Mise en place de l'environnement

Machine Virtuelle

En premier temps, j'ai téléchargé un logiciel de Machine virtuelle, afin d'installer un environnement sain et spécifique au développement.

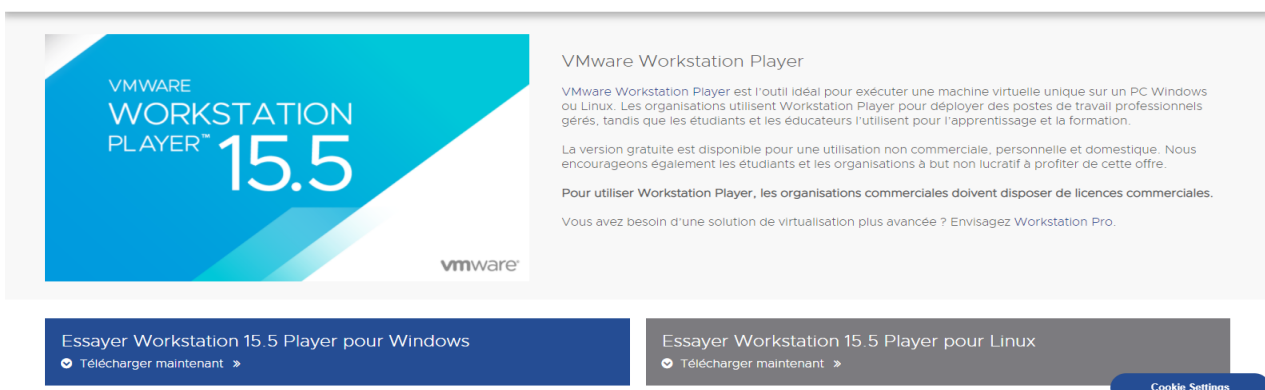
VMware Player

Mon tuteur m'a conseillé de télécharger VMware Player, j'ai donc téléchargé VMware Player.

Je suis donc partie sur le site :

- <https://www.vmware.com/fr/products/workstation-player/workstation-player-evaluation.html>

VMware Workstation 15 Player



VMware Workstation Player

VMware Workstation Player est l'outil idéal pour exécuter une machine virtuelle unique sur un PC Windows ou Linux. Les organisations utilisent Workstation Player pour déployer des postes de travail professionnels gérés, tandis que les étudiants et les éducateurs l'utilisent pour l'apprentissage et la formation.

La version gratuite est disponible pour une utilisation non commerciale, personnelle et domestique. Nous encourageons également les étudiants et les organisations à but non lucratif à profiter de cette offre.

Pour utiliser Workstation Player, les organisations commerciales doivent disposer de licences commerciales.

Vous avez besoin d'une solution de virtualisation plus avancée ? Envisagez Workstation Pro.

Essayer Workstation 15.5 Player pour Windows
Télécharger maintenant »

Essayer Workstation 15.5 Player pour Linux
Télécharger maintenant »

Cookie Settings

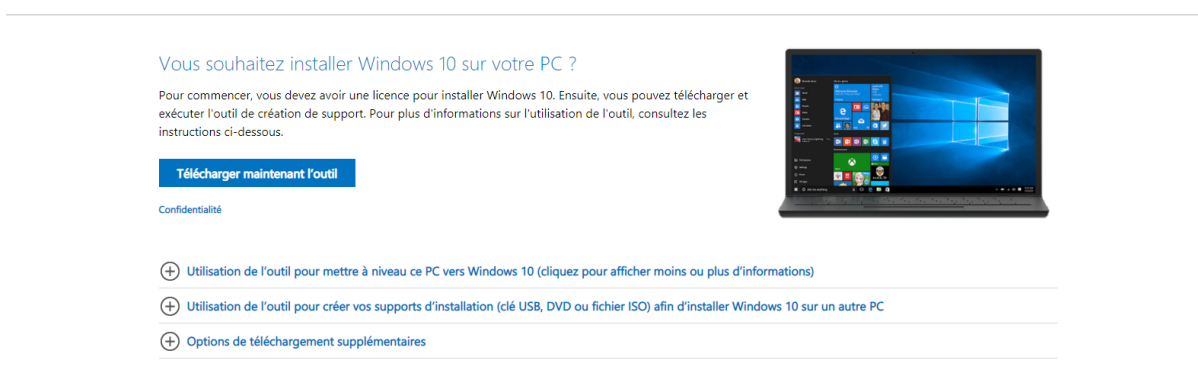
A ce moment j'ai choisie la version window.

Windows 10

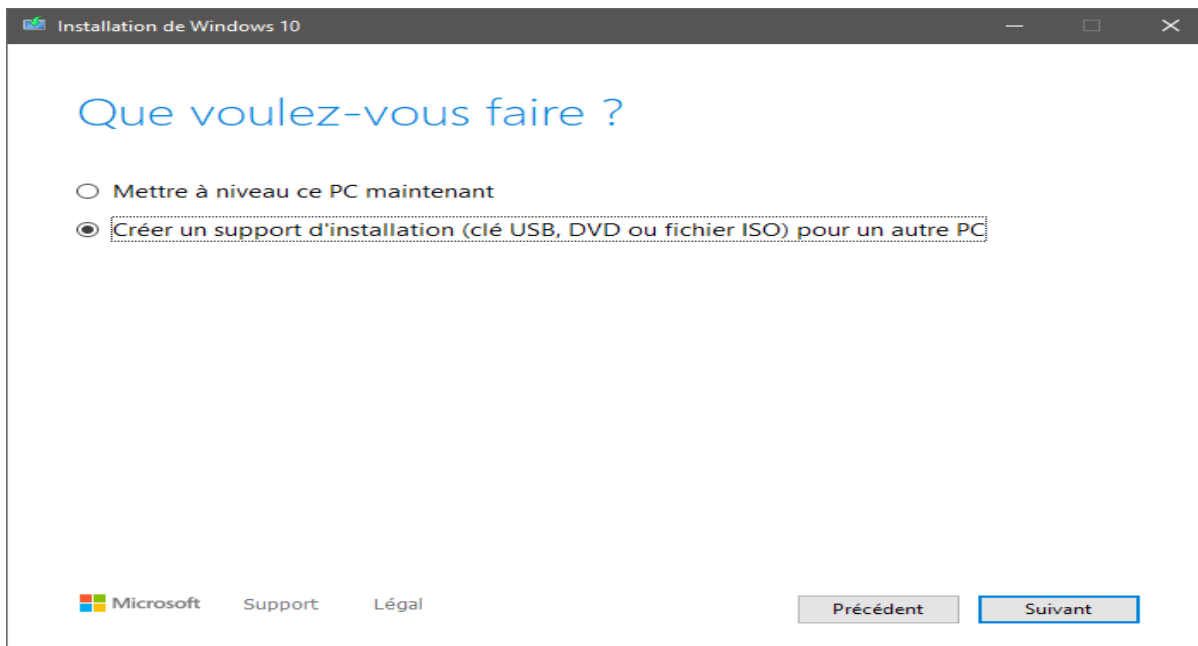
Pour avoir un environnement sain et spécifique au développement, j'ai téléchargé au format iso Windows 10 professionnelle.

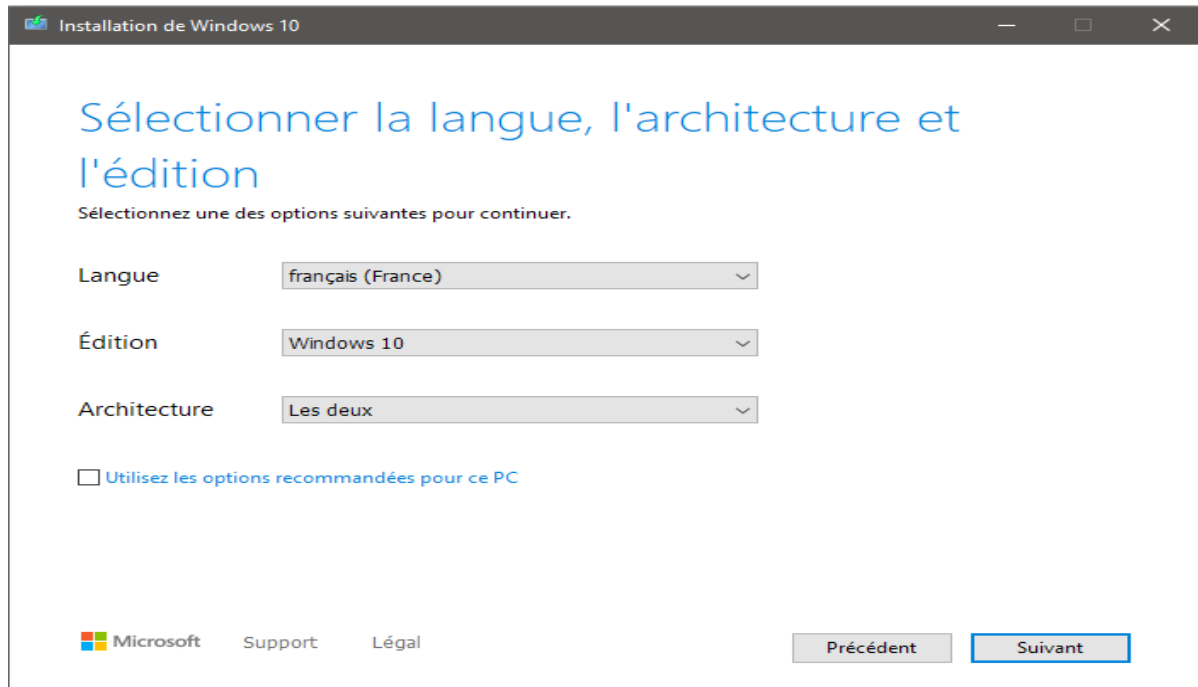
Je suis allé au site officielle de microsoft :

- <https://www.microsoft.com/fr-fr/software-download/windows10>

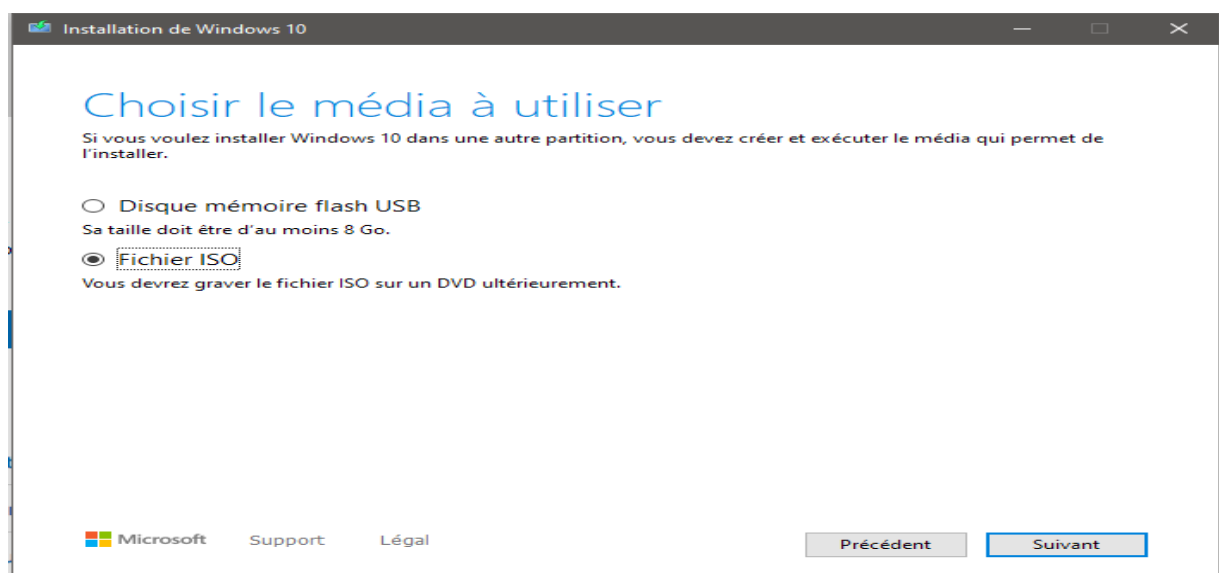


Je suis la procédure d'installation.





Ici je conseille pour se qui ne savent pas quelle et l'architecture de son pc, de sélectionner les deux plutôt que x64 ou x86.

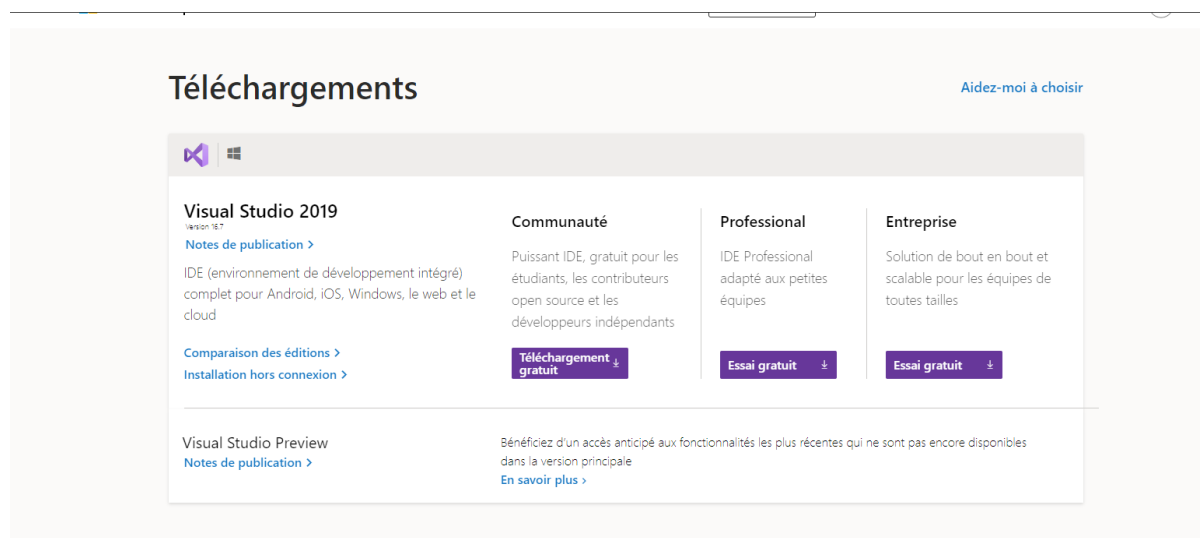


Pour finir je sélectionne le format iso et je lance l'installation.

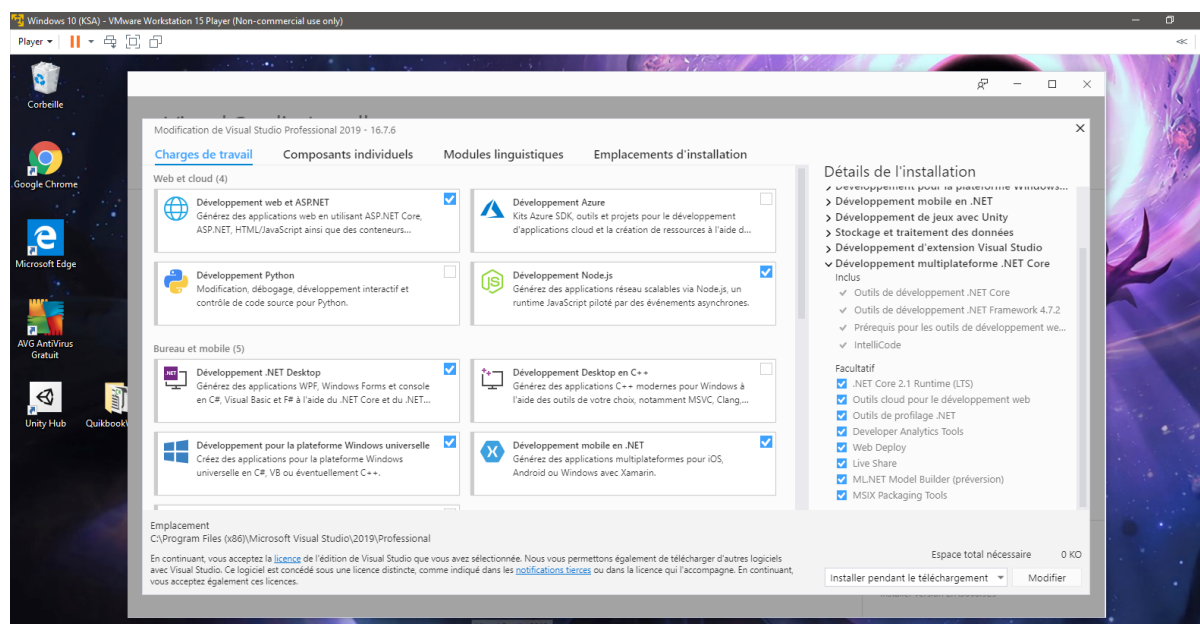
Visual studio 2019

Pour avoir un IDE des plus performant j'ai télécharger visual studio 2019 ici :
- <https://visualstudio.microsoft.com/fr/downloads/>

Ici j'ai choisie la version essaie professionnall et j'ai suivie les étape d'installation.



Ici je vais choisir les language les mieux adapter a mon projet et futur projets.



3.3.2 Développement de l'application

Mon maître de stage m'a demandé de réaliser en premier temps une Application bureau, qui permet la conversion de fichier Xml en Csv.

N'ayant pas de cahier des charges à proprement parlé écrit et ayant été sensibilisée dans le cadre de ma formation à l'AFIP aux avantages d'une approche de type Agile, nous avons régulièrement fait des réunions avec mon maître de stage pour valider chaque étape.

Suite à cette demande, j'ai conçu une maquette, un prototype de l'application avec Adobe XD et un diagramme de Gant que mon maître de stage a validé.

1. Diagramme de GANTT : Référer Page 11.

2. Maquette avec Adobe XD : Référer Page 19.

Pour coder cette application, j'ai utilisé .NET Framework 4.7.2 que j'ai appris Grâce à différents tuto et mon maître de stage M. Franck Deport.

Grâce à mon maître de stage M. Franck Deport, qui m'a aidé à comprendre le Debug de Visual Studio, j'ai pu à la moindre erreur rencontrée utiliser le système de point d'arrêt, d'espion express et de debug pas à pas afin de connaître le nom de l'erreur et la ligne concernée et lorsque je n'arrivais pas à déboguer soumettre mon erreur à « mon ami » Google qui me dirigeait vers des forums tels que developpez.net, la documentation microsoft ou stackoverflow.com qui m'ont souvent été d'un grand secours.

En premier lieu j'ai dû récupérer un exemple de fichier Kml afin d'analyser l'arborescence de celui ci.

1. Exemple fichier kml.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/kml/ext/2.2" xmlns:kml="http://www.opengis.net/kml/2.2" xmlns:atom="http://www.w3.org/2005/Atom">
3 <Folder>
4 <name>Offre en produits fermiers de l'aspos:Ain (créé par la Chambre d'aspos:Agriculture de l'aspos:Ain)</name>
5 <open1/open>
6 <Placemark>
7 <name>Gaeo P2MN</name>
8 <address>228 rue de l'aspos:eglise 01560 Courtes</address>
9 <description>[CDATA[<N>: <br>Catégorie: Produits laitiers<br>Nom-Prénom: Orange Mathieu<br>Adresse-mail: Contact@lesdelicesbressans.fr<br>Numéro de téléphone: 687472727<br>Adresse postale: 228 rue de l'eglise 01560 Courtes<br>Produits-proposés: Yaourts-natures, yaourts-aromatisés (citron, vanille, fraise, poire), yaourts-sur-lit-de-confiture (framboise, myrtille, figue-rouge) et-bouteille-1l-de-lait-pasteurisé.<br>Commande-préalable: Oui<br>Conditions-de-retrait: Colis-À-venir-chercher-au-point-de-vente-à-la-ferme<br>Conditions-de-livraison: <br>Ouverture-du-point-de-vente: Vendredi-de-10h-à-12h-et-de-14h-à-18h<br>Moyens-de-paiements-acceptés: Liquide, Chèque<br>Lien-internet: lesdelicesbressans-ou-www.lesdelicesbressans.fr]]</description>
10 <styleUrl>#icon-1899-1A2378</styleUrl>
11 <ExtendedData>
12 <Data name="N">
13 <value>1.0</value>
14 </Data>
15 <Data name="Catégorie">
16 <value>Produits laitiers</value>
17 </Data>
18 <Data name="Nom-Prénom">
19 <value>Orange Mathieu</value>
20 </Data>
21 <Data name="Adresse-mail">
22 <value>Contact@lesdelicesbressans.fr</value>
23 </Data>
24 <Data name="Numéro de téléphone">
25 <value>687472727</value>
26 </Data>
27 <Data name="Adresse postale">
28 <value>228 rue de l'aspos:eglise 01560 Courtes</value>
29 </Data>
30 <Data name="Produits-proposés">
31 <value>Yaourts-natures, yaourts-aromatisés (citron, vanille, fraise, poire), yaourts-sur-lit-de-confiture (framboise, myrtille, figue-rouge) et-bouteille-1l-de-lait-pasteurisé.</value>
32 </Data>
33 <Data name="Commande-préalable">
34 <value>Oui</value>
35 </Data>
36 <Data name="Conditions-de-retrait">
37 <value>Colis-À-venir-chercher-au-point-de-vente-à-la-ferme</value>
38 </Data>
39 <Data name="Conditions-de-livraison">
40 <value><br>Ouverture-du-point-de-vente</value>
41 </Data>
42 <Data name="Ouverture-du-point-de-vente">
43 <value>Vendredi-de-10h-à-12h-et-de-14h-à-18h</value>
44 </Data>
45 <Data name="Moyens-de-paiements-acceptés">
46 <value>Liquide, Chèque</value>
47 </Data>
48 <Data name="Lien-internet">
49 <value>lesdelicesbressans-ou-www.lesdelicesbressans.fr</value>
50 </Data>
51 </ExtendedData>
52 <Point>
53 <coordinates>5.0827652,46.4593191,0</coordinates>
54 </Point>
55 </Placemark>

```

Ensuite j'ai créé une méthode que j'ai appelé `buttonOuvrirFichier_Click()`, que j'ai ensuite splitté et commenté afin que le code soit compréhensible pour éventuellement les futurs collègues qui passeront dessus.

```

54
55
56
57
58
59
60
61
62
63
64

```

```

/// <summary>
/// Open File From the button
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
1 référence
private void buttonOuvrirFichier_Click(object sender, EventArgs e)
{
    OpenXmlFile();
    DisplayFileXMLInDataGridview();
}

```

J'ai donc créé une méthode `OpenXmlFile()` qui fait appel à la classe `OpenFileDialog`, qui va afficher une boîte de dialogue standard qui invite l'utilisateur à ouvrir un fichier, tout en englobant par l'instruction `using` qui va permettre l'optimisation de la mémoire.

```
75  /// <summary>  
76  /// Open File Dialog  
77  /// </summary>  
78  private void OpenXmlFile()  
79  {  
80  // open file dialog  
81  using (OpenFileDialog open = new OpenFileDialog())  
82  {  
83  // filters  
84  open.Title = "selectionner un fichier";  
85  open.Filter = "Tout Les Fichiers (*.*)|*.*|Fichier XML(*.xml)|*.xml|Fichier KML(*.kml)|*.kml";  
86  if (open.ShowDialog() == DialogResult.OK)  
87  {  
88  // xml file path  
89  textBoxCheminFichierXml.Text = open.FileName;  
90  }  
91  }  
92  }  
93  }
```

`Open.filter` : permet le filtrage du format de texte.

`if (open.ShowDialog() == DialogResult.OK){}` permet d'insérer une condition qui vérifie si l'utilisateur a bien sélectionné un fichier.

`Open.fileName` : permet de récupérer le chemin du fichier sélectionné par l'utilisateur.

Ensuite une méthode `DisplayFileXMLInDataGridView()` qui fait appel à la classe `XNamespace` et `XDocument`.

`XNamespace` : classe me permettant d'introduire un espace de nom dans une variable.

`XDocument.load()` : classe me permettant de créer un document à partir d'un fichier.

```
94  /// <summary>
95  /// Display file XML in DataGridView
96  /// </summary>
97  2 références
98  private void DisplayFileXMLInDataGridView()
99  {
100     var fileOpen = textBoxCheminFichierXml.Text.Trim();
101     if (fileOpen != "")
102     {
103         XNamespace ns = "http://www.opengis.net/kml/2.2";
104         var lDocument = XDocument.Load(@fileOpen);
105         var placemarks = lDocument.Descendants(ns + "Placemark").ToList();
106
107         foreach (var placemark in placemarks)
108         {
109             var masterName = placemark.Element(ns + "name").Value;
110             var adress = placemark.Element(ns + "address").Value;
111             var newRow = AddDataTableRow(dataTable, masterName, adress);
112
113             var extendedDatas = placemark.Descendants(ns + "ExtendedData");
114             foreach (var extendedData in extendedDatas)
115             {
116                 var datas = extendedData.Descendants(ns + "Data");
117                 foreach (var data in datas)
118                 {
119                     var dataName = data.Attribute("name").Value;
120                     var dataValue = data.Value;
121                     FillDataTableRow(dataTable, newRow, dataName, dataValue);
122                 }
123             }
124         }
125         dataGridView1.DataSource = dataTable;
126     }
```

Je remarque donc en analysant mon fichier kml précédemment présenter qu'il y a deux balise, la balise placemark qui englobe le tout et la balise extendData qui elle ensuite englobe une partie des donnée.

J'ai donc en premier créer une liste de mes placemark et ensuite utilisé la méthode foreach() afin de bouclée sur ma liste placemark et de récupérer les première données.

```
106  foreach (var placemark in placemarks)
107  {
108     var masterName = placemark.Element(ns + "name").Value;
109     var adress = placemark.Element(ns + "address").Value;
110     var newRow = AddDataTableRow(dataTable, masterName, adress);
111 }
```

Je crée donc une nouvelle méthode `AddDataTableRow()` (afin que mon code soit plus lisible et facilement compréhensible par un autre développeur) avec 3 paramètres ma `DataTable` et les données récupérer au préalable dans ma placemark afin de pouvoir les intégrer dans ma `DataTable` et gérer les retour à la ligne.

```
128  /// <summary>
129  /// add row entreprise and adress in datatable
130  /// </summary>
131  /// <param name="dataTable"></param>
132  /// <param name="masterName"></param>
133  /// <param name="adress"></param>
134  1 référence
135  private DataRow AddDataTableRow(DataTable dataTable, string masterName, string adress)
136  {
137      DataRow row = dataTable.NewRow();
138      if (!dataTable.Columns.Contains("Nom Entreprise") ||
139          !dataTable.Columns.Contains("Adresse"))
140      {
141          dataTable.Columns.Add("Nom Entreprise");
142          dataTable.Columns.Add("Adresse");
143      }
144      row[0] = masterName;
145      row[1] = adress;
146      dataTable.Rows.Add(row);
147      return row;
148  }
```

Je fais appel à la classe `DataRow` qui représente une ligne dans ma `DataTable` ensuite j'insère une `if()` condition afin d'insérer mes noms de colonne que si elle n'existe pas et grâce à `row[0]` et `row[1]` je spécifie dans quel colonne ou vont mes données.

Après vérification de la réception des donnée dans ma DataTable, je retourne faire la même procédure dans ma méthode DisplayFileXMLInDataGridView() pour les ExtendData et crée une autre méthode FillDataTableRow() avec en paramètres DataTable , DataRow , dataName et dataValue.

```
150      /// Add row in datatable
151      /// </summary>
152      /// <param name="dataTable"></param>
153      /// <param name="masterName"></param>
154      /// <param name="adress"></param>
155      /// <param name="dataName"></param>
156      /// <param name="dataValue"></param>
157      1 référence
158      private void FillDataTableRow(DataTable dataTable, DataRow datarow, string dataName, string dataValue)
159      {
160          if (!dataTable.Columns.Contains(dataName))
161          {
162              dataTable.Columns.Add(dataName);
163          }
164          datarow[dataName] = dataValue.Replace("\n", " ");
165      }
```

Ici j'insère une if() condition afin de vérifier si dans ma DataTable si mais DataName (nom de colonne) existe déjà, si elle n'existe pas je les insère. Ensuite pour chaque DataName j'insère sa value et je remplace les espace par un saut à la ligne.

Voici un exemple de résultat obtenue dans ma DataTable.



Grace à se visuelle graphique l'utilisateur à déjà un aperçue de se que va donnée son fichier csv.

Arriver à se stade la il me rester plus qu'à créer un bouton SaveFile() auquel je voulais convertir les donnée de ma DataTable et les insérez dans un fichier Csv.

J'ai donc splitter mon code et j'ai créer la méthode ConvertTooCsv() avec en paramètre ma DataTable.

```

34  /// <summary>
35  /// Save File from the menu
36  /// </summary>
37  /// <param name="sender"></param>
38  /// <param name="e"></param>
39  1 référence
40  private void enregistrezFichierToolStripMenuItem_Click(object sender, EventArgs e)
41  {
42      ConvertTooCsv(dataTable);
43  }

```

```
166 // <summary>
167 // Convert dataTable to Csv
168 // Save File Csv
169 // </summary>
170 private void ConvertToCsv(DataTable dataTable)
171 {
172     OpenLocationCsvFile();
173     var saveFile = textBoxCheminFichierXml.Text.Trim();
174
175     if (!String.IsNullOrEmpty(saveFile))
176     {
177         var result = new StringBuilder();
178         for (int i = 0; i < dataTable.Columns.Count; i++)
179         {
180             result.Append(dataTable.Columns[i].ColumnName);
181             result.Append(i == dataTable.Columns.Count - 1 ? "\n" : ";");
182         }
183
184         foreach (DataRow row in dataTable.Rows)
185         {
186             for (int i = 0; i < dataTable.Columns.Count; i++)
187             {
188                 result.Append(row[i].ToString());
189                 result.Append(i == dataTable.Columns.Count - 1 ? "\n" : "; ");
190             }
191         }
192         File.WriteAllText(saveFile, result.ToString(), Encoding.GetEncoding(1252));
193     }
194 }
```

Ici j'insère un `if()` condition qui va me permettre de gérer les espaces ou si l'utilisateur veut enregistrer sans avoir sélectionné un fichier. Ensuite je fais appel à la classe `StringBuilder()` qui permet de développer le nombre de caractères contenus dans la chaîne qu'il encapsule et j'insère mes données via une boucle `for` en spécifiant `Encoding` afin que les données soient lisibles.

4 - Conclusion

Pour conclure, je dirais que j'ai vraiment apprécié de mettre en pratique mes compétences dans un cadre professionnel. J'ai bien conscience que cette expérience ne couvre pas toutes celles demandées par le titre professionnel, mais elle m'a toutefois permis d'avoir un peu plus confiance en moi et d'appréhender ma recherche d'emploi dans le développement avec plus de sérénité.