

Wassime
El Yousfi
1^{re} année SLAM

Compte rendu

En tant que technicien développeur junior pour l'ESN InfoTech Services 86, j'ai travaillé sur le développement d'une application de bureau qui permet de gérer le personnel, leur affectation à un service et leurs absences, et cela pour différentes médiathèques à Vienne. L'application est écrite en C# et utilise MySQL.

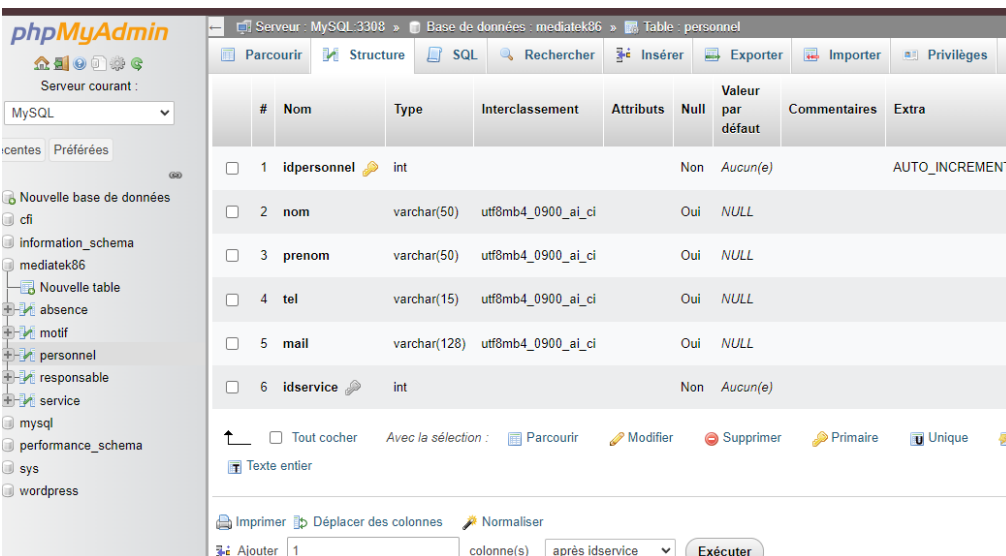
La mission globale était donc de créer une application de bureau qui peut communiquer avec une base de données sql en utilisant l'extension MySQL.Data. Il a fallut permettre l'authentification, avoir des pages pour modifier ou ajouter un personnel ou une absence, et remplir les listView d'après les données de la base de données. Les interfaces visuelles ont donc été créer, sans pour autant chercher à les rendre esthétiques.

Voilà une présentation et explication de chaque étape.

Étape 1 :

L'étape 1 consistait a préparer le développement de l'application de bureau en installant des logiciels comme Wampserver, Visual Studio, etc.. Il a fallut créer la base de données mediatek86 à partir du script SQL fournit, l'alimenter avec des exemples et créer un utilisateur avec les droits d'accès en utilisant phpmyadmin. Phpmyadmin fonctionne sur un serveur local (localhost) grâce au logiciel Wampserver, on peut donc simuler une véritable base de données en ligne sur un serveur.

Phpmyadmin :

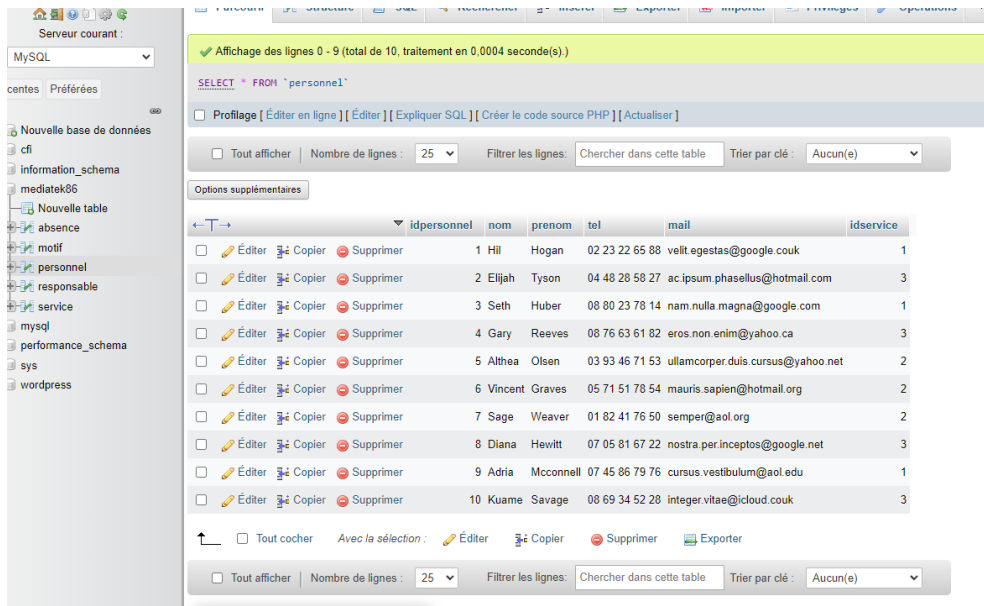


The screenshot shows the phpMyAdmin interface. On the left, the database 'mediatek86' is selected, and the table 'personnel' is highlighted. The main area displays the table structure with the following columns:

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	idpersonnel	int			Non	Aucun(e)		AUTO_INCREMENT
2	nom	varchar(50)	utf8mb4_0900_ai_ci		Oui	NULL		
3	prenom	varchar(50)	utf8mb4_0900_ai_ci		Oui	NULL		
4	tel	varchar(15)	utf8mb4_0900_ai_ci		Oui	NULL		
5	mail	varchar(128)	utf8mb4_0900_ai_ci		Oui	NULL		
6	idservice	int			Non	Aucun(e)		

At the bottom, there are options to 'Ajouter' (Add) a new record, with a dropdown set to 'après idservice' and an 'Exécuter' (Execute) button.

Table personnel alimenté d'exemples en utilisant le site <https://generatedata.com/> :



idpersonnel	nom	prenom	tel	mail	idservice
1	Hil	Hogan	02 23 22 65 88	velit.egestas@google.couk	1
2	Elijah	Tyson	04 48 28 58 27	ac.ipsam.phasellus@hotmail.com	3
3	Seth	Huber	08 80 23 78 14	nam.nulla.magna@google.com	1
4	Gary	Reeves	08 76 63 61 82	eros.non.enim@yahoo.ca	3
5	Althea	Olsen	03 93 46 71 53	ullamcorper.duis.cursus@yahoo.net	2
6	Vincent	Graves	05 71 51 78 54	mauris.sapien@hotmail.org	2
7	Sage	Weaver	01 82 41 76 50	semper@aol.org	2
8	Diana	Hewitt	07 05 81 67 22	nostra.per.inceptos@google.net	3
9	Adria	Mcconnell	07 45 86 79 76	cursus.vestibulum@aol.edu	1
10	Kuame	Savage	08 69 34 52 28	integer.vitae@cloud.couk	3

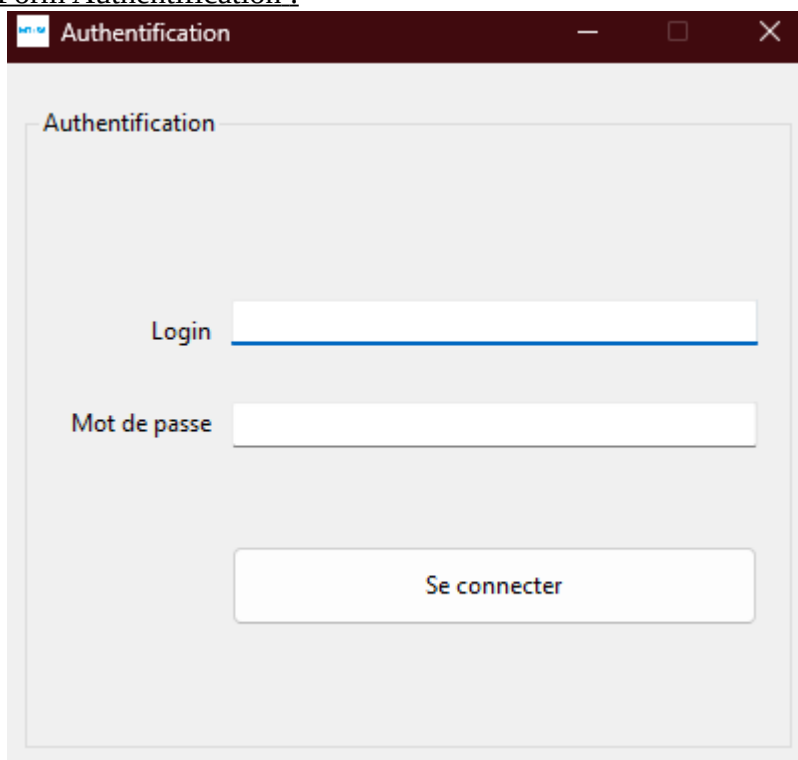
Pour conclure, les objectifs ont été remplis, l'environnement pour développer le logiciel a donc été mis en place correctement, et les exemples créés avec le site <https://generatedata.com/> ont permis de facilement vérifier si le code était fonctionnel.

Étape 2 :

Cette étape était primordiale car il a fallu étudier le dossier documentaire. En effet, le dossier documentaire définit en quelque sorte le logiciel dans sa globalité, ne pas l'étudier ferait en sorte que le logiciel ne respecterait pas correctement les demandes du client. En utilisant Pencil j'ai pu me faire une idée des interfaces du logiciel sans coder quoi que ce soit. J'ai ensuite créé les packages pour respecter le schéma Modèle-Vue-Contrôleur, j'ai donc créé différents dossiers dans Visual Studio pour accueillir les classes correspondantes. J'ai ensuite fait une sauvegarde avec Github et créé sur ce même site un projet pour pouvoir suivre mon avancé sur le logiciel.

J'ai finalement créé le visuel des interfaces dans l'IDE Visual Studio.

Form Authentification :



Authentification

Login

Mot de passe

Form Gestion :

Gestion personnel

Gestion

id	nom	prenom	tel	mail	service
1	Hil	Hogan	02 23 22 65 88	velit.egestas...	administratif
2	Elijah	Tyson	04 48 28 58 27	ac.ipsun.ph...	prêt
3	Seth	Huber	08 80 23 78 14	nam.nulla...	administratif
4	Gary	Reeves	08 76 63 61 82	eros.non.eni...	prêt
5	Althea	Olsen	03 93 46 71 53	ullamcorper...	médiation c...
6	Vincent	Graves	05 71 51 78 54	mauris.sapi...	médiation c...
7	Sage	Weaver	01 82 41 76 50	semper@ao...	médiation c...
8	Diana	Hewitt	07 05 81 67 22	nostra.per.i...	prêt
9	Adria	Mcconnell	07 45 86 79 76	cursus.vesti...	administratif
10	Kua	Savage	08 69 34 52 11	integer.vitae...	prêt

id	datedebut	datefin	motif
1	19/06/2023 07:59:42	16/01/2025 22:50:19	congé parental

Modifier

Supprimer

Actualiser

Modifier

Supprimer

Ajouter Personnel

Ajouter Absence

L'un des dernier commit sur gitHub :

WassimeEY / Atelier2-MedTk86

Q Type to search

Code Issues Pull requests Actions Projects Security Insights Settings

Commit

Atelier 2 complet, il me reste à générer la documentation technique en...

WassimeEY committed 7 hours ago

Showing 15 changed files with 1253 additions and 99 deletions.

Controleur

FormAuthController.cs

FormGestionController.cs

MedTk86.csproj

Modele

Absence.cs

Motif.cs

Personnel.cs

Service.cs

Program.cs

Vue

FormAuthentication.Des...

FormAuthentication.cs

FormGestion.Designer.cs

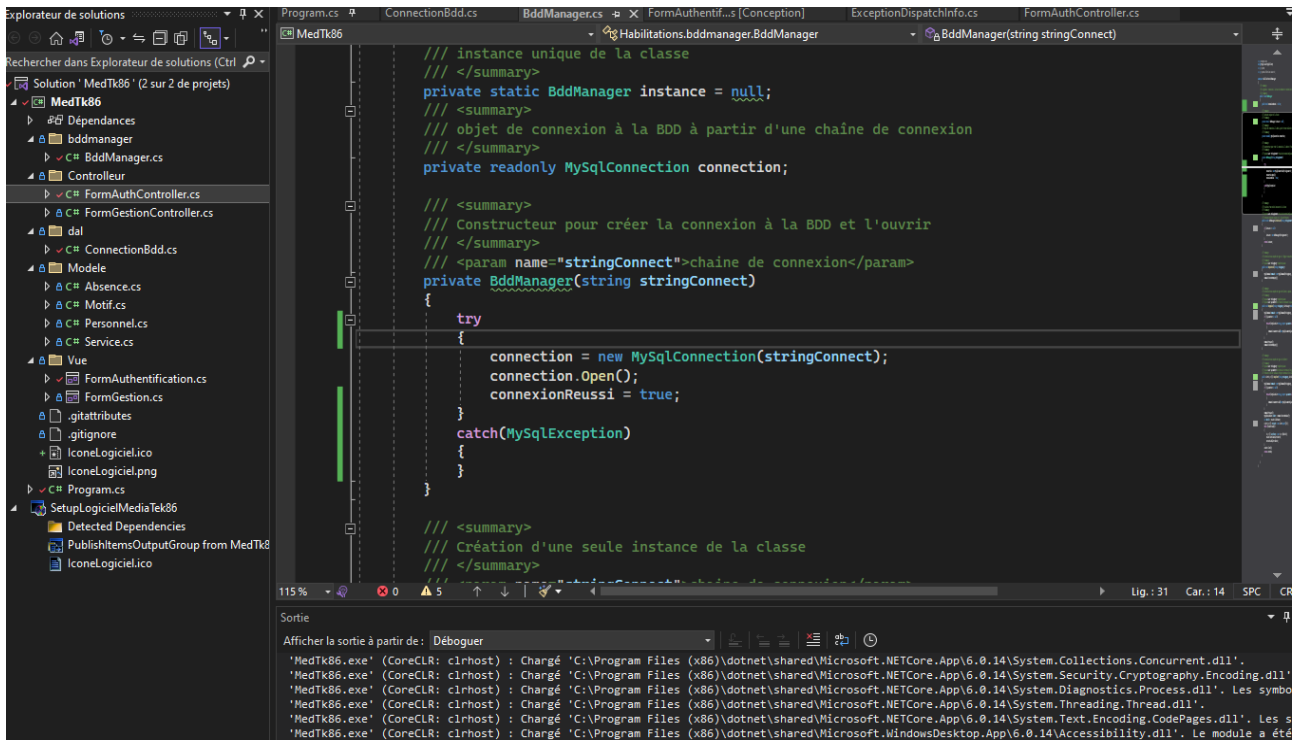
```
1 using Habilitations.bddmanager;
2 using MedTk86.dal;
3 using MedTk86.Vue;
4 using System;
5 using System.Collections.Generic;
6 using System.Linq;
7 using System.Runtime.CompilerServices;
8 using System.Text;
9 using System.Threading.Tasks;
10
11 namespace MedTk86.Controleur
12 {
13     /// <summary>
14     /// Singleton : controleur de la form Authentication
15     /// </summary>
16     public class FormAuthController
17     {
18         private static FormAuthController instance = null;
19         private BddManager connection = ConnectionBdd.GetConnectionBdd();
20         private static FormAuthentication formAuthblef;
21     }
```

Pour conclure, la partie visuelle avait été faite, le projet était sauvegardé sur gitHub, utilisé pencil a permit d'avoir une idée avant de créer les interfaces, ce qui ma fait gagner du temps. La partie Vue de l'application maintenant faite, je devais passer à l'étape 3.

Étape 3 :

J'ai insérer la classe bddManager dans le projet, cette classe ma permit de communiquer avec la base de données grace aux fonctions incluse dedans. J'ai aussi créer les classes métiers correspondants aux tables de la base de données. Je me suis aussi occupé de créer les commentaires et la documentation technique grace à SandCastle, j'ai choisi de le faire avec un fichier .chm .

Structure de la solution et le constructeur de BddManager où on peut voir la connexion :



Procédure ConnecterAvecBdd :

```
static public void ConnecterAvecBdd()
{
    string chaineConnexion =
    "server=127.0.0.1;uid=gerant;pwd=gerantDeLaBDD;database=mediatek86;port=3306";
    connectionBddManager = BddManager.GetInstance(chaineConnexion);
}
```

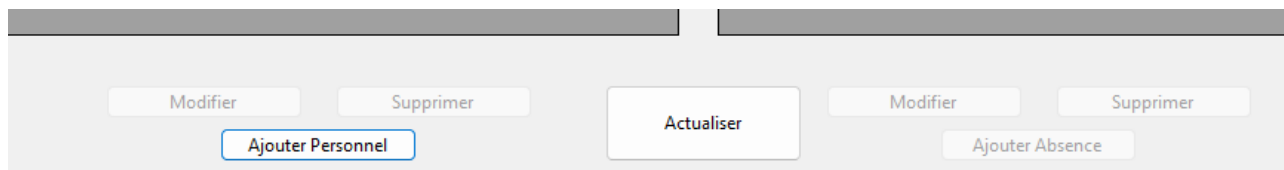
Pour conclure, j'ai ici, utiliser la classe bddManager pour pouvoir me connecter avec la base de données, le package dal a contenu la classe ConnectionBdd qui agit comme un pont entre les controleurs et la classe BddManager. La documentation technique a permit de déjà voir plus clair dans le logiciel jusqu'à présent.

Étape 4 :

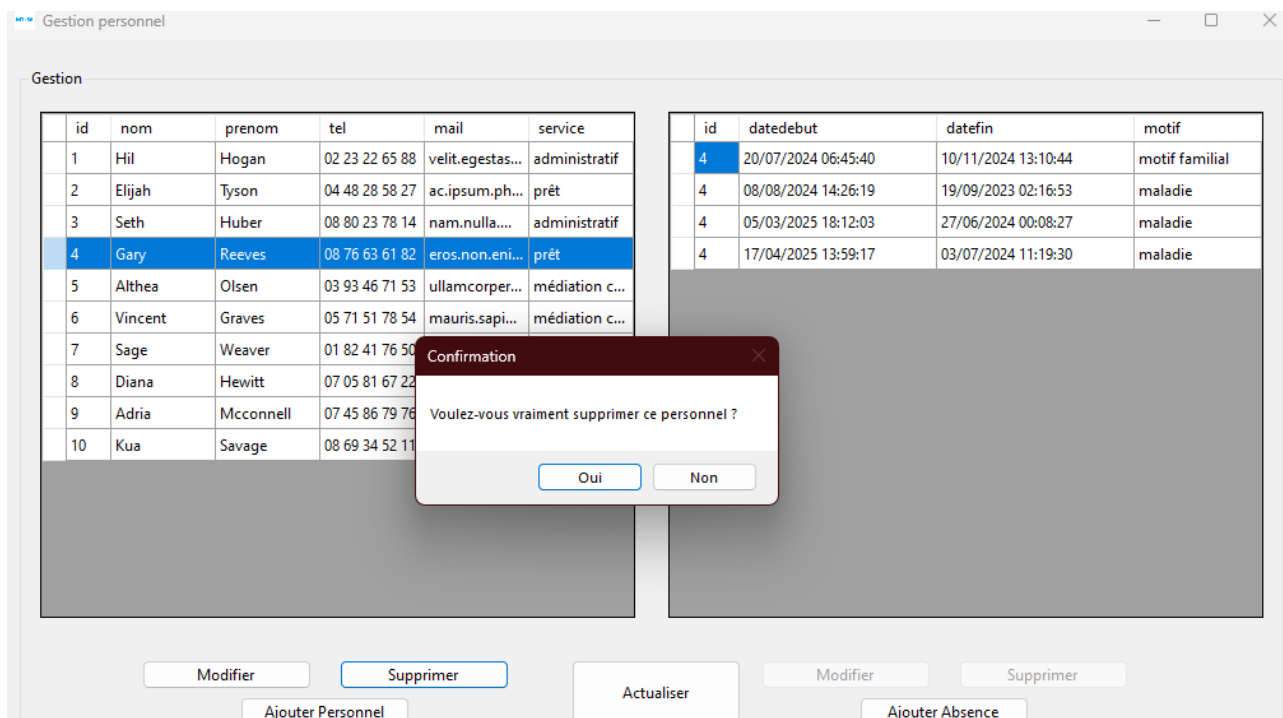
Cette étape était globalement la création du code de tout le logiciel, j'ai enfin fait fonctionner toutes les fonctionnalités de l'application en me basant bien sûr, sur le dossier documentaire, j'ai par exemple fait apparaître des boites de dialogues pour respecter les confirmations avant la suppression ou la modification d'un personnel ou d'une absence, comme demandé dans le dossier documentaire. J'ai tester et donc vérifier plusieurs scénarios et j'ai rencontré des bugs, par exemple il a fallut empecher la multi sélection dans les data grid view pour faire en sorte que l'utilisateur puisse sélectionner seulement un personnel et une absence pour éviter de créer une erreur, car le logiciel a était conçu pour s'occuper d'une sélection à la fois (suppresion d'un personnel et d'une

absence à la fois). Il a également fallu empêcher de cliquer sur les boutons si certaines conditions n'étaient pas remplies, par exemple, si aucun personnel n'est sélectionné alors les boutons Supprimer et Ajouter devait être désactivé (grisé).

Boutons grisé selon la situation :



Boite de dialogue pour confirmer la suppression d'un personnel :



Code pour désactivé les boutons selon la sélection :

```
private void dgv_Personnel_SelectionChanged(object sender, EventArgs e)
{
    if (dgv_Personnel.SelectedRows.Count == 1)
    {
        idPersonnelSelected = (int)dgv_Personnel.CurrentRow.Cells[0].Value;
        idPersonnelSelectedLocal = idPersonnelSelected - 1;
        lbl_idPersonnelSelectedNb.Text = idPersonnelSelected.ToString();
        lbl_idPersonnelSelectedNbAbsence.Text = idPersonnelSelected.ToString();
        btn_GoModifierPersonnel.Enabled = true;
        btn_SupprimerPersonnel.Enabled = true;
        btn_GoAjouterAbsence.Enabled = true;
        RemplirListboxAbsence();
    }
    else
    {
        btn_GoModifierPersonnel.Enabled = false;
        btn_SupprimerPersonnel.Enabled = false;
        btn_GoAjouterAbsence.Enabled = false;
        RemplirListboxAbsence();
    }
}
```

Pour conclure, pour moi cette partie était la plus complexe de loin car c'était la création des fonctionnalités du logiciel, c'est ce qui ma pris le plus de temps bien sûr. En tout cas pour moi, c'est ici que j'ai appris et découvert énormément de chose.

Étape 5 :

L'idée était de faire une vidéo comme documentation utilisateur, une sorte de « tutoriel ». Je n'ai jamais vraiment fait de vidéo « tutoriel », ça était en tout cas peu complexe et plutôt sympathique à faire. J'ai essayé d'articuler le plus possible et d'être clair et efficace. Créer une documentation utilisateur est en soi primordiale pour s'assurer que l'utilisateur comprenne comment utiliser le logiciel correctement, même si différents formats sont possibles comme une vidéo, un wiki, etc..

Vidéo Youtube

Étape 6 :

Je devais m'occuper du setup du logiciel. J'ai rencontré un problème plutôt dérangeant avec l'extension « Microsoft Visual Studio Installer Projects 2022 », j'ai réussi à le résoudre grâce à plusieurs recherches sur internet. En effet, il est très courant de passer par internet pour trouver des solutions à des problèmes complexes car quelques autres développeurs ont rencontrés le problème que j'ai eu.

Site stackoverflow où j'ai pu trouver des informations pour résoudre le problème :

The screenshot shows a Stack Overflow page for a question titled ".dll Targeting 'AMD64' incompatible with x86 Project (Tried Common Solutions)". The question was asked by lishiyu on Mar 17, 2022. It has 3 answers, with the top one by Mike Irving providing a link to a Microsoft article on configuring projects to target platforms. The page also shows a list of related questions on the right side.

Question: .dll Targeting 'AMD64' incompatible with x86 Project (Tried Common Solutions)

Answers:

- 1 To configure a project to target a 64-bit platform
- On the menu bar, choose Build > Configuration Manager.
- In the Active solution platform list, choose a 64-bit platform for the solution to target, and then choose the Close button.

Related Questions:

- VS2010 project doesn't have x64 platform target
- 'X86' conflicts with target machine type 'x64'
- targeting amd64 is not compatible with the project's target platform x86 c#
- How to resolve the module machine type 'X86' conflicts with target machine type 'x64' Visual Studio
- Visual Studio: Not able to run any project when setting x64 as platform
- Could not load file or assembly 'Microsoft.Build.Framework, Version=15.1.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a'
- visual studio 2019 version c# probleme
- How to fix Visual Studio 2019 "The C# Project is targeting ".NETFramework,Version=v2.0", which is not installed on this machine
- How to build x64 assembly project in visual studio 2017/2019
- The C# project is targeting NETFramework Version=v4.7.1, which is not installed on this machine

Conclusion :

Tout d'abord, la création de ce projet à été très enrichissante. En effet j'ai pu créer un projet du début jusqu'à l'arrivée sur mon portfolio, j'ai rencontré beaucoup de bugs et de problèmes divers même si le problème le plus inattendu a été celui lié aux problèmes de build du setup. J'ai maintenant clairement une meilleure vision de la création complète d'une application de bureau, en débutant par la préparation de l'environnement de travail, la création des maquettes avec Pencil, la création de la Vue avec les interfaces graphiques, la création des classes métiers qui devait donc stocker les données qui étaient ici dans des tables SQL, puis enfin la création des fonctionnalités en elle mêmes. Pour conclure, ce projet m'a clairement permis de voir que la création d'une application de bureau nécessite une très bonne organisation, préparation et adaptabilité pour réagir aux divers problèmes qui pourraient survenir durant le processus de production.