

# DOCUMENT TECHNIQUE

[Adresse de la société]

## ***SOMMAIRE***

<b><i>I°) INTRODUCTION</i></b>	<b><i>2</i></b>
<b><i>II°) LE LANGAGE</i></b>	<b><i>2</i></b>
<b><i>III°) L'INTERFACE D'UTILASTION</i></b>	<b><i>3</i></b>
<b><i>IV °) LES FONCTIONS IMPORTANTES</i></b>	<b><i>6</i></b>
<b><i>V °) LES BASES DE DONNEES</i></b>	<b><i>10</i></b>
<b><i>VI °) LA SECURITE</i></b>	<b><i>11</i></b>
<b><i>VII°) LES LIBRAIRIES UTILISEES</i></b>	<b><i>13</i></b>

## I°) INTRODUCTION

Ce document est destiné à montrer les différentes solutions choisies, plans et motivations pour la création de l'application. Il va du langage utilisé aux librairies incorporées.

## II°) LE LANGAGE

Le langage utilisé pour concevoir l'application est le langage de programmation C#.

C# est un langage de programmation moderne, orienté objet et de type sécurisé. Il permet aux développeurs de créer de nombreux types d'applications sécurisées et fiables qui s'exécutent dans .NET. L'utilisation des fonctionnalités Windows Forms sont utilisées pour la création de l'application.

Windows Forms) est une plate-forme de création d'interfaces graphiques créée par Microsoft. Elle est adossée au Framework .NET, cette technologie suit le paradigme de programmation événementielle : une application Windows Forms est pilotée par des événements auxquels elle réagit.

Il réagit à des événements provenant du système ou de l'utilisateur. L'ordre d'exécution des instructions n'est donc plus prévu. C'est l'utilisateur qui a le contrôle

```
// PROGRAMME
Main()
{
    ...
    while(true) // tantque Mamie s'active
    {
        // récupérer son action (faire une maille ...)
        e = getNextEvent();
        // traiter son action (agrandir le tricot ...)
        processEvent();
    }
    ...
}
```



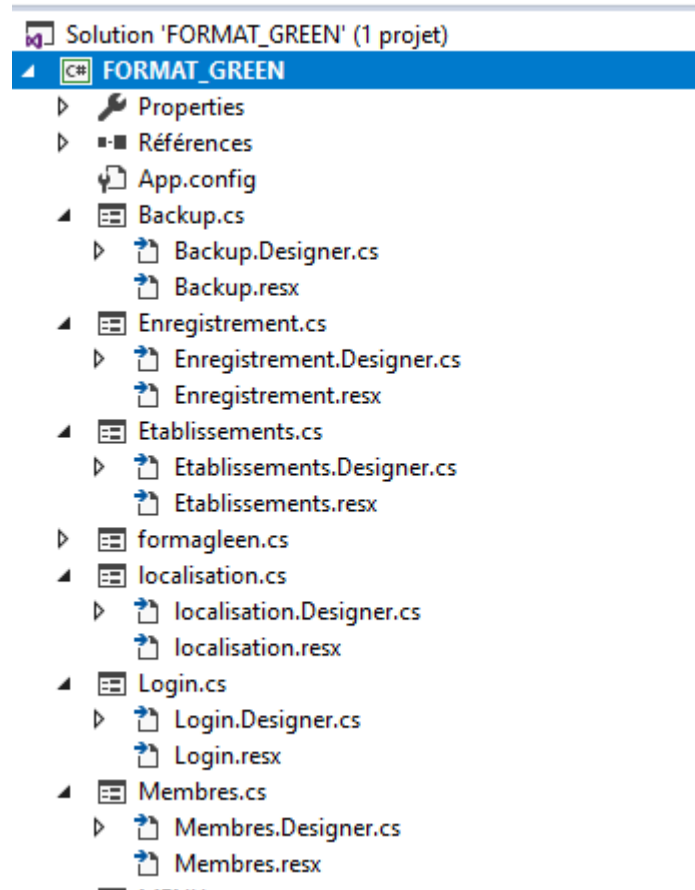
La programmation événementielle s'oppose par conséquent à la programmation séquentielle. Elle est notamment utilisée pour gérer des interactions riches avec l'utilisateur, comme celles des interfaces graphiques homme-machine (GUI, Graphical User Interface)

La particularité de Windows Forms se manifeste dans sa maniabilité à structurer et créer des interfaces au goût de son utilisateur sans rencontrer d'imprévu.

### III°) L'INTERFACE D'UTILASTION

Une application Windows Forms ou WinForms est structurée autour d'un ou plusieurs formulaires, appelés Forms.

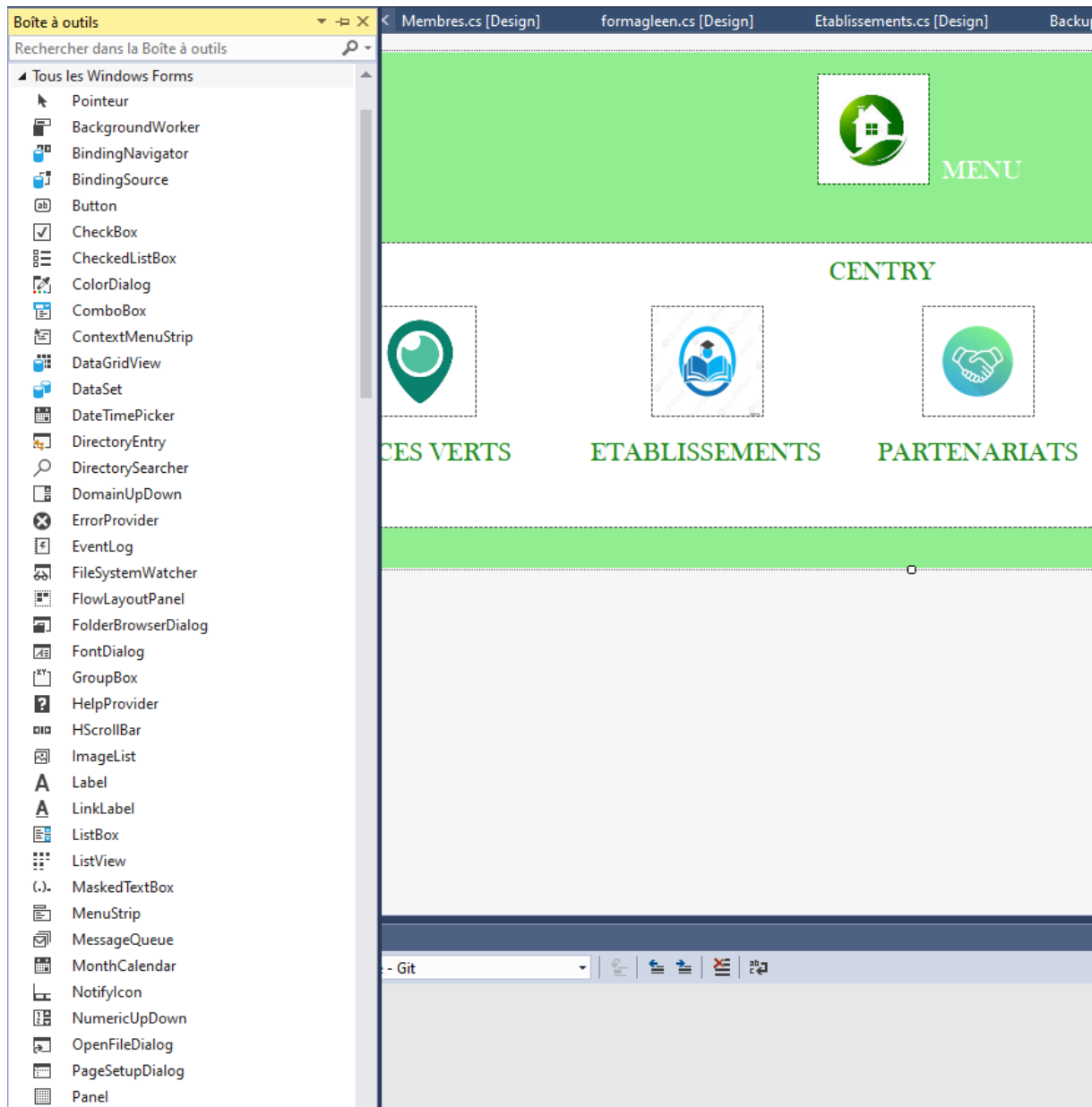
Lorsqu'on crée une nouvelle application WinForms, l'IDE Visual Studio génère automatiquement plusieurs éléments qu'il est important d'identifier.



Chaque formulaire WinForms est décrit par deux fichiers :

- Un fichier.Designer.cs qui contient le code généré automatiquement par l'IDE lors de la conception graphique du formulaire ;
- Un fichier .cs qui contient le code C# écrit par le développeur pour faire réagir le formulaire aux événements qui se produisent. Ce fichier est appelé code Bhind

Un double-clic sur un composant CS [design] dans l'arborescence déclenche l'apparition du concepteur de formulaire. Cette interface va permettre d'éditer l'apparence du formulaire



L'édition du formulaire se fait en y glissant ou déposant des *contrôles*, rassemblés dans une boîte à outils (liste de gauche). De nombreux contrôles sont disponibles pour répondre à des besoins variés et construire des IHM riches et fonctionnelles. Parmi les plus utilisés, on peut citer :

- Label qui affiche un simple texte
- TextBox qui crée une zone de saisie de texte
- Button qui affiche un bouton
- ListBox qui regroupe une liste de valeurs

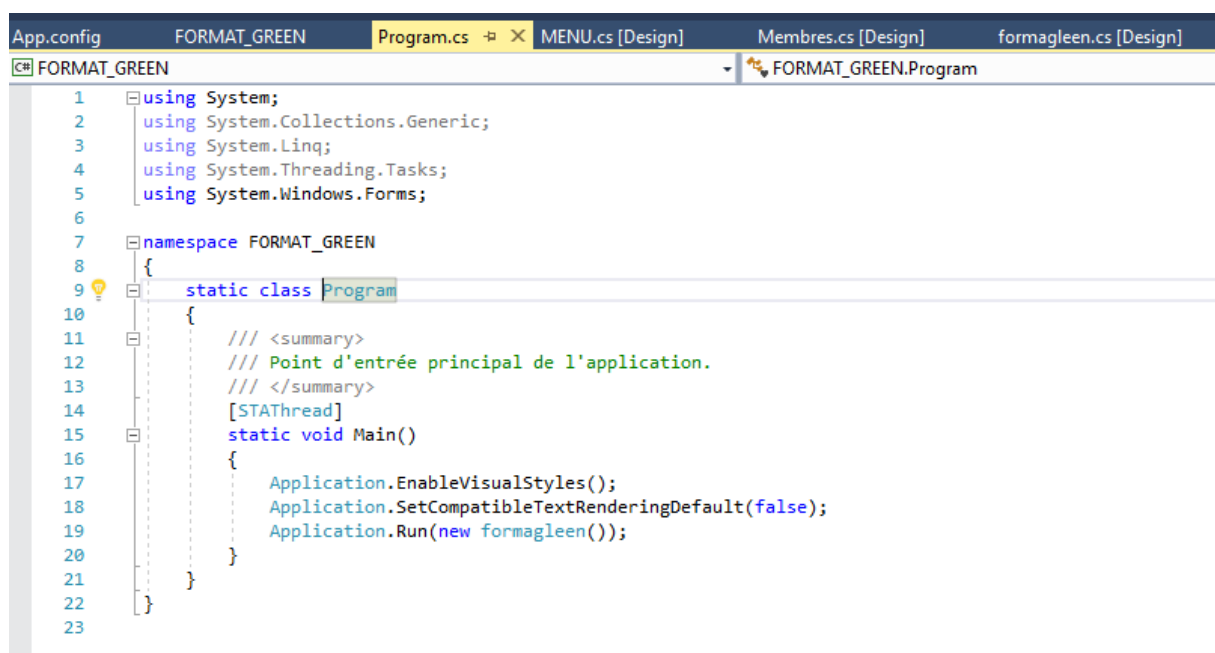
- CheckBox qui affiche une case à cocher

Pour l'application les formulaires seront constitués d'une interface qui servira de menu général qui redirigera vers des sous menus en cliquant soit sur des icones ou des mots.

Chaque sous-menus auront un rôles précis que nous verrons plus bas

## IV°) LES FONCTIONS IMPORTANTES

En double-cliquant sur un contrôle dans le concepteur de formulaire, on ajoute un gestionnaire d'évènements pour l'évènement par défaut associé au contrôle. On arrive alors à l'endroit de création de fonctions. Ces fonctions permettent l'exécution de programmes qui font fonctionner l'ensemble de l'application.



```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using System.Windows.Forms;
6
7  namespace FORMAT_GREEN
8  {
9      static class Program
10     {
11         /// <summary>
12         /// Point d'entrée principal de l'application.
13         /// </summary>
14         [STAThread]
15         static void Main()
16         {
17             Application.EnableVisualStyles();
18             Application.SetCompatibleTextRenderingDefault(false);
19             Application.Run(new formagleen());
20         }
21     }
22 }
23

```

Les fonctions les plus utilisées pour ce projet sont basées sur le modèle CRUD pour create, read, update, delete. Elle donne la possibilité à l'utilisateur de créer, lire, modifier, voir même de supprimer des éléments.

```
private void btnifuThinButton22_Click(object sender, EventArgs e)
{
    if (Id.Text == "")
    {
        MessageBox.Show("ajouter le di du membre");
    }
    else
    {
        try
        {
            Con.Open();
            string query = "delete from EtablissementDb where Id='" + Id.Text + "'";
            SqlCommand cmd = new SqlCommand(query, Con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Suppression reussi");
            Con.Close();
            pop();
        }
        catch (Exception Ex)
        {
            MessageBox.Show(Ex.Message);
        }
    }
}

private void btnifuThinButton21_Click(object sender, EventArgs e)
{
    if (Id.Text == "" || Nom.Text == "" || Representé.Text == "" || Adresse.Text == "")
    {
        MessageBox.Show("tout les champs ne sont pas saisis");
    }
    else
    {
        try
        {
            Con.Open();
            string query = "update EtablissementDb set Nom='" + Nom.Text + "',Adresse='" + Adresse.Text + "',Type='" + Type.SelectedItem.ToString() + "',Representé='" + Representé.Text + "'";
            SqlCommand cmd = new SqlCommand(query, Con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("données modifiées ");
            Con.Close();
        }
        catch (Exception Ex)
        {
            MessageBox.Show(Ex.Message);
        }
    }
}
```

Des éléments importants du projet ensuite viennent les fonctions comme l'intégration de module web pour la création d'un programme de localisation afin d'avoir la carte,



```

private void Cherche_Click(object sender, EventArgs e)
{
    string rue = Rue.Text;
    string ville = Ville.Text;
    try
    {
        StringBuilder queryadress = new StringBuilder();
        queryadress.Append("http://google.com/maps?q=");

        if (rue != string.Empty)
        {
            queryadress.Append(rue + "," + "+");
        }
        if (ville != string.Empty)
        {
            queryadress.Append(ville + "," + "+");
        }

        Map.Navigate(queryadress.ToString());
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString(), "Erreur");
    }
}

```

De module pour pouvoir scanner via la caméra de notre ordinateur,

```

private void Go_Click(object sender, EventArgs e)
{
    captureDevice = new VideoCaptureDevice(filterInfoCollection[Came.SelectedIndex].MonikerString);
    captureDevice.NewFrame += captureDevice_NewFrame;
    captureDevice.Start();
    timer1.Start();
}

private void captureDevice_NewFrame(object sender, NewFrameEventArgs eventArgs)
{
    Ecran.Image = (Bitmap)eventArgs.Frame.Clone();
}

private void Scan_FormClosing(object sender, FormClosingEventArgs e)
{
    if (captureDevice.IsRunning)
        captureDevice.Stop();
}

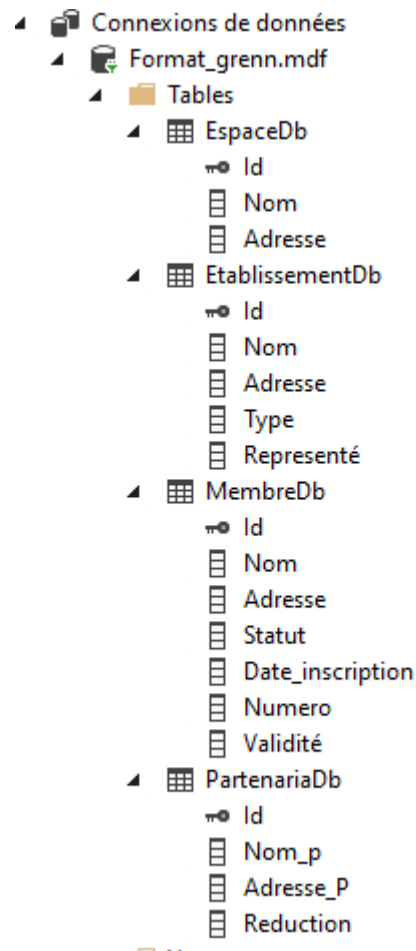
```

Un système de fichier pour que l'application puisse créer des documents

```
private async void boutonThinButton25_Click(object sender, EventArgs e)
{
    using (OpenFileDialog ofd = new OpenFileDialog() { Filter = "Text Documents|*.text", ValidateNames = true, Multiselect = false })
    {
        if (ofd.ShowDialog() == DialogResult.OK)
        {
            using (StreamReader sr = new StreamReader(ofd.FileName))
            {
                rapport.Text = await sr.ReadToEndAsync();
            }
        }
    }
}

private async void ecrire_Click(object sender, EventArgs e)
{
    using (SaveFileDialog sfd = new SaveFileDialog() { Filter = "Text Documents|*.text", ValidateNames = true })
    {
        if (sfd.ShowDialog() == DialogResult.OK)
        {
            using (StreamWriter sw = new StreamWriter(sfd.FileName))
            {
                await sw.WriteLineAsync(rapport.Text);
                MessageBox.Show("informations enregistrées.", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
    }
}
```

Et la mise en place d'un système de base données dans l'application.



## V°) LES BASES DE DONNEES

Le système de base de données intégré à l'application se nomme SQL SEVER. Structured Query Language ou SQL, est un langage informatique permettant d'exploiter des bases de données.

Concrètement, un SQL server est un outil qui possède toutes les caractéristiques pour pouvoir accompagner l'utilisateur dans la manipulation, le contrôle, le tri, la mise à jour, et bien d'autres actions encore, de bases de données grâce au langage SQL. Grâce aux fonctions de SQL que les fonctions CRUD sont réalisable c'est donc le choix idéal pour la composition des données.

L'application est composée d'une base de données nommée comme l'association avec quatre tables qui sont reliées à un sous menu. La connexion se fait grâce à l'utilisation du langage SQL `SqlConnection con = new SqlConnection` intégré dans la programmation des fonctions

```
SqlConnection Con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\warel\Documents\Format_grenn.mdf;Integrated Security=True;Connect Timeout=30");
```

- EspaceDb pour les données des espaces verts
- EtablissementDB pour les données des établissements ou l'association a déjà travaillé
- MembreDb pour les données des membres de l'association
- PartenariatDb pour les données des partenaires de l'association

## VI°) LA SECURITE

Afin d'éviter tout perte de l'application ou des données, elle dispose d'un système d'identification avec mot de passe accessible aux seuls administrateurs

```

namespace FORMAT_GREEN
{
    public partial class Login : Form
    {
        public Login()
        {
            InitializeComponent();
        }

        private void CONNEXION_Click(object sender, EventArgs e)
        {
            if (Idn.Text == "" || Mdp.Text == "")
            {
                MessageBox.Show("entrer un id et un mot de passe");
            }
            else if (Idn.Text == "Admin" && Mdp.Text == "123")
            {
                this.Hide();
                MENU home = new MENU();
                home.Show();
            }
            else
            {
                MessageBox.Show("ERREUR REESSAYER SVP");
            }
        }

        private void pictureBox7_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}

```

Mais également d'un système de sauvegarde et de récupération de données qui contient les bases de données. Ce qui permet à l'association d'éviter de nombreux inconvénients liés à la sécurité informatique,

```

private void Bk_Click_1(object sender, EventArgs e)
{
    Con.Open();
    String database = Con.Database.ToString();
    try
    {
        if (Chemin.Text == string.Empty)
        {
            MessageBox.Show("ERREUR REESSAYER CORRECTEMENT" +
                "");
        }
        else
        {
            string q = "BACKUP DATABASE [" + database + "] TO DISK='" + Chemin.Text + "\\\" + "Database" + "-" + DateTime.Now.ToString("yyyy-MM-dd--HH-mm-ss") + ".bak";
            SqlCommand scmd = new SqlCommand(q, Con);
            scmd.ExecuteNonQuery();

            MessageBox.Show("SAUVEGARDE REUSSE", " success", MessageBoxButtons.OK, MessageBoxIcon.Information);
            Bk.Enabled = false;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally { Con.Close(); }
}

private void Emp_Click(object sender, EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.Filter = "SQL SERVER database backup files|.bak";
    ofd.Title = "Database Restore";
    if (ofd.ShowDialog() == DialogResult.OK)
    {
        Nchemin.Text = ofd.FileName;
        Emp.Enabled = true;
    }
}

```

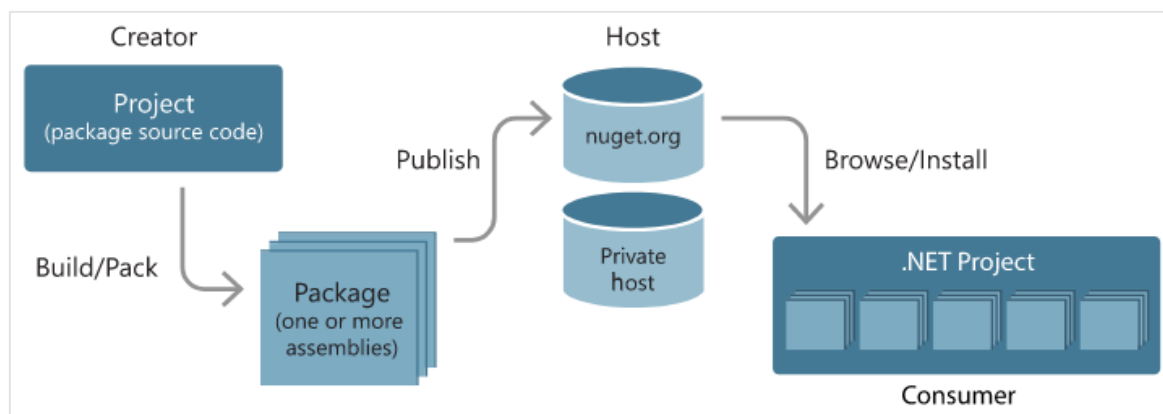
Une sensibilisation sur la sécurité est également conseillée aux utilisateurs de l'application afin d'éviter tout pertes internes.

## VII°) LES PAQUETS UTILISEES


















.net offre la possibilité d'utiliser de nombreux outils qui permettent de concevoir divers applications aux préférences de l'utilisateur. Grâce à NUGET, c'est le gestionnaire de paquets de la plateforme de développement Microsoft .NET. C'est un logiciel libre et open source principalement développé par Microsoft.

Il est distribué sous forme d'extension des environnements de développement Visual Studio et SharpDevelop. Il vise à simplifier le processus d'intégration de bibliothèques externes à une application développée dans .NET en automatisant certaines tâches répétitives : télécharger et installer une bibliothèque, modifier les paramètres de configuration, et répéter la même opération pour les autres bibliothèques auxquelles la première fait appel.

Les créateurs produisent des packages NuGet pratiques et les publient sur un hôte, les consommateurs recherchent des packages pratiques et compatibles sur les hôtes accessibles, les téléchargent et incluent ces packages dans leurs projets. Une fois installés dans un projet, les API des packages sont disponibles pour le reste du code du projet.



Plusieurs paquets ont été utilisés pour rendre l'utilisation du logiciel accessible : des paquets pour gérer la caméra, d'autres pour la gestion des bases de données mais également pour le design et le confort visuel.

	<b>AForge</b> par AForge.NET Core library of the AForge.NET framework	 v2.2.5 
	<b>AForge.Video</b> par AForge.NET AForge.Video library of the AForge.NET framework	 v2.2.5
	<b>AForge.Video.DirectShow</b> par AForge.NET AForge.Video.DirectShow library of the AForge.NET framework	 v2.2.5
	<b>Azure.Core</b> par Microsoft This is the implementation of the Azure Client Pipeline	 v1.6.0 v1.19.0
	<b>Azure.Identity</b> par Microsoft This is the implementation of the Azure SDK Client Library for Azure Identity	 v1.3.0 v1.4.1
	<b>Bunifu.Charts.WinForms</b> par BunifuLabs Create stunning visualizations and responsive charts for your dashboards and data-centric applications.	 v1.1.3 v1.1.5
	<b>Bunifu.UI.WinForms</b> par BunifuLabs Create professional .NET applications with a ready-made suite of beautifully crafted and customizable controls.	 v5.0.9 v5.2.1
	<b>Guna.UI2.WinForms</b> par Sobatdata, Ilham Mutaqin, Russell Chidhakwa Guna UI is the suite for creating groundbreaking desktop app UI. It is for developers targeting the .NET Windows Forms platform. Guna UI guarantees faster development and improved productivity.	 v2.0.2

## ■ ■ Références



Analyseurs

- ■ AForge
- ■ AForge.Video
- ■ AForge.Video.DirectShow
- ■ Azure.Core
- ■ Azure.Identity
- ■ BasselTech CamCapture
- ■ Bunifu.Charts.WinForms
- ■ Bunifu.Licensing
- ■ Bunifu.UI.WinForms.1.5.3
- ■ Bunifu.UI.WinForms.BunifuButton
- ■ Bunifu.UI.WinForms.BunifuCheckBox
- ■ Bunifu.UI.WinForms.BunifuCircleProgress
- ■ Bunifu.UI.WinForms.BunifuColorTransition
- ■ Bunifu.UI.WinForms.BunifuDataGridView
- ■ Bunifu.UI.WinForms.BunifuDatePicker
- ■ Bunifu.UI.WinForms.BunifuDropdown
- ■ Bunifu.UI.WinForms.BunifuFormDock
- ■ Bunifu.UI.WinForms.BunifuGauge
- ■ Bunifu.UI.WinForms.BunifuGradientPanel

- ■ System.IdentityModel
- ■ System.IdentityModel.Tokens.Jwt
- ■ System.Memory
- ■ System.Net
- ■ System.Net.Http
- ■ System.Numerics
- ■ System.Numerics.Vectors
- ■ System.Runtime.CompilerServices.Unsafe
- ■ System.Security
- ■ System.Security.AccessControl
- ■ System.Security.Cryptography.ProtectedData
- ■ System.Security.Permissions
- ■ System.Security.Principal.Windows
- ■ System.ServiceProcess
- ■ System.Text.Encoding.Web
- ■ System.Text.Json
- ■ System.Threading.Tasks.Extensions
- ■ System.Transactions
- ■ System.ValueTuple
- ■ System.Windows.Forms
- ■ System.Xml
- ■ System.Xml.Linq
- ■ WindowsBase
- ■ zxing