

cned

Application gestion personnel

Compte rendu

soumia ben baha
02/06/2021

Contexte :	2
Rappel de la mission :	2
La préparation de l'environnement de travail et la création de la base de données :	3
Le maquettage et l'interface de l'application :	3
Le maquettage :	3
Le visuel de l'interface sous Visual Studio 2019	7
Configurer l'IDE pour accéder à la base de données	8
Optimisation du code :	9
Bilan :	10

Application gestion du personnel

Compte rendu :

Contexte :

Infotech service 82 est une entreprise avec 32 collaborateurs on peut séparer son activité en deux pôles développement, système et réseaux.

Le premier consiste en solutions d'hébergements sur des serveurs dédiés, les développeurs ont pour rôle le développement de logiciel de gestion, de création de base de données.

Le second consiste en conseil et audit apporte des solutions et fourniture de matériel informatiques il est aussi doté d'un centre de support et d'assistance pour ses clients.

ITS86 est la pour l'accompagnement de votre projet et s'engage a fournir une conception d'application adapté à vos besoins.

Médiattech86 se compose d'un service administratif qui gère les achats livres et matériels divers, ainsi que d'une médiation culturelle qui organise les expositions, les campagnes d'informations et de communication, et enfin d'un service de prêt.

Son rôle est de centralisé les prêts et créer une plateforme qui permet de réunir toutes les médiathèques du département, de fournir un service d'annuaire active directory, d'un compte pour chaque utilisateur

Cela permet une meilleure communication de proposer de nouvelle offre de service tel que l'emprunt de film en vod ainsi que des formations aux outils du numérique ou encore des autoformations en ligne.

C'est un portail web unique, un réseau qui regroupe les responsables de chaque médiathèque : chef de projet numérique, chargé de communication numérique, infographiste.

Tous les postes informatiques sont connectés au portail internet digimediatech86 avec accès sécurisé qui gère les prêts et les acquisitions qui permet également la réservation de document pour les prêts inter-médiathèque et l'accès aux prêts des adhérents (consultation, réservation, prolongation).

Rappel de la mission :

La création d'une application de gestion de cette entreprise au niveau des personnels (identifiant, nom, prénom, service, etc..) et la gestion des absences (date de début, date de fin et le motif d'absence).

Dans

Etape 1 : la préparation de l'environnement du travail.

Etape 2 : la création de la base de données.

Etape 3 : le maquettage et l'interface de l'application.

Etape 4 : la création des packages connexion, dal, modèle, contrôleur, et vue.

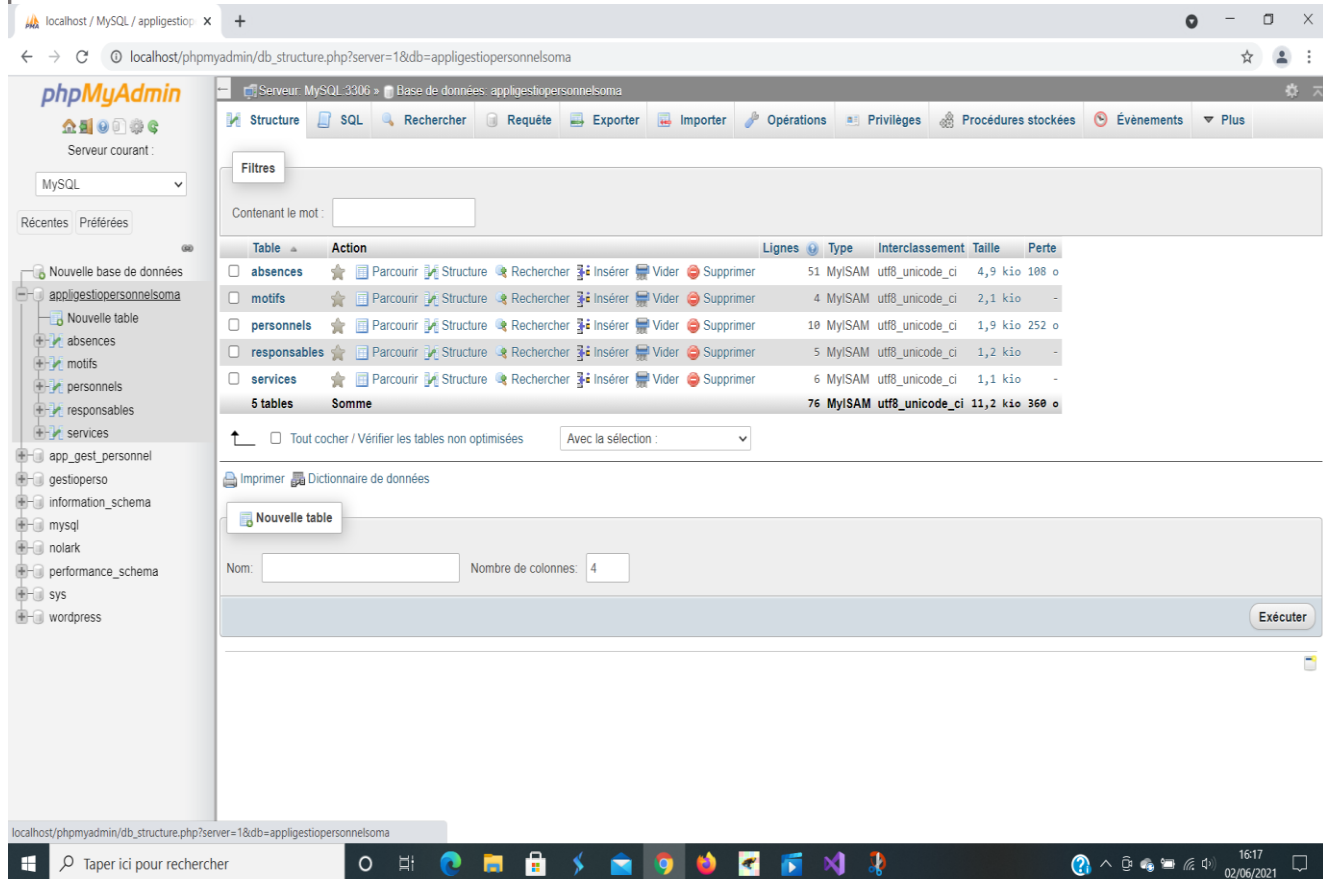
Etape 6 : modification et optimisation du code.

Etape 7 : la vidéo démonstrative.

Etape 8 : bilan.

La préparation de l'environnement de travail et la création de la base de données :

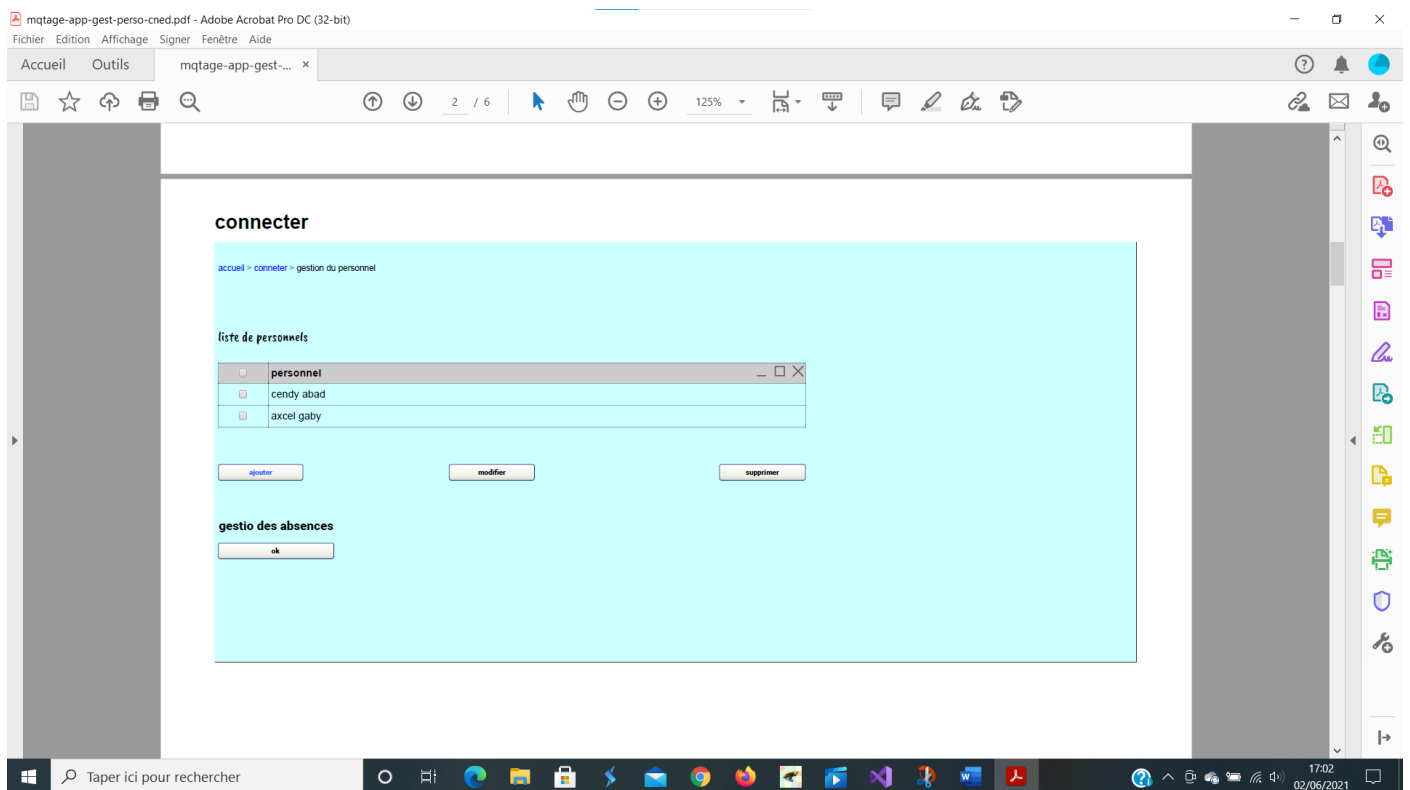
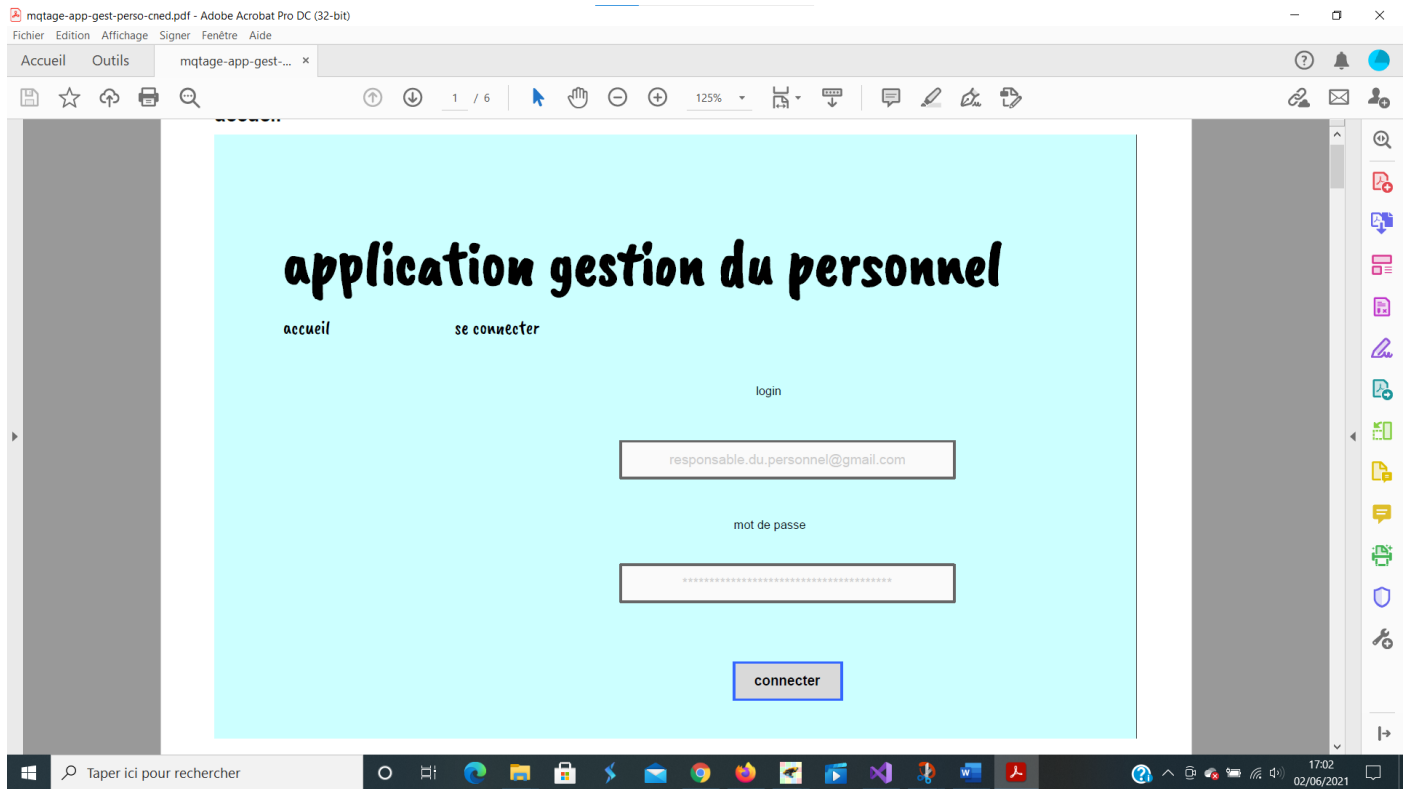
Après Avoir installé WampServer j'ai pu créer le script SQL et la base de donnée correspondants a l'application envisageable au format MySQL, après la création de l'utilisateur qui peut accéder a cette base de donner avec un mot de passe, après j'ai ajouté les différents tables a cette base de donnée (la table responsable , la table service , la table motif, la table personnel. Les tables service et motif elles ne sont pas modifiables)

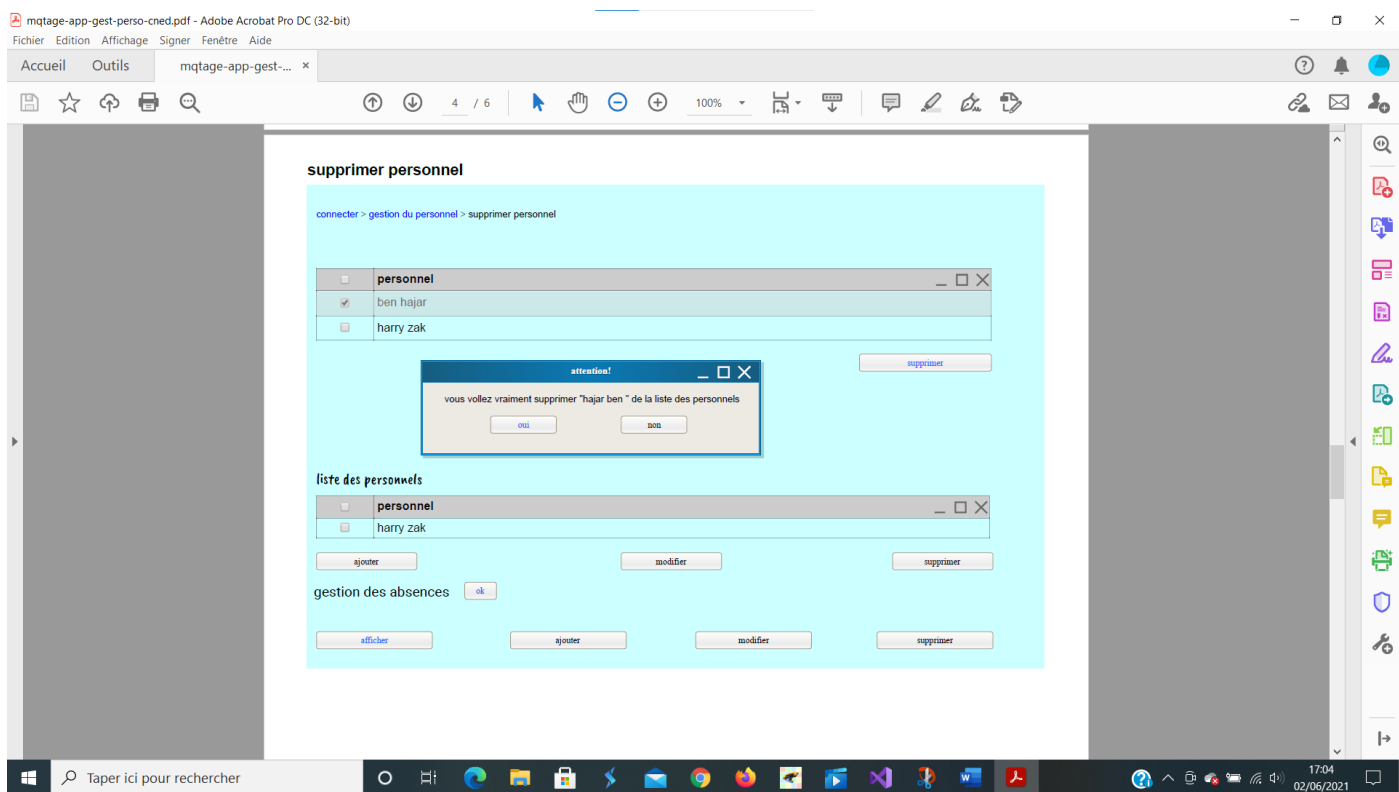
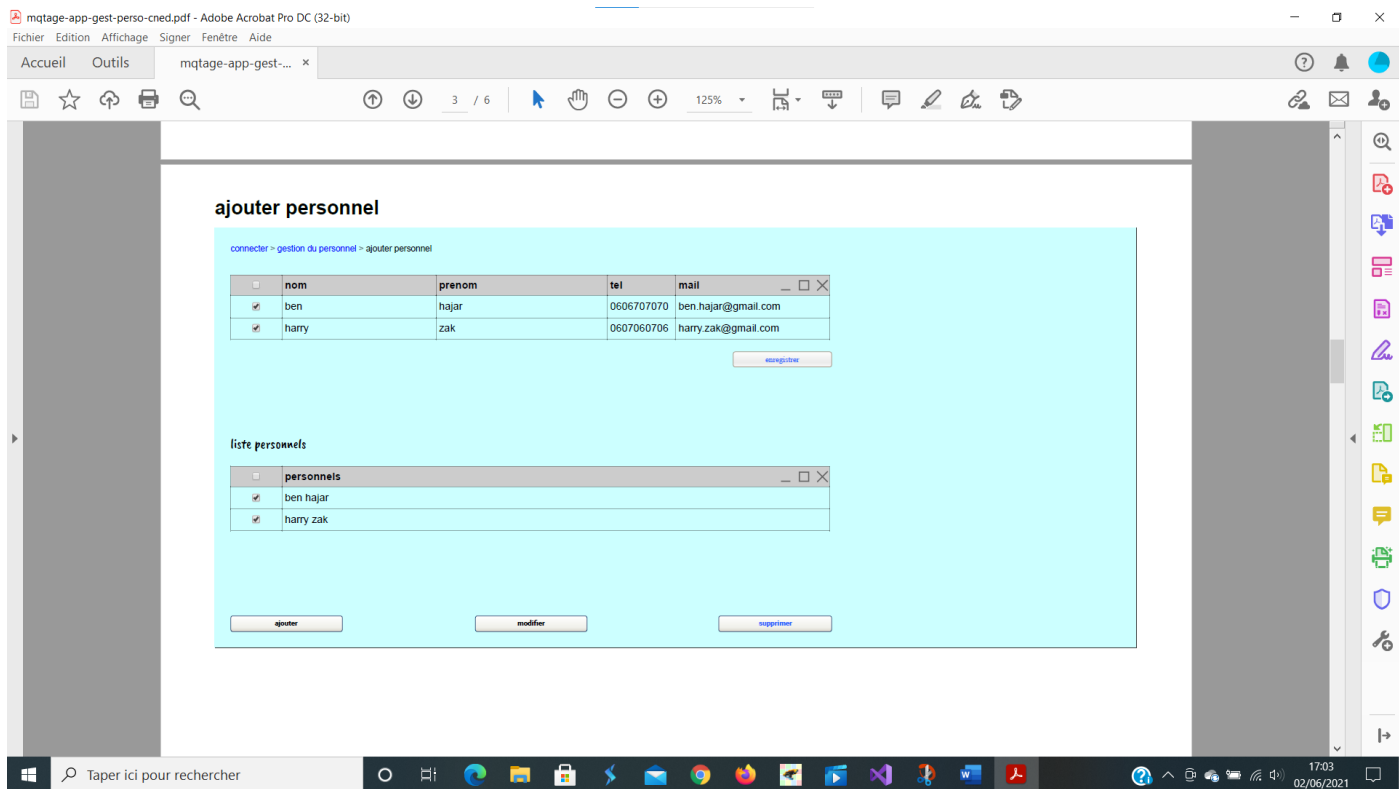


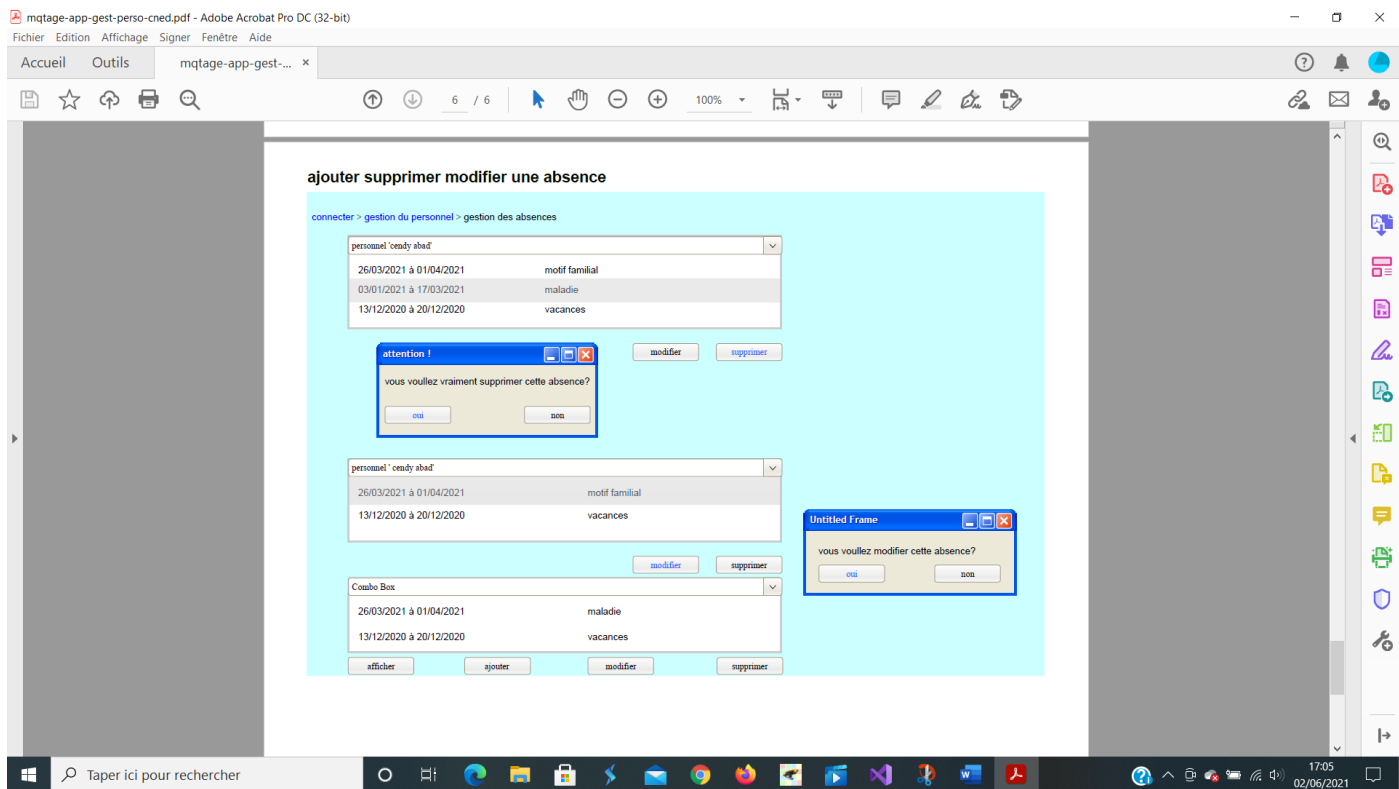
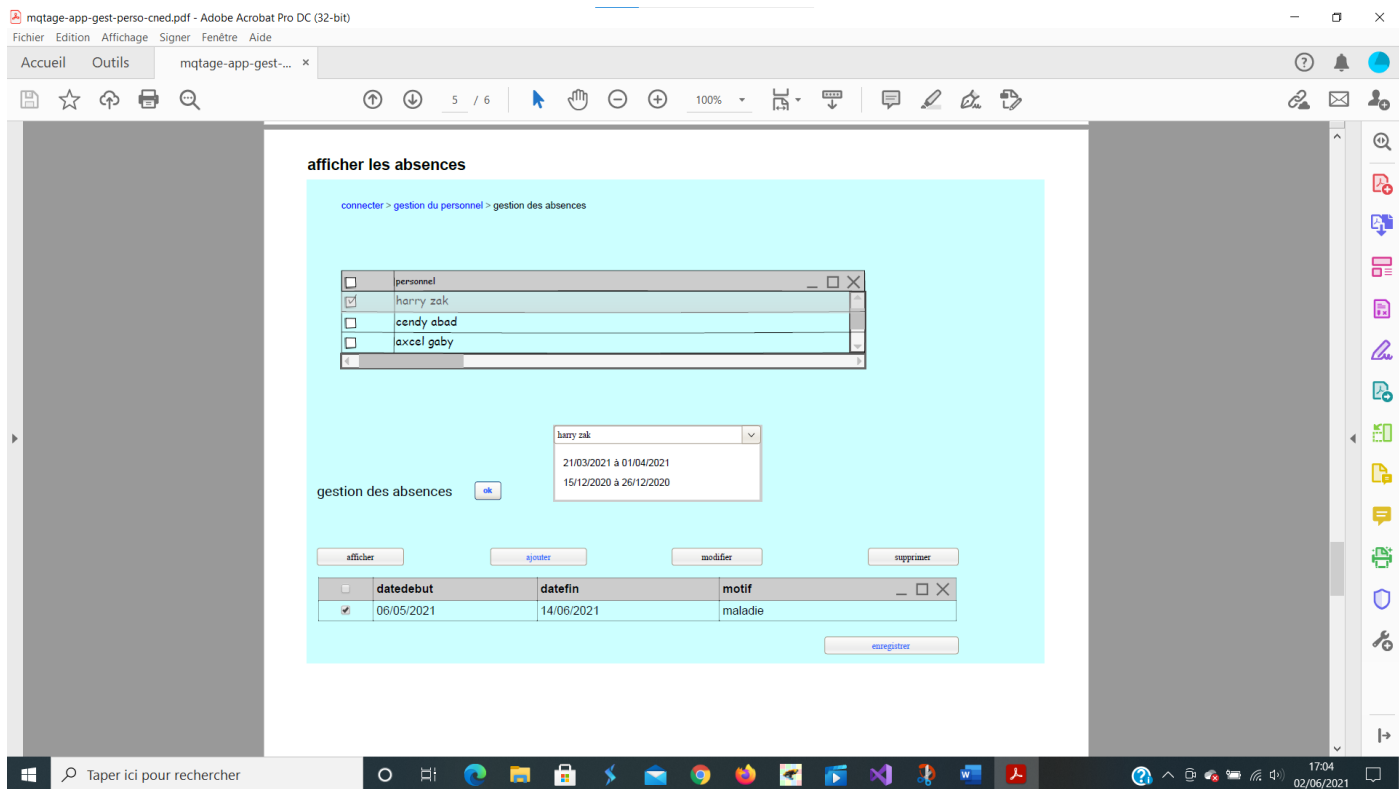
Après avoir récupérer et étudier le diagramme de cas d'utilisation et le descriptif de chaque cas d'utilisation j'ai pu faire un maquetage de l'application pour répondre aux besoins du client et l'équivalent visuel du diagramme de cas d'utilisation avec l'outil Pencil du maquetage ce qui m'a permis de coder sous Visual studio l'interface de l'application. Et la création des 3 packages correspondant au MVC.

Le maquetage et l'interface de l'application :

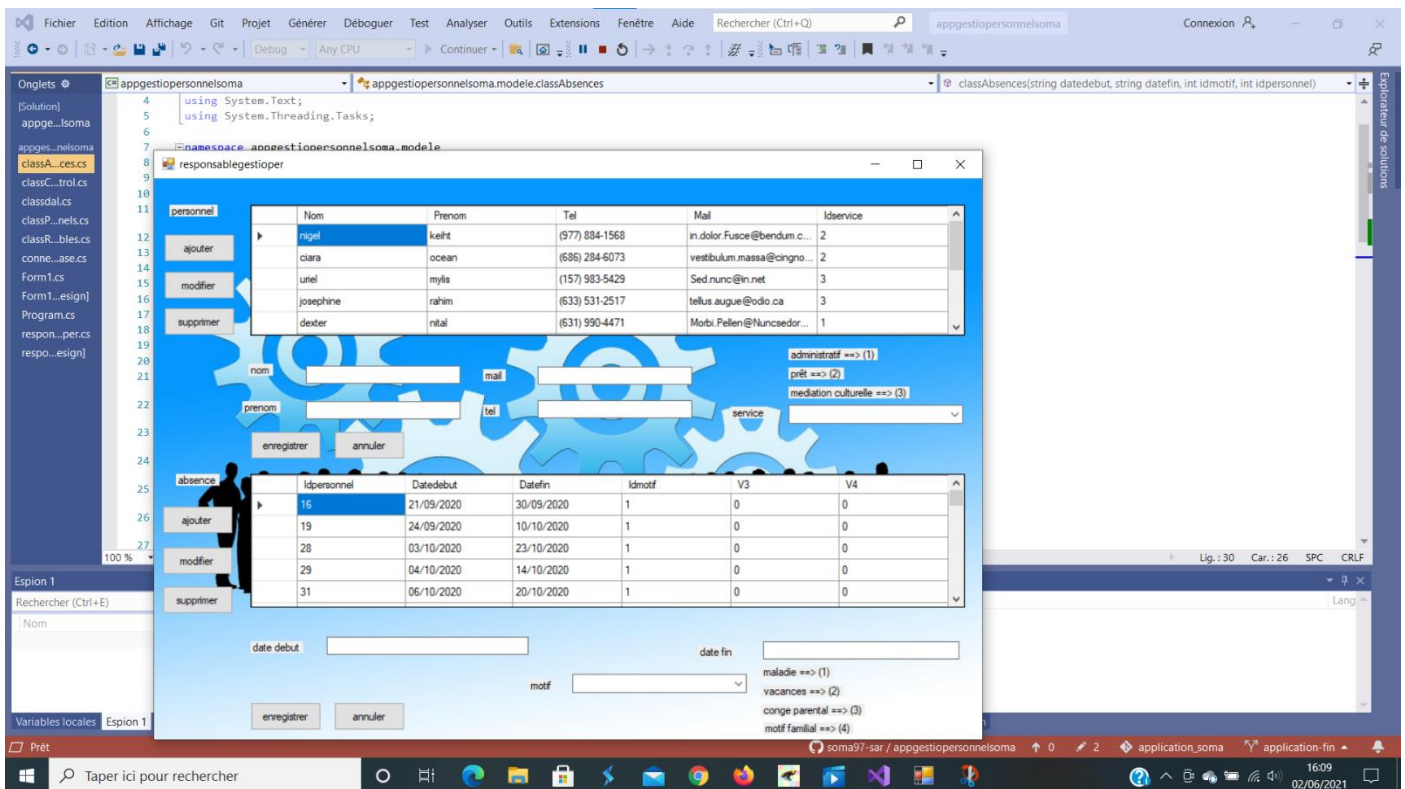
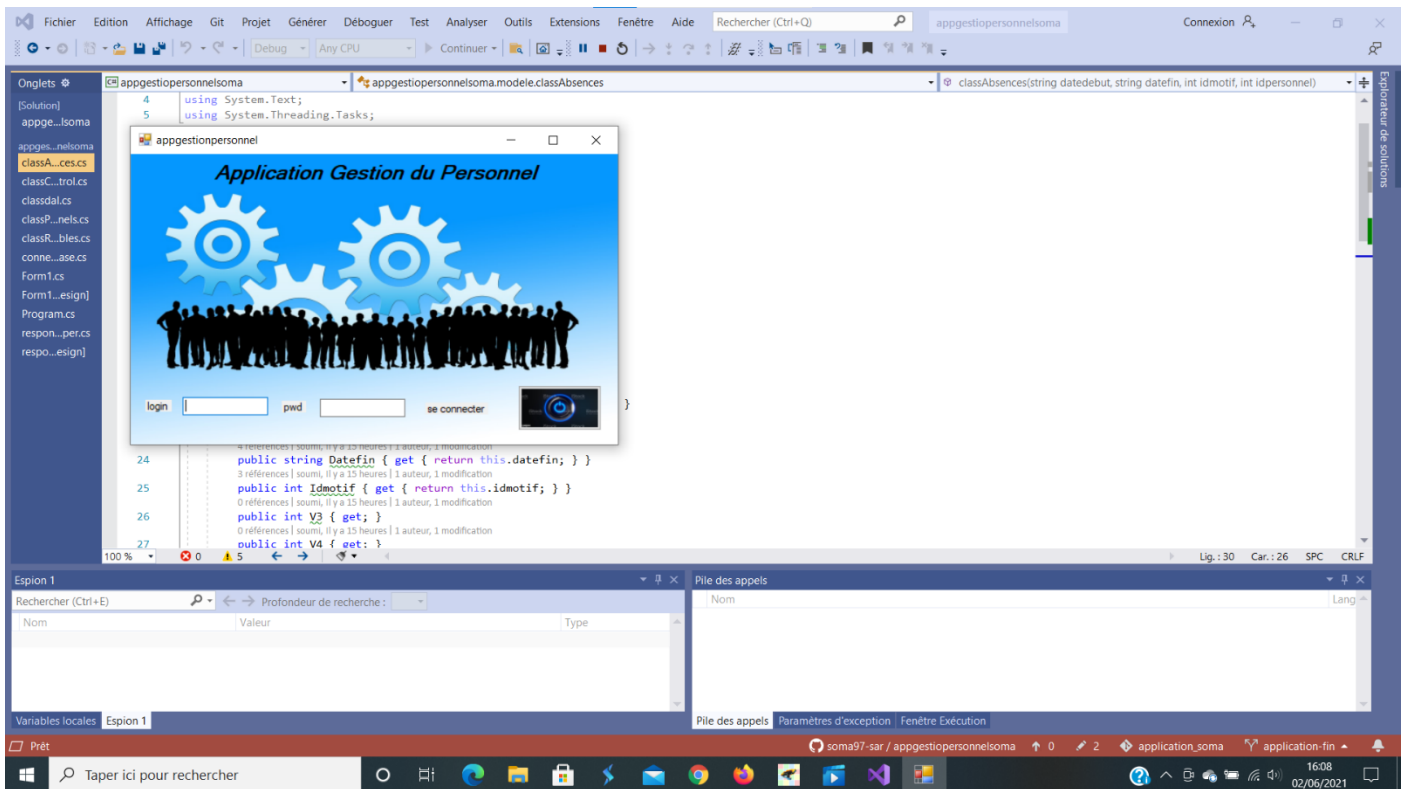
Le maquetage :



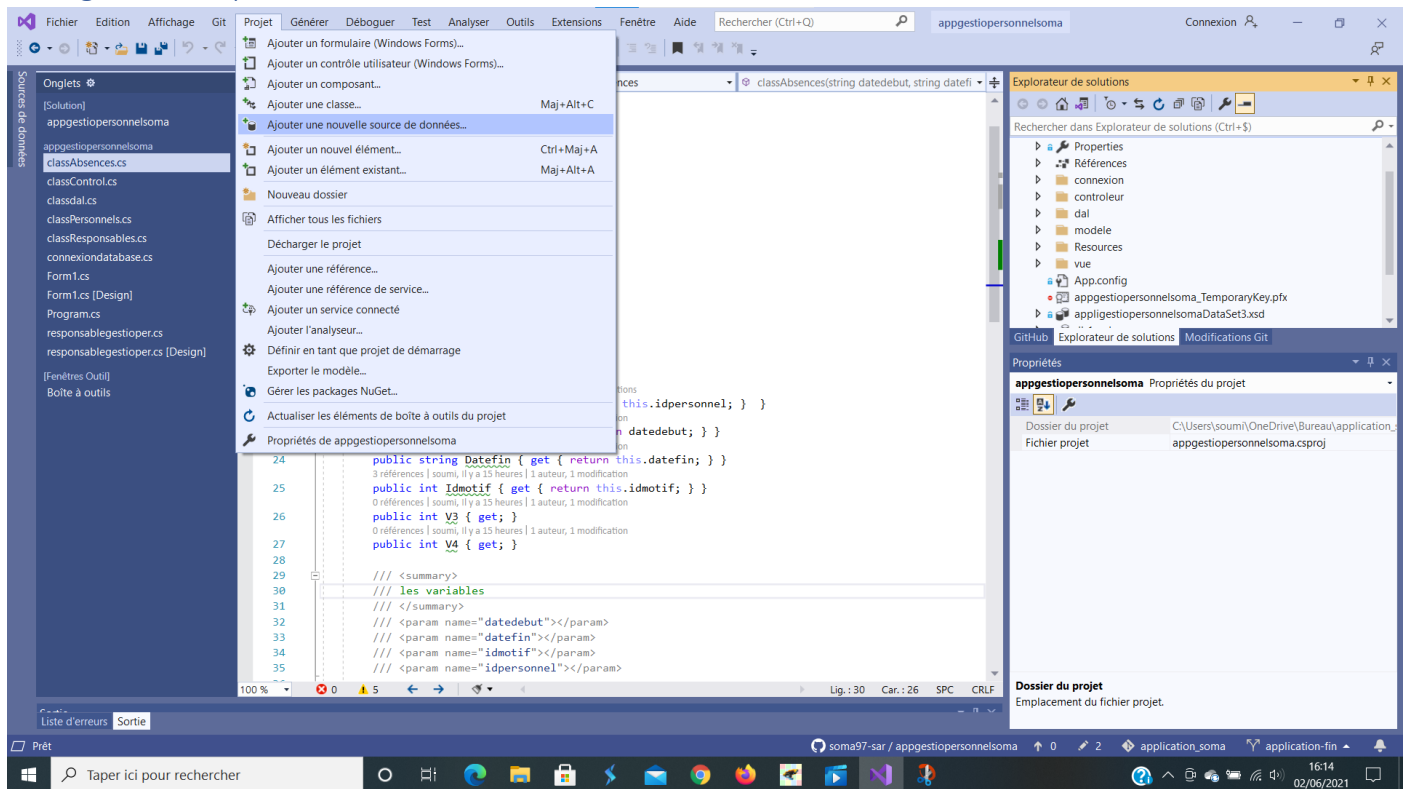




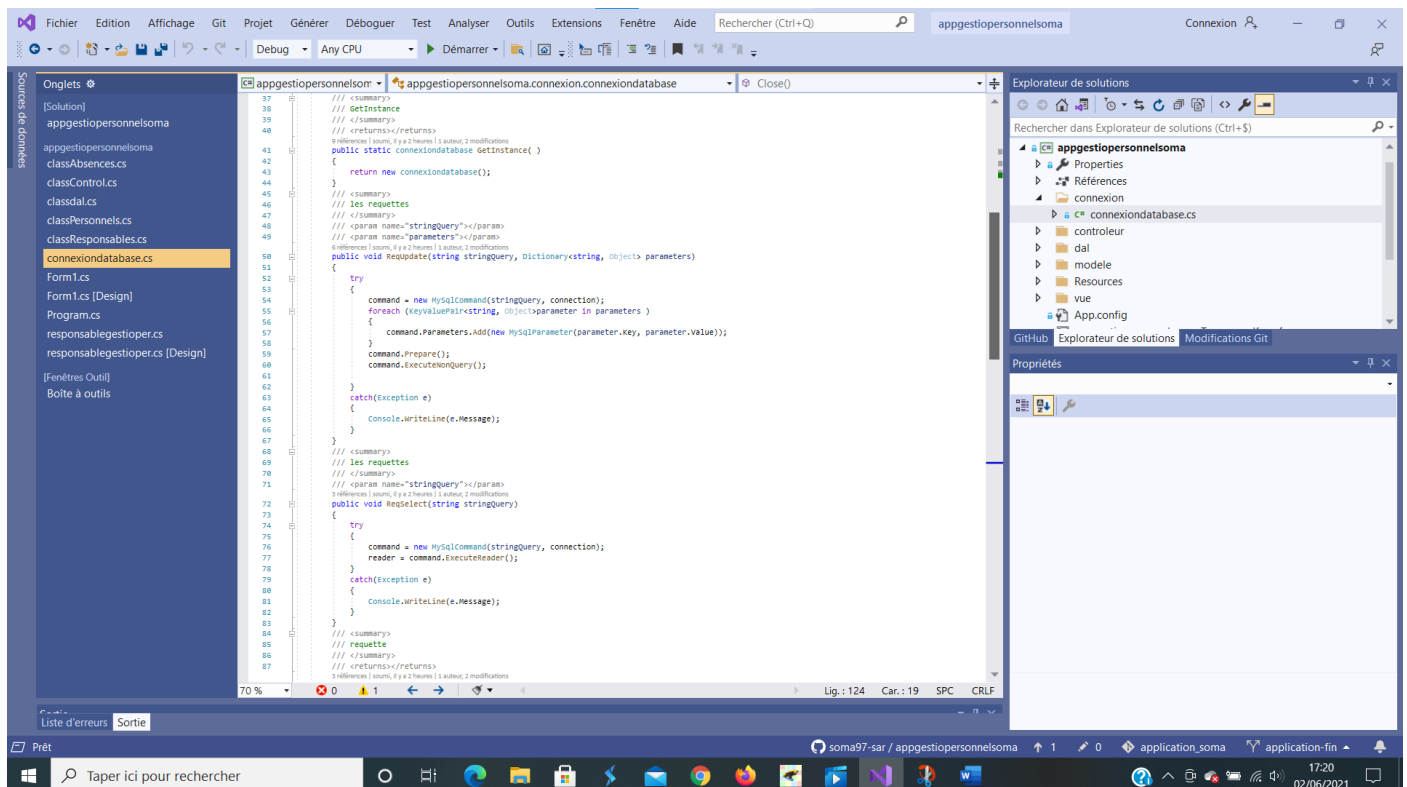
Le visuel de l'interface sous Visual Studio 2019



Configurer l'IDE pour accéder à la base de données



La création des packages:



La création du package dal qui contient une classe pour coder les différentes fonctions de l'application en résumé c'est la classe qui contient les fonctions de différentes classe dans le code comme les manipulation d'ajout de personnels ou les absences en bref c'est la classe qui relie les autres classes de code, et aussi la création du

package contrôle qui contrôle en exploitant les différentes classes du code , et dans le package modèle y’a différentes classes métiers correspondant aux tables de la base de données .

Ps : a chaque étape de ce projet il faut actualiser la documentation technique et aussi la sauvegarde sur le dépôt en ligne GitHub.

Optimisation du code :

Au niveau de cette étape il faut bien optimiser le code afin d’obtenir un code lisible et efficace :

Au niveau de la classe dal qui contient les manipulations de les class métiers qui ce trouve dans le package modèle,

Par exemple si on veut récupérer la table les personnels dans la base de donnée crée sous MySQL on peut utiliser ce code ou cette logique :

```
public static List<classPersonnels> GetPersonnels()
{
    List<classPersonnels> lesPersonnels = new List<classPersonnels>();
    string req = "select p.idpersonnels as idpersonnels,p.idservices as idservices, p.mail
as mail, p.nom as nom, p.prenom as prenom, p.tel as tel";
    req += " from personnels p join services s on (p.idservices = s.idservices) ";
    connexiondatabase curs = connexiondatabase.GetInstance();
    curs.RegSelect(req);
    while (curs.Read())
    {
        classPersonnels personnel = new classPersonnels((int)curs.Field("idpersonnels"),
(int)curs.Field("idservices"), (string)curs.Field("mail"), (string)curs.Field("nom"),
(string)curs.Field("prenom"), (string)curs.Field("tel"));
        lesPersonnels.Add(personnel);
    }
    curs.Close();
    return lesPersonnels;
}
```

Et pour avoir et retourner la table des absences on peut adopter la même méthode.

Et pour la suppression d’un personnel ou plusieurs de la table dans la base de données on peut utiliser ce code bien optimisé :

```
public static void DelPersonnels(classPersonnels personnel)
{
    string req = "delete from personnels where idpersonnels = @idpersonnels;";
    Dictionary<string, object> parameters = new Dictionary<string, object>();
    parameters.Add("@idpersonnels", personnel.Idpersonnel);
    connexiondatabase conn = connexiondatabase.GetInstance();
    conn.RegUpdate(req, parameters);
}
```

Toujours c’est la même chose pour la table des absences.

Après pour toutes les fonctions (ajouter un personnel ou une absence et modifier un personnel ou une absence) c’est la même chose et c’est presque la même logique.

Jusqu’à que le code soit bien optimisée et pratique et toutes les fonctions marche comme il faut et comme demandé et toujours dans le respect de MVC.

Le code est optimisé l’application marche et elle n’a pas des bus le but est obtenu.

La documentation de l’application en vidéo :

A ce niveau c'est demandé de réaliser une vidéo démonstrative et montrer chaque fonctionnalités de l'application (l'ajout d'un personnel, la modification d'un personnel et la suppression d'un personnel) et aussi (l'ajout d'une absence , la modification d'une absence et la suppression d'une absence) .

Bilan :

Globalement la partie code a été plus difficile pour moi sur tout au niveau de l'affichage ou bien l'addition des personnels ou des absences, ce projet me permis de mettre en pratique les connaissances que j'ai appris dans les cours, ce projet s'apparente a ce qui peut nous être demandé au milieu professionnel.