

Compte rendu atelier n°2 :

Mediatek86

Nicolas FRERE

-Sommaire :

Introduction

Étape #1 :

- Mise en place de la base de données

Étape #2 :

- Création du projet et de ces dossiers
- Création du kanban et du repos sur github
- Test de la connexion a la base de données
- Prototypage des vus
- Mise en places des vus

Étape #3 :

- Code de la parti modèle, classes métiers
- Implémentation du dal
- Génération de la première documentation technique au format chm

Étape #4 :

- Code de la connexion (de la form jusqu'au dal) :
 - _création et code de dal.connectionAcces, code des fonctionnalité de la vue et de son contrôleur
- Code de la parti personnel (de la form jusqu'au dal) :
 - _création d'un deuxième contrôleur, code de la parti personnel de la vue, du contrôleur, création et code de dal.personnelAcces
- Code de la parti absence (de la form jusqu'au dal) :
 - _code de la parti absence de la vue, du contrôleur, création et code de dal.absenceAcces
- Code de l'initialisation des parti graphique pour les services et les motifs
 - _code du dal initAcces et du code nécessaire a son fonctionnement dans la vue et le contrôleur
- Finalisation :
 - _modifications, testes et vérifications des diverses fonctionnalités

Conclusion

Introduction :



Le programme est une interface graphique permettant d'interagir sur une base de données.

Il a été développé en C# sous Visual Studio 2019 et avec MySQL en SGBD.

C'est une application de bureau qui va permettre de gérer le personnel de médiathèques, leur affectation à un service et leurs absences.

Son but est de pouvoir modifier/supprimer/ajouter du personnel sur une base de donnée et de même pour les absences de chaque personnel.

La structure de la base de données nous est fournit.

Le programme est paramétré pour tourner en local sur la même machine que le SGBD.

Étape #1 :

-Mise en place de la base de données :

Pour mettre en place la base de données je commence par récupérer le schéma MCD puis je génère le modèle logique. Je crée une base sous Myphpadmin, j'importe le script créer avec Windesign. J'ai le squelette de ma base de données.

Je dois générer des données pour la remplir, pour cela j'utilise generatedata.com.

The screenshot displays the 'New Data Set' interface on generatedata.com. It features a table with columns: Data Type, Column Name, Examples, and Options. The table lists six data types: Auto-Increment (idpersonnel), Names (nom, prenom), Phone / Fax (tel), Email (mail), and Number Range (idservice). The 'Options' column for each row contains specific configuration details, such as 'Start at: 1', 'Increment: 1', 'Placeholder string', 'Name', 'Surname', 'Phone', 'Email', and 'Between 1 and 3'.

Below the table, a 'MySQL' section shows the generated SQL code. The code includes a 'DROP TABLE IF EXISTS' statement followed by a 'CREATE TABLE' statement with columns: id, idpersonnel, nom, prenom, tel, and mail. The 'id' column is defined as 'mediumint(8) unsigned NOT NULL auto_increment'.

Arrows point to the 'MySQL' section, highlighting the 'Format code' field, which is set to 'y-LL-dd HH:mm:ss'. A note indicates that this format is used to avoid date overlaps (datedebut before datefin).

At the bottom, a 'Pour dl les data format sql' note points to the 'GENERATE' button.

! MySQL datetime pour le format
rien trouvé de tres pertinent pour éviter
que les dates ces surperpose (datedebut
avant datefin)

Pour dl les data format sql

Puis avec les données générer et celles fournit, je remplis ma base de données

Exécuter une ou des requêtes SQL sur la table « ap2.responsable »

```
1 INSERT INTO responsable (login, pwd) VALUES
2 ('patrick', SHA2('play@ts', 256));
```

✓ 1 ligne insérée. (traitement en 0.0003 seconde(s))

```
INSERT INTO responsable (login, pwd) VALUES ('patrick', SHA2('play@ts', 256));
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

Exécuter une ou des requêtes SQL sur la table « ap2.motif »

```
1 INSERT INTO motif (idmotif, libelle) VALUES
2 (1, 'vacances'),
3 (2, 'maladie'),
4 (3, 'motif familial'),
5 (4, 'congé parental');
6
```

✓ 4 lignes insérées. (traitement en 0.0002 seconde(s))

```
INSERT INTO motif (idmotif, libelle) VALUES (1, 'vacances'), (2, 'maladie'), (3, 'motif familial'), (4, 'congé parental');
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

Exécuter une ou des requêtes SQL sur la table « ap2.service »

```
1 INSERT INTO service (idservice, nom) VALUES
2 (1, 'administratif'),
3 (2, 'médiation culturelle'),
4 (3, 'prêt');
```

✓ 3 lignes insérées. (traitement en 0.0002 seconde(s))

```
INSERT INTO service (idservice, nom) VALUES (1, 'administratif'), (2, 'médiation culturelle'), (3, 'prêt');
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

Afficher la zone SQL

✓ 8 lignes insérées.
Identifiant de la ligne insérée : 0 (traitement en 0.0003 seconde(s))

```
INSERT INTO personnel (idservice, nom, prenom, tel, mail) VALUES (1, "Aaron", "Barton", "0725780805", "aaron.barton@gmail.com"), (2, "Armand", "Hall", "0625778294", "armand.hall@gmail.com"), (3, "Owen", "Albert", "0625778478", "owen.albert@gmail.com"), (4, "Brian", "Bartlett", "0625784785", "brian.bartlett@gmail.com"), (5, "Elliot", "Powell", "0778542794", "elliot.powell@gmail.com"), (6, "Kennan", "Emerson", "0778489841", "kennan.emerson@gmail.com"), (7, "Hannik", "Stokes", "0625488728", "hannik.stokes@gmail.com"), (8, "Samuel", "Edwards", "064057414", "samuel.edwards@gmail.com");
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

Je crée aussi la table responsable qui me permettra de filtrer l'accès au logiciel de gestion et dans laquelle je crée un responsable.

Et un utilisateur à qui j'élève les droits sur base de données du projet que j'utiliserais pour m'y connecter.

```
1 CREATE USER 'patrick'@'localhost' IDENTIFIED BY 'motdepasseuser';
2 GRANT USAGE ON *.* TO 'patrick'@'localhost';
3 GRANT ALL PRIVILEGES ON `atelier2`.* TO 'patrick'@'localhost';
4
```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0096 seconde(s).)

```
CREATE USER 'patrick'@'localhost' IDENTIFIED BY 'motdepasseuser';
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0001 seconde(s).)

```
GRANT USAGE ON *.* TO 'patrick'@'localhost';
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0003 seconde(s).)

```
GRANT ALL PRIVILEGES ON `atelier2`.* TO 'patrick'@'localhost';
```

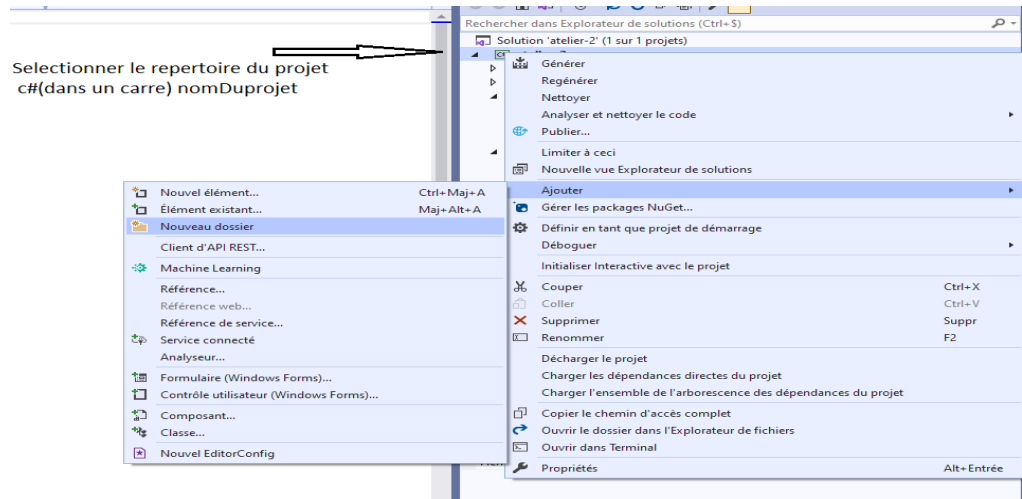
[Éditer en ligne] [Éditer] [Créer le code source PHP]

Étape #2 :

-Création du projet et de ces dossiers

Création d'un nouveau projet windows form avec Visual Studio .

Je crée les différents dossiers du projet, la view , le controller, le dal et le model .

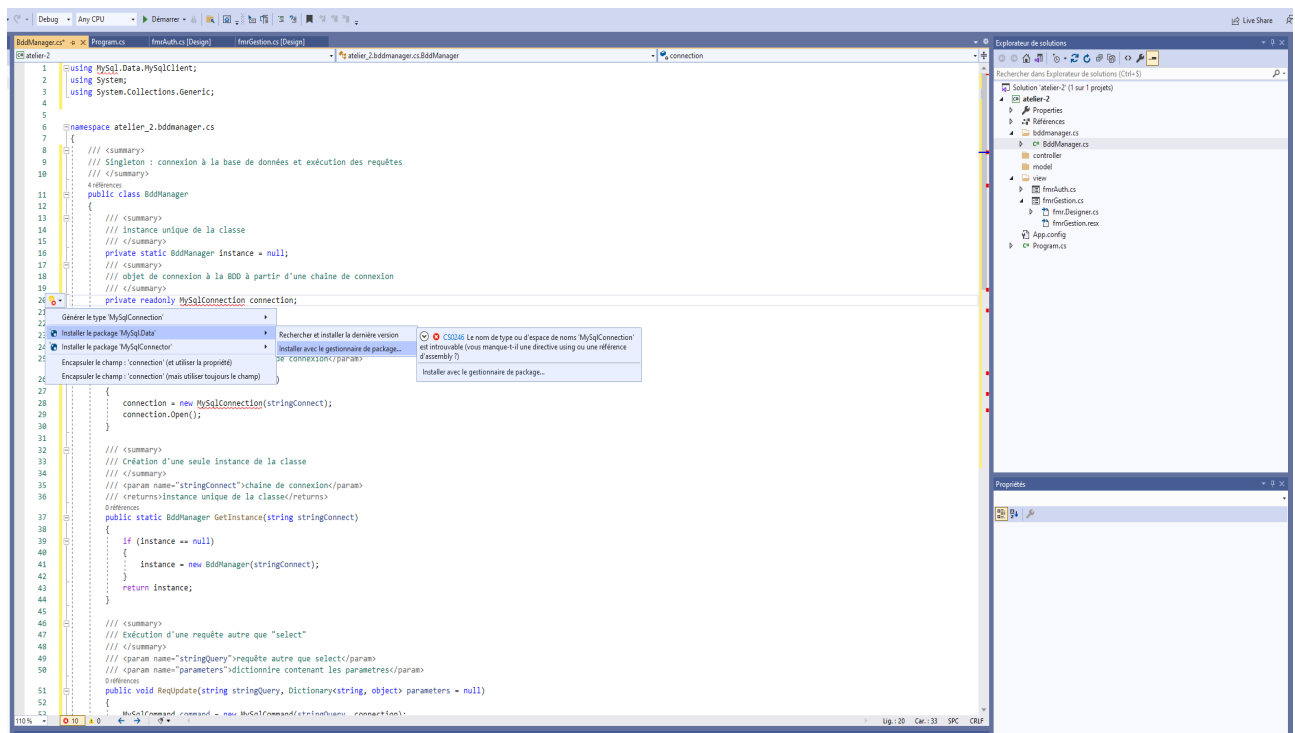
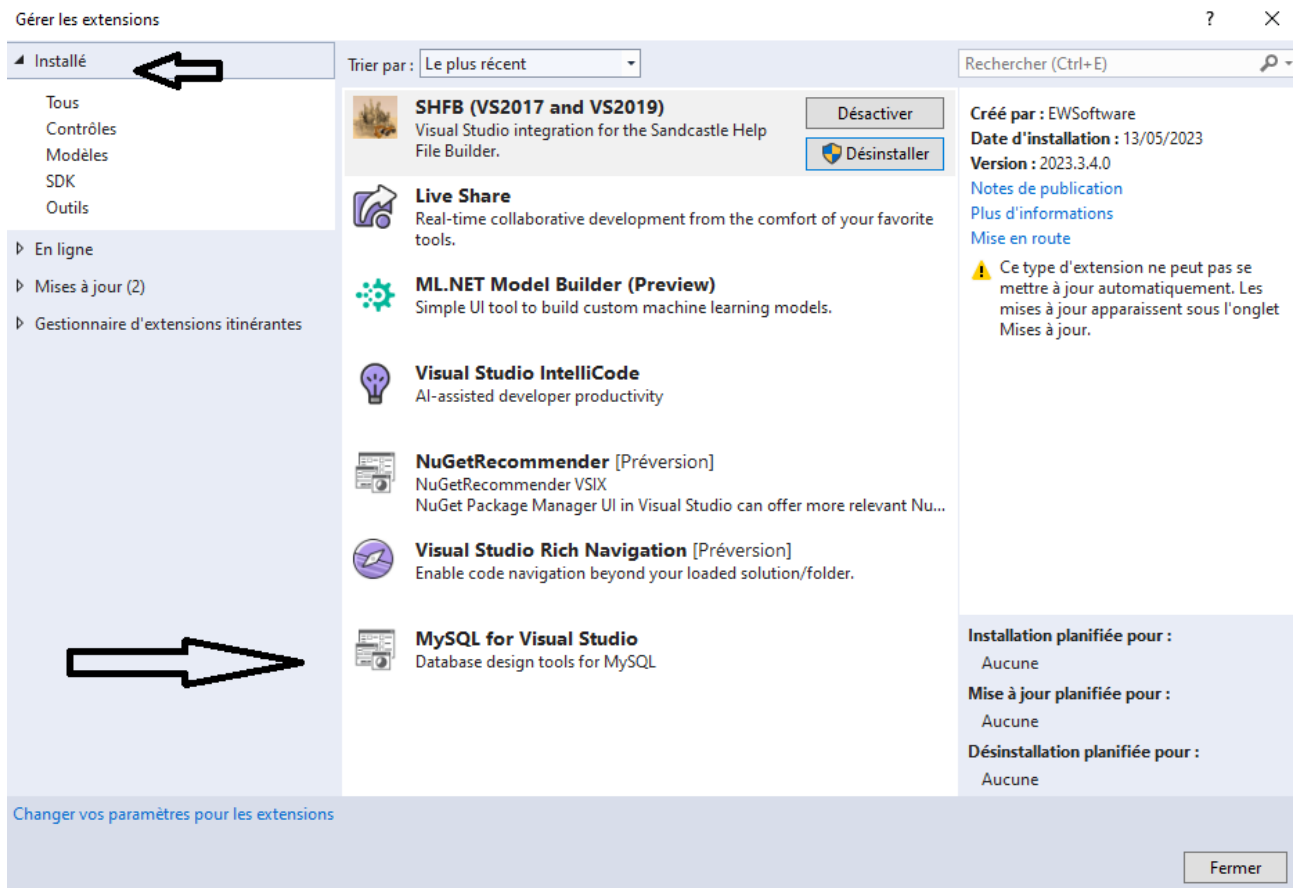


-Création du kanban et du repos sur github

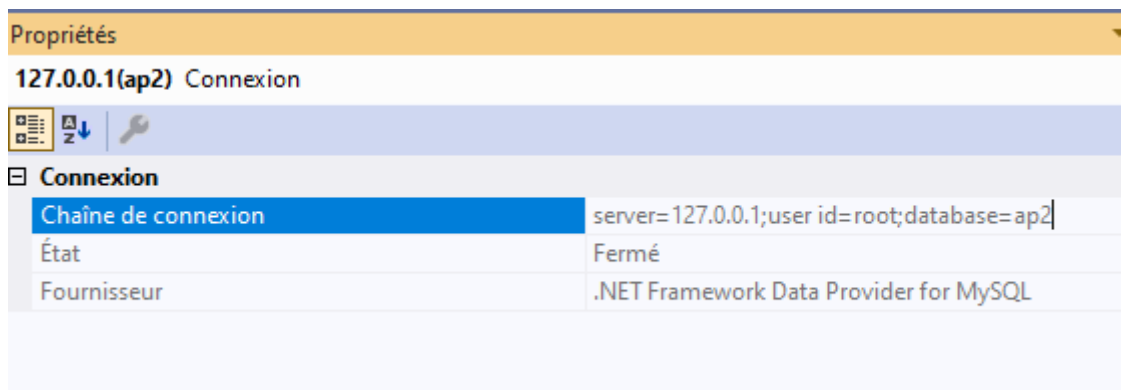
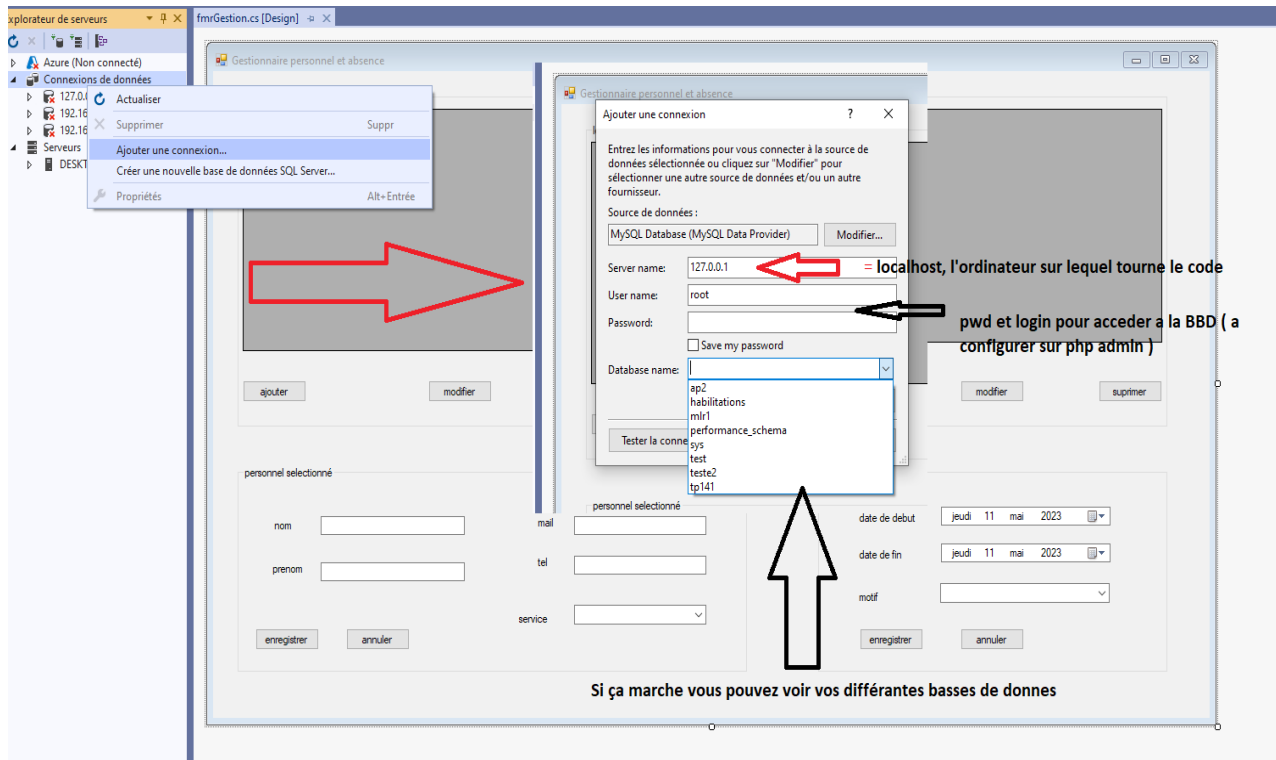
Je crée le répertoire github nécessaire au projet et le kanban qui lui est associé. N'ayant encore jamais utilisé de kanban, je lui fais afficher "hello world!" . Le test étant concluant je passe à la suite.

-Test de la connexion a la base de données

Je récupère le package bddManager que j'incorpore à mon programme et installe le packages MySql.Data. J'ai pu rencontrer des soucis en fonctionnent comme dans les cours. J'ai installé mySql directement depuis Visual Studio 2019 dans les extensions et a chaque nouveau programme installer le package pour que cela marche.



Je vérifie la connexion à ma base de données dans le menu de l'IDE, cela fonctionne, je passe à la suite



(le nom de la database sera changé par la suite par atelier2, ici l'image est un test avec comme user root, pour l'atelier l'user est "patrick" et il faut rajouter mot de passe : "Pwd = motdepasseuser ;")

-Prototypage des vus

Après études des cas d'utilisation et du diagramme de la base de données, je dessine un prototype

avec Pensil qui me servira de modèle a la création de mon interface graphique

The image shows a software prototype window titled "Gestionnaire personnel et absence". It is divided into two main sections: "le personnel" on the left and "absence(s)" on the right. Each section contains a list box with sample data, a set of action buttons (modifier, supprimer, ajouter), and a form for adding or editing records. The "le personnel" form includes fields for nom, prénom, and service. The "absence(s)" form includes fields for date de debut, date de fin, and motif. A central arrow button points from the personnel list to the absence list.

le personnel		
Nom, Prénom , Service, " 0244XXXXXX", "nom@mail.com"		
Nom, Prénom , Service, " 0244XXXXXX", "nom@mail.com"		
Nom, Prénom , Service, " 0244XXXXXX", "nom@mail.com"		
Nom, Prénom , Service, " 0244XXXXXX", "nom@mail.com"		
Nom, Prénom , Service, " 0244XXXXXX", "nom@mail.com"		
Nom, Prénom , Service, " 0244XXXXXX", "nom@mail.com"		
Nom, Prénom , Service, " 0244XXXXXX", "nom@mail.com"		

Buttons: modifier, supprimer, ajouter

Form fields: nom (O6XXXXXXX), prénom (email@hebergeur.org), service (dropdown)

Buttons: enregistrer, annuler

absence(s)		
14/08/2021 , 16/08/2021 , Motif		
14/08/2021 , 16/08/2021 , Motif		
14/08/2021 , 16/08/2021 , Motif		
14/08/2021 , 16/08/2021 , Motif		
14/08/2021 , 16/08/2021 , Motif		

Buttons: modifier, supprimer, ajouter

Form fields: date de debut (dropdown), date de fin (dropdown), motif (dropdown)

Buttons: enregistrer, annuler

The image shows a software prototype window titled "Authentification". It contains two input fields: "Login" with the text "Jean-michel" and "pwd" with masked text "*****". Below the fields is a button labeled "se connecter".

Login: Jean-michel

pwd: *****

se connecter

-Mise en places des vus

The screenshot shows a Windows-style application window titled "Gestionnaire personnel et absence". The interface is divided into several sections:

- le personnel**: A large empty rectangular area for displaying personnel data. Below it are three buttons: "ajouter", "modifier", and "supprimer".
- absences**: A large empty rectangular area for displaying absence data. Below it are three buttons: "ajouter", "modifier", and "supprimer".
- afficher absence(s)**: A button located between the "le personnel" and "absences" sections.
- personnel selectionné**: A form with input fields for "nom", "prenom", "mail", "tel", and a dropdown for "service". It includes "enregistrer" and "annuler" buttons.
- absence selectionné**: A form with date pickers for "date de debut" and "date de fin" (both showing "jeudi 11 mai 2023"), a dropdown for "motif", and "enregistrer" and "annuler" buttons.

Comme dans le programme étudié en cours "Habilitation" je fais le choix d'utiliser des DataGridView pour les affichages principaux. Les DateTimePicker seront aussi très pratique pour afficher les dates de début et date de fin de mes absences (la vue sera modifiée plus tard pour plus de praticités.)

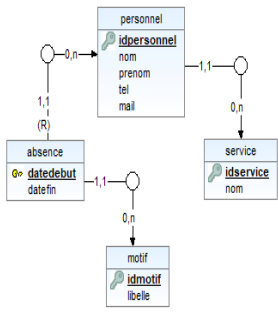
Je clone mon répertoire avec gitbash et je fais mon premier commit .

Etape #3 :

-Code de la parti modèle, classes metiers

L'affichage se faisant avec des objets de type BindingSource qui permettent l'affichage des contenant d'une classe ou leur méthode ToString(), je fais le choix de créer une classe pour chaque table de ma base. À la lecture du modèle conceptuel de données, on peut constater que chaque personnel fait partie d'un service et chaque absence a un motif. Je conçois les classes en conséquence.

```
/
3 namespace atelier2.model
3 {
3     /// <summary>
1     /// Classe métier liée à la table Personnel
2     /// </summary>
1 référence
3     class Personnel
4     {
5         /// <summary>
5         /// Valorise les propriétés
7         /// </summary>
3         /// <param name="idpersonnel"></param>
3         /// <param name="nom"></param>
3         /// <param name="prenom"></param>
1         /// <param name="tel"></param>
2         /// <param name="mail"></param>
3         /// <param name="service"></param>
0 références
4         public Personnel(int idpersonnel, string nom, string prenom, string tel, string mail, Service service)
5         {
5             this.Idpersonnel = idpersonnel;
7             this.Nom = nom;
3             this.Prenom = prenom;
3             this.Tel = tel;
3             this.Mail = mail;
1             this.Service = service;
2         }
3         1 référence
3         public int Idpersonnel { get; }
4         1 référence
4         public string Nom { get; set; }
5         1 référence
5         public string Prenom { get; set; }
5         1 référence
5         public string Tel { get; set; }
7         1 référence
7         public string Mail { get; set; }
3
3         1 référence
3         public Service Service { get; set; }
1     }
2 }
```



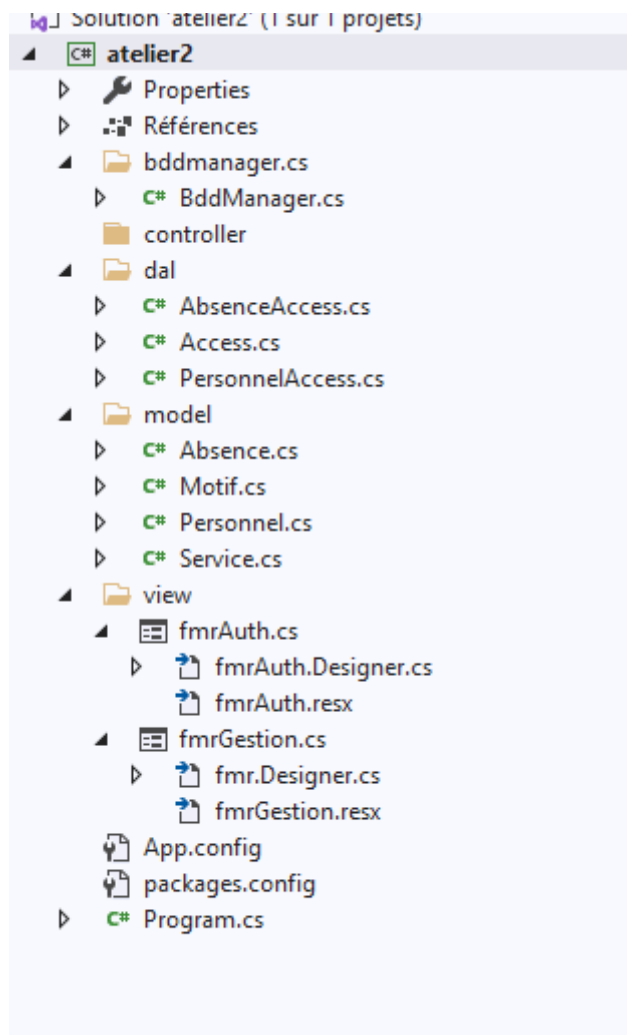
Les variables de classe sont en public, majuscule avec setter et getter pour pouvoir être instancié ou modifié depuis le dal correspondant, ici personnelAcces . Pour rester cohérent avec la base de données les variables "clef primaire n'ont pas de setter.

Je crée une méthode override ToString() pour que les objets de type service et motif ne m'affiche que leur nom/libelle dans mes dataGridView .

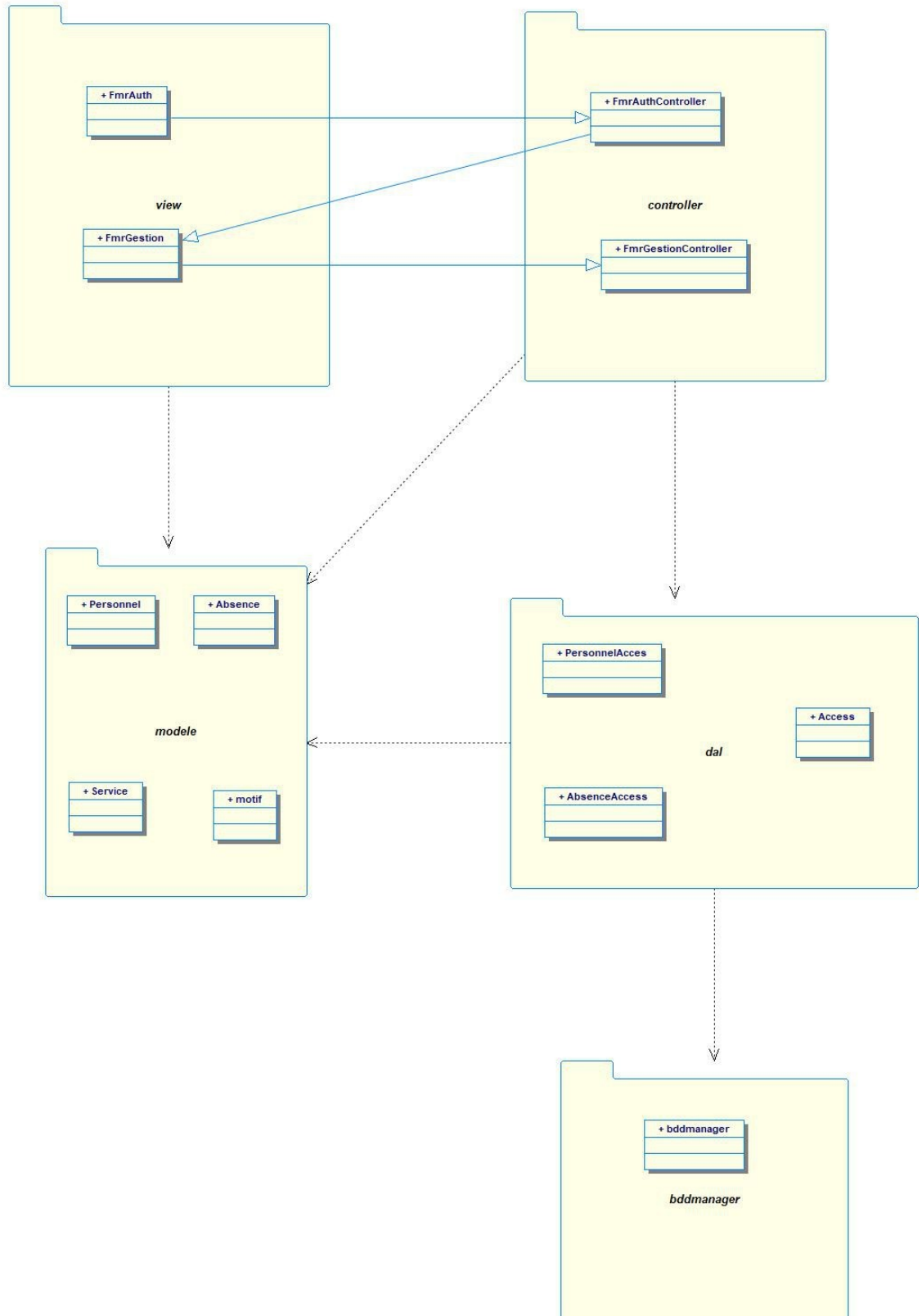
-Implémentation du dal

Je crée mes différentes classes qui me permettront par la suite d'accéder à ma base de données. Access classe en singleton me permettant de centraliser la seule et unique instance de Bddmanager (reprise comme BddaManager du td "Habitations").

Pour la gestion du personnel et celle des absences, je crée PersonnelAccess et AbsencesAccess que je compléterai plus tard.

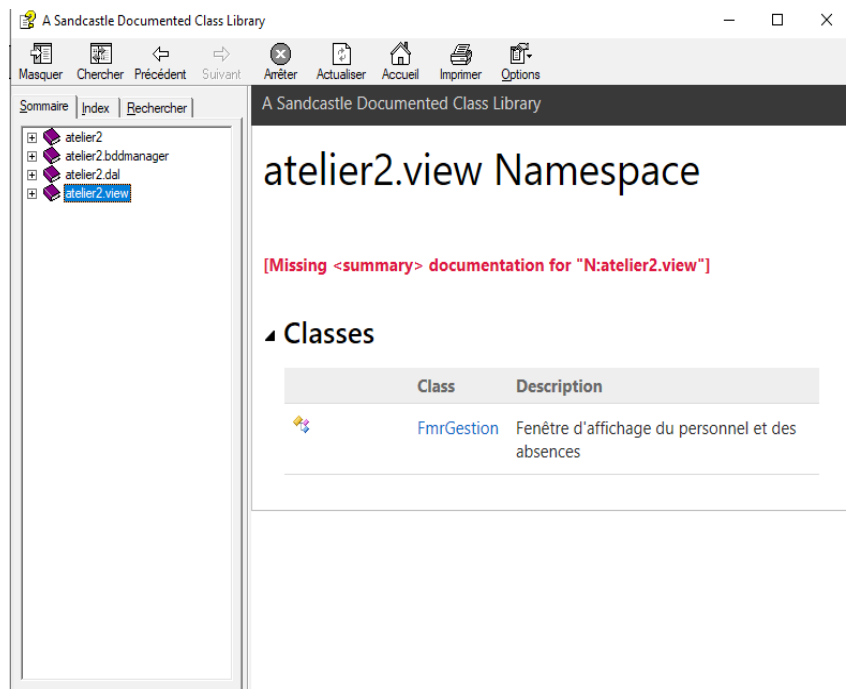


Je dessine un brouillon de l'architecture de mon projet pour pouvoir développer plus rapidement.



-Génération de la premiere documentation technique au format chm

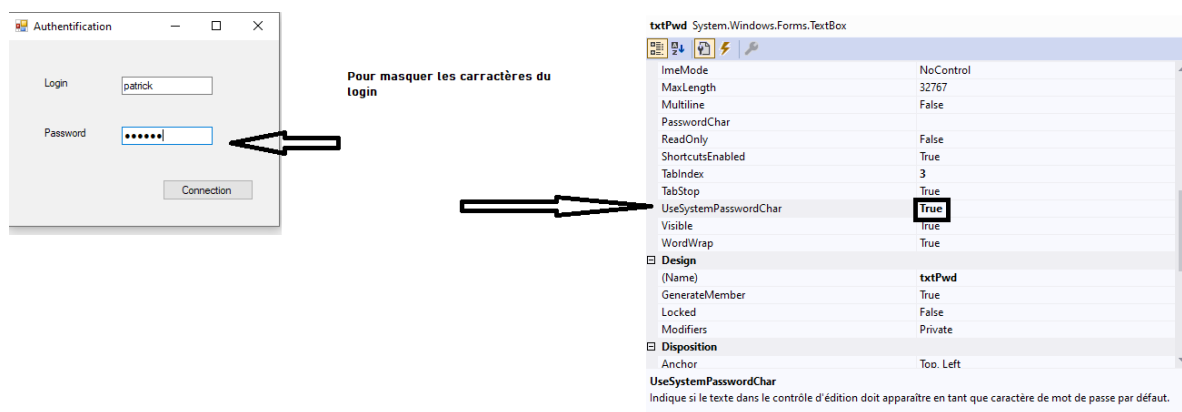
Après quelques difficultés a comprendre le fonctionnement du logiciel Sandcastle je génère la première documentation technique du projet en format chm.



Etape #4 :

-Code de la connexion (de la form jusqu'au dal):

-création et code de dal.connectionAcces, code des fonctionnalités de la vue et de son contrôleur



Je crée un contrôleur pour la forme, FmrAuthController (instancier par la vue (FmrAuth)) et ConnectionsAcces dans le dal avec qui je ferais la **requête** de vérification de mot de passe.

```
/// <summary>
/// retourne vrai si la combinaison mdp pwd existe dans la base de données
/// </summary>
/// <param name="login"></param>
/// <param name="pwd"></param>
/// <returns>vrai ou faux</returns>
1 référence
public bool GetLaConnection(string login, string pwd)
{
    string req = "SELECT * from responsable WHERE login = @login AND pwd = SHA2(@pwd, 256)";
    Dictionary<string, object> parameters = new Dictionary<string, object>();
    parameters.Add("@login", (Object)login);
    parameters.Add("@pwd", (Object)pwd);
    try
    {
        List<Object[]> records = access.Manager.ReqSelect(req, parameters);
        if (records.Count == 1)
        {
            return true;
        }
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
        Environment.Exit(0);
    }
    return false;
}
```

```
/// <summary>
/// Valide ou non l'authentification login pwd du responsable
/// Si valide lance la fenêtre de gestion et renvoi vrai
/// Sinon valide renvoi false
/// </summary>
/// <param name="login"></param>
/// <param name="psw"></param>
/// <returns></returns>
1 référence
```

```
public bool GetLaConnection(string login, string psw)
{
    if (connectionAccess.GetLaConnection(login, psw))
    {
        initGestion();
        return true;
    }
    return false;
}
```

Si le mdp est valide, la deuxième Form est lancé depuis une methode privé



```
1 référence
private void initGestion()
{
    gestion = new FmrGestion();
    gestion.Show();
}
```

Ici dans le controller de la Form Connection

-Code de la parti personnel (de la form jusqu'au dal) :

-création d'un deuxième contrôleur, code de la parti personnel de la vue, du contrôleur, création et code de dal.personnelAcces

Si le retour est positif, mon contrôleur instancie ma deuxième vue (celle du logiciel de gestion) qui instanciera son contrôleur.

Je crée donc la classe FmrGestionController qui me servira à traiter les demandes de la vue à la base de données (tout ce qui est dans le dal) et qui lui renverra à la vue les informations nécessaires si besoin.

Je crée la méthode GetLesPersonnel dans PersonnelAccess et dans le controller pour pouvoir tester l'affichage de ma première DataGridView dans la vue.

Je code rapidement un affichage en dur pour remplir les comboBox.

Je fais le choix conscient a ce moment du développement d'utiliser le pattern SOLID, malgré la lourdeur d'avoir un code peu factorisé et long, le fait d'avoir des responsabilités uniques et une ségrégation des interfaces me permettra une meilleure maintenance et une meilleure lisibilité en cas de modification.

Nom	Prenom	Tel	Mail	Service
Aaron	Barton	0725789885	quis.arcu@google.edu	administratif
Armand	Wall	0625778294	arte@yahoo.edu	médiation culturelle
Brian	Bartlett	0625784785	quisque.ac@aol.net	médiation culturelle
Elliott	Powell	0778542794	tristique.pharetra@protonmail.org	prêt
JeanPatrick	Dupont	0754568714	patrick@supastart.com	prêt
Kennan	Emerson	0778489841	orci.adipiscing.non@google.edu	administratif
Mannix	Stokes	0625488728	dolor@hotmail.net	médiation culturelle
Owen	Albert	0225778478	mauris.magna.duis@cloud.net	administratif
Pascal	Michel	0245412847	Fmichel241@protonmail.com	administratif
Samuel	Edwards	0549857414	gravida.praesent.eu@google.com	médiation culturelle

Le code étant fonctionnel, je passe à la suite et je code les différentes options pour gérer le personnel. Pour le moment je n'ai pas besoin de réfléchir car cela a déjà été fait de manière très similaire dans le td "Habitations". Je code donc rapidement les différentes méthodes nécessaire dans le contrôleur personnelAcces et la form en adaptant à la situation . Je supprime mon bouton ajouter, les labels des groupbox me permettant de guider l'utilisateur (ici, ils indiquent si un personnel est en cours de modification, ou si il est possible d'en créer un nouveau, il n'est pas possible de faire les deux en même temps).

-Code de la parti absences (de la form jusqu'au dal) :

-code de la parti absence de la vue, du controlleur, creation et code de dal.absenceAcces

Sur le même schéma, je commence par coder les éléments me permettant de charger les absences d'un personnel dans le controller et AbsencesAccess. Si aucune ligne n'est sélectionné dans la dataGridView où est affiché le personnel, je code le bouton "afficher les absences" pour qu'il avertisse l'utilisateur qu'il doit sélectionner une ligne et il ne se passe rien. Dans le cas contraire le personnel sélectionné est assigné a une variable de classe de ma form pour faciliter la gestion. En cas d'action sur la liste de personnel, remplirListePersonnel() s'actualise et actualise l'assignation du personnel sélectionné . Je code remplirListeAbsence() sur le même modèle que remplirListepersonnel().

```
/// <summary>
/// Affiche le personnel
/// </summary>
3 références
private void RemplirListePersonnel()
{
    List<Personnel> lesPersonnels = controller.GetLesPersonnels();
    bdgPersonnel.DataSource = lesPersonnels;
    dgvPerso.DataSource = bdgPersonnel;
    dgvPerso.Columns["idpersonnel"].Visible = false;
    dgvPerso.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
    dgvPerso.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;
}

/// <summary>
/// Affiche les absences du personnel sélectionné
/// </summary>
/// <param name="personnel"></param>
4 références
private void RemplirListeAbsence(Personnel personnel)
{
    List<Absence> lesAbsences = controller.GetLesAbsences(personnel);
    bdgAbsence.DataSource = lesAbsences;
    dgvAbs.DataSource = bdgAbsence;
    dgvAbs.Columns["idpersonnel"].Visible = false;
    dgvAbs.Columns["datedebut"].DefaultCellStyle.Format = "d/M/yyyy";
    dgvAbs.Columns["datefin"].DefaultCellStyle.Format = "d/M/yyyy";
    dgvAbs.Columns["datedebut"].HeaderText = "Date de debut";
    dgvAbs.Columns["datefin"].HeaderText = "Date de fin";
    dgvAbs.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
    dgvAbs.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;
    gbxAbsences.Text = "absence de " + personnel.Nom + " " + personnel.Prenom;
    gbxAbs.Enabled = true;
}
```

```
1 référence
public List<Absence> GetLesAbsences(Personnel personnel)
{
    List<Absence> lesAbsences = new List<Absence>();
    if (access.Manager != null)
    {
        string req = "select a.idpersonnel as idpersonnel, a.datedebut as datedebut, a.datefin as datefin, a.idmotif as idmotif, m.libelle as libelle ";
        req += "from absence a join motif m on (a.idmotif = m.idmotif) ";
        req += "where idpersonnel = @idpersonnel ";
        req += "order by datedebut desc ";
        try
        {
            Dictionary<string, object> parameters = new Dictionary<string, object>();
            parameters.Add("@idpersonnel", personnel.Idpersonnel);
            List<Object[]> records = access.Manager.ReqSelect(req, parameters);
            if (records != null)
            {
                Console.WriteLine(records.Count);

                foreach (Object[] record in records)
                {
                    Motif motif = new Motif((int)record[3], (string)record[4]);
                    Absence absence = new Absence((int)record[0], (DateTime)record[1], (DateTime)record[2], motif);
                    lesAbsences.Add(absence);
                }
            }
        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
            Environment.Exit(0);
        }
    }
    return lesAbsences;
}
```


le personnel

Nom	Prenom	Tel	Mail	Service
Arma	Wall	0625778294	ante@yahoo.edu	médiation culturelle
Brian	Bartlett	0625784785	quisque.ac@aol.net	médiation culturelle
coucou	ploip	sdsd	lkedkl@ploip.com	administratif
Elliott	Powell	0778542794	tristique.pharetra@protonmail.org	prêt
JeanPatrick	Dupont	0754568714	patrick@supastart.com	prêt
Kennan	Emerson	0778489841	orci.adpiscing.non@google.edu	administratif
Mannix	Stokes	0625488728	dolor@hotmail.net	médiation culturelle
Owen	Albert	0225778478	mauris.magna.duis@cloud.net	médiation culturelle
Pascal	Michel	0245412847	Pmichel241@protonmail.com	administratif

modifier

supprimer

absences

Datedebut	Datefin	Motif
02/09/2023 14:19	11/03/2023 05:04	motif familial
20/04/2023 22:12	21/07/2023 23:41	motif familial
05/01/2023 06:06	24/01/2023 06:06	maladie
07/12/2022 11:44	16/09/2023 07:17	maladie
06/04/2022 02:45	05/02/2023 04:13	maladie
27/01/2022 12:13	02/06/2023 14:07	vacances

modifier

supprimer

ajouter un personnel

nom

mail

prenom

tel

service

enregistrer

annuler

ajouter une absence

date de debut

lundi 15 mai 2023

date de fin

lundi 15 mai 2023

motif

enregistrer

annuler

afficher absence(s)

L'affichage fonctionnant je passe aux différentes options de la partie absences. Je recopie les fonctions très proche de PersonnelAccess dans AbsencesAccess et change les noms. Les dateTimePicker en C# étant très simple d'utilisation avec les objets de type DateTime la difficulté a été de convertir les MySqlDateTime en objet de type DateTime et inversement. Après une rapide recherche j'arrive à obtenir quelque chose de fonctionnel . Le souci se situe au niveau des heures et des secondes. J'ai fait le choix arbitraire sur mon programme de ne pouvoir paramétrer que les jours dans l'affichage alors que mon format de données s'arrête a l'échelle des secondes.

```

1 référence
public void AddAbsence(Absence absence)
{
    if (access.Manager != null)
    {
        Console.WriteLine("AddAbsence");

        string req = "insert into absence(idpersonnel, datedebut, datefin, idmotif) ";
        req += "values (@idpersonnel, @datedebut, @datefin, @idmotif)";
        Dictionary<string, object> parameters = new Dictionary<string, object>();
        parameters.Add("@idpersonnel", absence.Idpersonnel);
        parameters.Add("@datedebut", absence.Datedebut.ToString("yyyy-MM-dd HH:mm"));
        parameters.Add("@datefin", absence.Datefin.ToString("yyyy-MM-dd HH:mm"));
        parameters.Add("@idmotif", absence.Motif.Idmotif);
        try
        {
            access.Manager.ReqUpdate(req, parameters);
        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
            Environment.Exit(0);
        }
    }
}

```

DateTime.ToString pour etre au format datetime MySQL

```

/// <summary>
/// Demande de suppression d'une absence
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
1 référence
private void BtnAbsSupprimer_Click(object sender, EventArgs e)
{
    if (dgvAbs.SelectedRows.Count > 0)
    {
        Absence absence = (Absence)bdgAbsence.List[bdgAbsence.Position];
        if (MessageBox.Show("Voulez-vous vraiment supprimer l'" + gbxAbsences.Text + " du " + absence.Datedebut.ToString("d/M/yyyy")
            + " au " + absence.Datefin.ToString("d/M/yyyy") + " ?", "Confirmation de suppression", MessageBoxButtons.YesNo) == DialogResult.Yes)
        {
            controller.DelAbsence(absence);
            Console.WriteLine("supprimer");
            RemplirListeAbsence(personnelSelect);
        }
    }
    else
    {
        MessageBox.Show("Une ligne doit être sélectionnée.", "Information");
    }
}

```

Je bricole pour que cela soit cohérent : a l'utilisation quand on modifie ou ajoute une absence mon programme vérifie si le jour n'a pas déjà été renseigné pour ce personnel.

```

/// <summary>
/// Verifie qu'une absence n'a pas deja été assigné le même jour
/// </summary>
/// <param name="absence"></param>
/// <returns></returns>
2 références
private bool VerifAbs(Absence absence)
{
    int compt = 0;
    if (enCoursDeModifAbsence)
        compt--;
    foreach (Absence absenceAVerifier in bdgAbsence.List)
    {
        if (absence.Datedebut.ToString("yyyy-MM-dd") == absenceAVerifier.Datedebut.ToString("yyyy-MM-dd"))
        {
            compt++;
            if (compt == 1)
            {
                MessageBox.Show("Cette date est deja renseigné", "Information");
                return false;
            }
        }
    }
    return true;
}

```

La faite de vérifier si une absence n'empiète ou n'est pas comprise dans une autre n'étant pas dans le cahier des charges, je ne l'ai pas rajouté, mais en cas de demande cela peu être intégré très rapidement au programme.

On ne peut pas modifier une absence quand une ou plusieurs son renseigné le même jour. Si on veut changer la date de début d'une absence, il faut la supprimer et la recréer.

L'état de mon contrôleur a ce moment du développement (test avant création des commentaires) .

```
3 références
public class FmrGestionController
{
    /// <summary>
    /// objet d'accès aux opérations possibles sur Personnel
    /// </summary>
    private readonly PersonnelAccess personnelAccess;
    /// <summary>
    /// objet d'accès aux opérations possibles sur Absence
    /// </summary>
    private readonly AbsenceAccess absenceAccess;

    /// <summary>
    ///
    /// </summary>
    1 référence
    public FmrGestionController()
    {
        personnelAccess = new PersonnelAccess();
        absenceAccess = new AbsenceAccess();
    }

    1 référence
    public List<Personnel> GetLesPersonnels()
    {
        return personnelAccess.GetLesPersonnels();
    }
    - ...

    1 référence
    public List<Absence> GetLesAbsences(Personnel personnel)
    {
        return absenceAccess.GetLesAbsences(personnel);
    }

    1 référence
    public void AddPersonnel(Personnel personnel)
    {
        personnelAccess.AddPersonnel(personnel);
    }

    1 référence
    public void AddAbsence(Absence absence)
    {
        absenceAccess.AddAbsence(absence);
    }

    1 référence
    public void UpdatePersonnel(Personnel personnel)
    {
        personnelAccess.UpdatePersonnel(personnel);
    }

    1 référence
    public void UpdateAbsence(Absence absence)
    {
        absenceAccess.UpdateAbsence(absence);
    }

    1 référence
    public void DelPersonnel(Personnel personnel)
    {
        personnelAccess.DelPersonnel(personnel);
    }

    1 référence
    public void DelAbsence(Absence absence)
    {
        absenceAccess.DelAbsence(absence);
    }
}
```

-Code de l'initialisation des parti graphique pour les services et les motifs

-code du dal initAcces et du code nécessaire a son fonctionnement dans la vue et le contrôleur

Le remplissage de mes comboBox se faisait pour le moment en dur, ce qui n'est pas très évolutif (en cas d'ajout de table dans la base de données le programme ne marcherait plus). Sur les conseils de ma professeure, je décide donc de les charger depuis ma base de données.

Pour ne pas saturer de demande le SGBD, la liste de motifs et de services ne seront chargés qu'à l'initialisation de fmrGestion pour etre assigné au comboBox.

Pour conserver le style de conception du programme ou chaque fonction a une tache unique, je crée une classe InitAccess contenant les méthodes getLesServices() et getLesMotifs().

L'architecture du programme me permet d'effectuer très rapidement ces modifications, même si le code peu ce "répéter", il est très simple d'utilisation et organisé.

```
/// <summary>
/// Initialisation:
/// Creation du controleur et remplissage des listes
/// </summary>
1 référence
private void Init()
{
    controller = new FmrGestionController();
    RemplirListePersonnel();
    gbxAbs.Enabled = false;
    EnCourseDeModifPersonnel(false);
    EnCourseDeModifAbsence(false);
    RemplirListeService();
    RemplirListeMotif();
}
```

Dans FmrGestion (la vue)

```
/// <summary>
/// Affiche les services
/// </summary>
1 référence
private void RemplirListeService()
{
    List<Service> lesServices = controller.GetLesServices();
    bdgServices.DataSource = lesServices;
    cbxService.DataSource = bdgServices;
}

/// <summary>
/// Affiche les motifs
/// </summary>
1 référence
private void RemplirListeMotif()
{
    List<Motif> lesMotifs = controller.GetLesMotifs();
    bdgMotifs.DataSource = lesMotifs;
    cbxMotif.DataSource = bdgMotifs;
}
```

-Finalisation :

-modifications, testes et vérifications des diverses fonctionnalités

Le programme ne supprime pas les absences d'un personnel, quand le personnel est supprimé, je code cette fonctionnalité. Je n'ai qu'à rajouter un module DelLesAbsences dans AbsencesAcces et dans le contrôler, à qui je fournis avant de supprimer un personnel, l'objet de ce personnel.

```
/// <summary>
/// Demande de suppression d'un personnel
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
1 référence
private void BtnProSupprimer_Click(object sender, EventArgs e)
{
    if (dgvPerso.SelectedRows.Count > 0)
    {
        Personnel personnel = (Personnel)bdcPersonnel.List[bdcPersonnel.Position];
        if (MessageBox.Show("Voulez-vous vraiment supprimer " + personnel.Nom + " " + personnel.Prenom + " ?", "Confirmation de suppression", MessageBoxButtons.YesNo) == DialogResult.Yes)
        {
            controller.DelLesAbsences(personnel); ←
            controller.DelPersonnel(personnel);
            RemplirListePersonnel();
            if (dgvAbs.RowCount > 0)
            {
                Personnel personnelSelect = (Personnel)bdcPersonnel.List[bdcPersonnel.Position];
                this.personnelSelect = personnelSelect;
                RemplirListeAbsence(personnelSelect);
            }
        }
    }
    else
    {
        MessageBox.Show("Une ligne doit être sélectionnée.", "Information");
    }
}
```

Dans FmrGestion (vue)

```
/// <summary>
/// Demande de suppression des absences d'une personnel
/// </summary>
/// <param name="personnel">le personnel dont on effasse les absences</param>
1 référence
public void DelLesAbsences(Personnel personnel)
{
    if (access.Manager != null)
    {
        string req = "delete from absence where idpersonnel = @idpersonnel ";
        Dictionary<string, object> parameters = new Dictionary<string, object>();
        parameters.Add("@idpersonnel", personnel.Idpersonnel);
        try
        {
            access.Manager.ReqUpdate(req, parameters);
        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
            Environment.Exit(0);
        }
    }
}
```

Je modifie le programme pour avertir l'utilisateur par message en cas d'erreur, comme l'image si dessous .

```
/// <summary>
/// Création d'une seule instance de la classe
/// </summary>
/// <param name="stringConnect">chaîne de connexion</param>
/// <returns>instance unique de la classe</returns>
1 référence
public static BddManager GetInstance(string stringConnect)
{
    if (instance == null)
    {
        try
        {
            instance = new BddManager(stringConnect);
        }
        catch (Exception e)
        {
            MessageBox.Show(e.Message + "!", "Erreur", MessageBoxButtons.OK, MessageBoxIcon.Error);
            Environment.Exit(0);
        }
    }
    return instance;
}
```

Fenêtre d'erreur quand la connection ne peut pas s'effectuer, puis fermeture du programme
(ici dans bdd manager)



-Conclusion :

Après débogages les testes sont concluants le programme est fonctionnel.

J'ai essayé de bien séparer mes classes et modules pour conserver le principe de responsabilité unique. Malgré ça longueur le programme a été développé en moins d'une semaine. Avec la documentation, le site a fournir, la vidéo le projet a prix le double de ce temps.

J'ai pus m'intéresser a la sécurité en C#, a la protection contre les injections SQL.

Aux bonnes pratiques, la responsabilité unique contre l'interfaçage .

Des questions qui m'ont amené a d'autre questions elle m'ont permit de progresser mais " Plus on apprend, plus on mesure l'étendue de notre ignorance" .

J'espère que mon programme est clair, j'ai essayé de le faire le plus générique au possible. Certaines fonctionnalité auraient pus être codé, beaucoup en faite mais il n'y a pas de cahier des charges sur ce projet, juste des études de cas d'utilisation. J'ai essayé de rester le plus simple possible pour que vous n'ayez pas un autre programme que celui attendu .

FIN