**Rapport du projet Framework**

1. **Conception de la Base de données :**
2. **Implémentation :**
3. **Fonctionnalités Admin :**

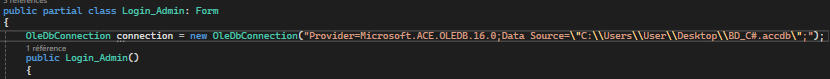
* **Login**

On a décidé que dans notre application il existe un seul admin qui a été déjà ajouter lors la création de la base de données donc pour l’admin le sign-up n’est pas disponible.

**Scénario :** Lors de la login l’Admin Tape son pseudo et son mot de passe puis Tapez sur le Button connecter, en cliquant sur le Button le système va vérifier automatiquement qu’il y a une inscription existe ou que le mot de passe est valide

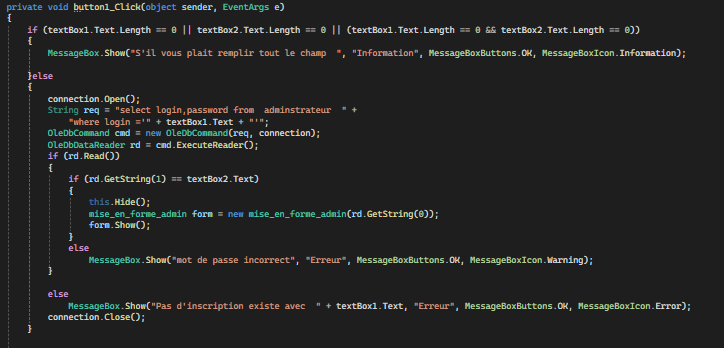
**Implémentation :**

* D’abord on a déclaré l ‘objet connexion en tant que variable globale pour éviter de le déclarer à chaque simulation d’évènement.



* On a simulé l’évènement de click sur le Button Se ***connecter***:

D’abord on va vérifier si les champs relatifs au pseudo et mot de passe sont remplie ensuite on va récupérer le mot de passe qui existe dans la base de données et qui est relative u pseudo saisie dans le textBox, si cette requête retourne un résultat (if rd. Read ()) l’admin va être orienter vers la form de gestion des utilisateurs.



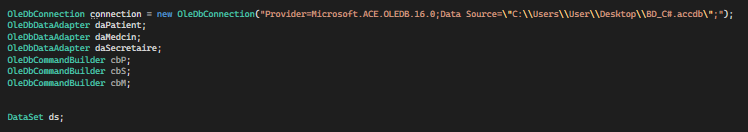
* **Le Dashboard :**
* **Gestion des utilisateurs :**

Pourles fonctionnalités de gestion d’utilisateur (ajout, modification, suppression) l’admin doit d’abord sélectionner l’utilisateur (patient, médecin ou secrétaire), après de choisir l’utilisateur une datagridview sera charger immédiatement avec son contenu représente tous les utilisateurs avec leurs cordonnées.

En effet pour réaliser les opérations de gestion de gestion des utilisateurs on a utilisé 3 form ayant on communs les caractéristiques, les événements suivants :

* **L’utilisation du mode déconnecter**

On a déclaré l’objet connexion, les dataAdapter nécessaire pour les utilisateurs du système, les commandBuilder pour exportations des données à la base ainsi que la dataset.

****

* **Chargement du formulaire :** lors du chargement des form on va remplir notre dataset avec trois tableau (médecin, patient, secrétaire) en utilisant la méthode **Fill ()** du dataAdapter, charger la comboBox avec les types d’utilisateurs et enfin remplir le comboBox2 avec les spécialités disponibles.

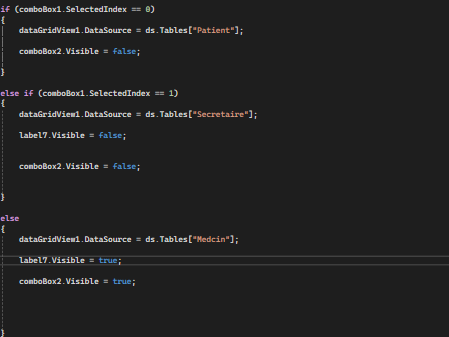
Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

* **Le choix d’utilisateur :**

du choix de l’utilisateur le comboBox du spécialité et label relative sera gérer d’une manière dynamique (ils sont affichés si l’utilisateur est un médecin)

Aussi la datagridview sera charger avec les listes des utilisateur adéquate.

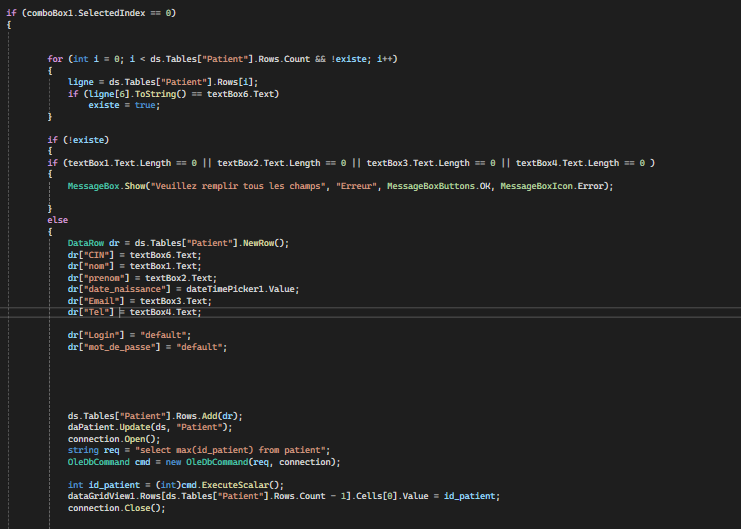


* **Ajout des utilisateurs (CrudAdmin.form) :**

**Scénario :** pour ajouter un nouvel utilisateur au système l admin doit d’abord remplir correctement les coordonnées d’utilisateur puis cliquer sur le Button ajouter.

**Implémentation :**

Par exemple lors de l’ajout d’un nouvel patient on va vérifier que tout les champ sont remplir d’une manière si oui on va vérifier que ce patient n’existe pas dans la base de données via son CIN ,si la patient existe un message d’erreur sera afficher sinon on va ajouter le patient au base via l’instruction row.add() de dataTable puis exporter les modification vers la base on utilisant la méthode update() du dataAdapter .



-le pseudo et le mot de passe sera ajouter par le patient lors de la création de son compte.

* **Problème :** l’id de patient est auto-incrémenté qui n’est pas charger directement dans la dataTable car elle est générée par le SGBD

Donc la solution consiste à récupérer l’id depuis la base de données

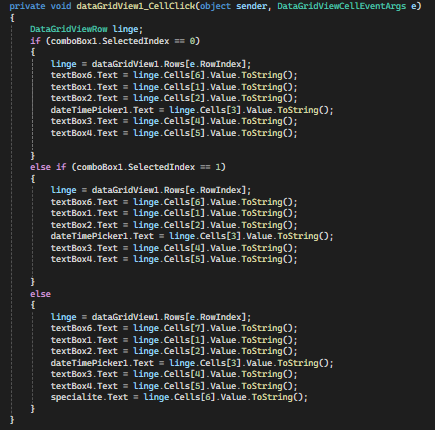
* **Suppression des utilisateurs (CrudAdminsupprimer.form) :**

**Scénario :**  pour supprimer un utilisateur l’admin doit d’abord choisit le type d’utilisateur

Puis sélectionner l’utilisateur à supprimer depuis le datagridview.

**Implémentation :**

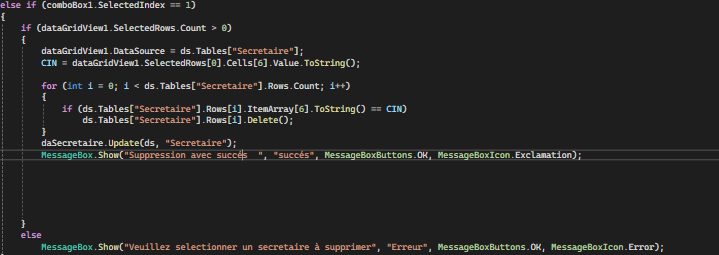
* **Le choix d‘utilisateur :** l’événement datagridview\_cellClick nous permet de récupérer la ligne sélectionner dans la datagridview car le paramètre (e) représente l’événement. Les cordonnés seront ensuite afficher dans les champ adéquate dans la forme

****

* **La suppression :**

Pour la suppression on va d’abord vérifier si une ligne de la datagridview est déjà sélectionné (via l’instruction **selectedRow.count**) ou non si non un message d’erreur sera affiché sinon :

On va récupérer d’abord le CIN, chercher dans la datable l’utilisateur avec ce CIN, supprimer l’utilisateur et enfin exporter les modifications vers la base.



* **Modification des utilisateurs (CrudAdminModif.form) :**

**Scénario :**  pour modifier un utilisateur l’admin doit d’abord choisit le type d’utilisateur

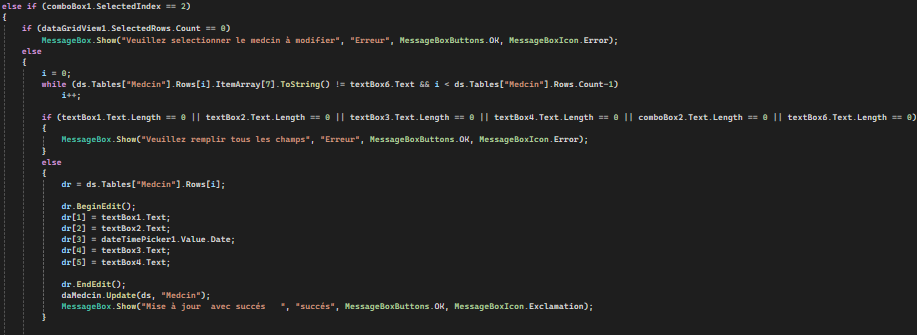
Puis sélectionner l’utilisateur à modifier depuis le datagridview.

L’admin peut modifier toutes les coordonnées à l’exception de la CIN, pseudo et pwrd.

**Implémentation :**

* **Choix de l’utilisateur :** on a suivi la même démarche que la suppression.
* **Modification :** lors de la clique sur la Button modifier on va d’abord récupérer la ligne sélectionner de la datagridview puis vérifier si existe dans la data Table relative à l’utilisateur si oui on va enregistrer cette ligne dans un data row temporaire changer les coordonnées puis exporter les modifications vers la BD.

Sinon un message d’erreur sera affiché.

****

* **Gestion de disponibilité du médecin :**

**Scénario :**  administrateur commence par sélectionner un médecin dans une **ComboBox**. Une fois le médecin sélectionné, ses **informations personnelles** (nom et prénom) ainsi que ses **horaires de travail** (heures de début et de fin) pour chaque jour de la semaine s’affichent automatiquement

Pour **modifier la disponibilité** du médecin sur un jour spécifique :

1. L’administrateur sélectionne la ligne correspondant au jour souhaité dans la DataGridView.
2. Il saisit les **nouveaux horaires** (heure de début et heure de fin)
3. Il valide les modifications en cliquant sur un **bouton "Enregistrer"**.

Si le médecin **n’est pas disponible** un jour donné, ses horaires pour ce jour sont affichés par défaut à **00:00 - 00:00** .

**Implémentation :**

Lors de chargement de la forme on va récupérer tous les médecins dans notre BD par leur ID et les charger dans le comboBox.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.

Ensuite selon le médecin sélectionner on va récupérer son nom et prénom et le charger dans les textbox relative aussi on va charger la datagridview les jour de travail de ce médecin.

Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

Enfin lors de la clique sur le Button modifier on va d’abord vérifier qu’un médecin est déjà sélectionné, et une ligne de la datagridview est aussi sélectionnée si oui on va lancer une requête update pour modifier les heures de début et fin de ce jour et de ce médecin Aussi on va appliquer la modification sur la datagridview

Sinon un message d’erreur sera affiché.

Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

1. **Fonctionnalités Médecin :**

* **Login & sign-up :**

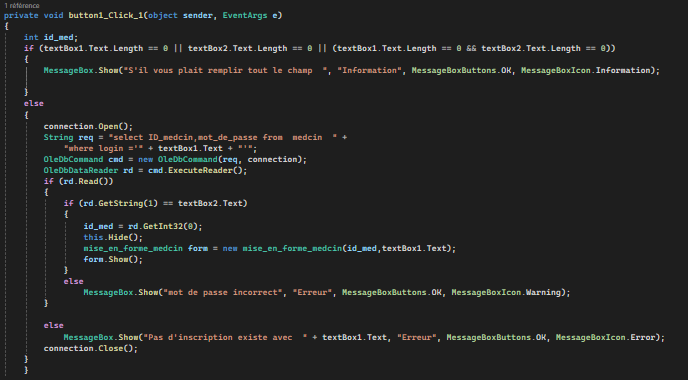
**Scénario :**

Lors de la login le médecin tape son pseudo et mot de passe pour connecter à son interface.

Si le médecin ne dispose pas une inscription alors il faut créer un compte et pour effectuer cette action le médecin doit tapez son adresse électronique saisie un pseudo et un mot de passe.

**Implémentation :**

Pour le login on va sélectionner depuis la BD le mot de passe selon le pseudo tapez si aucune information ne sera retourne un message d’erreur sera affiche e l’utilisateur doit créer un compte sinon le médecin va connectez à son interface .

****

* Pour le sign up le médecin doit tapez son mail et remplir les champs nécessaires ensuite on vérifier si le médecin existe dans la base à travers son mail si oui le médecin peut créer un compte sinon un message d’erreur sera affiché .

Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

* **Gestion des rendez-vous :**

**Scénario :** après la connexion une emploi de temp sera affiché au médecin contient tous les rendez-vous d’aujourd’hui en effet le médecin doit sélectionné un rendez-vous pour le gérer c’est un dire ajouter des dossiers et des document médicales les historiques des rendez-vous avec un patient ..

**Implémentation :**

* **Emploi du temp(emploimedcin.cs) :**

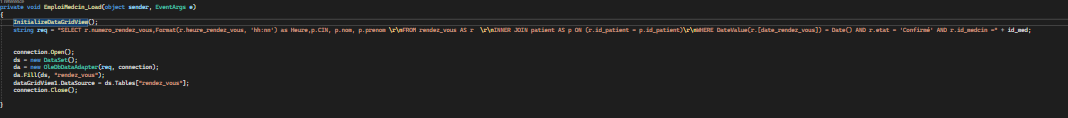
Pour charger les rendez-vous d’un médecin, une requête est envoyée à la base de données afin de récupérer les **informations suivantes** :

* Numéro de rendez-vous
* Heure du rendez-vous (au format HH:mm)
* CIN, nom et prénom du patient

Les données sont filtrées selon les **critères suivants** :

* Le rendez-vous est associé à l’**ID du médecin sélectionné**
* La **date du rendez-vous est aujourd’hui**
* Le **statut du rendez-vous est "confirmé"**

Ensuite ces donnez sont enregistrer dans une datagridview.

****

Ensuite selon le rendez-vous sélectionner le médecin va orienter vers la page de gestion des rendez-vous toute en conservant id de rendez-vous. (Attribut de la forme suivante)

* **Gestion des rendez-vous (Gérer\_rendez-vous.cs) :**

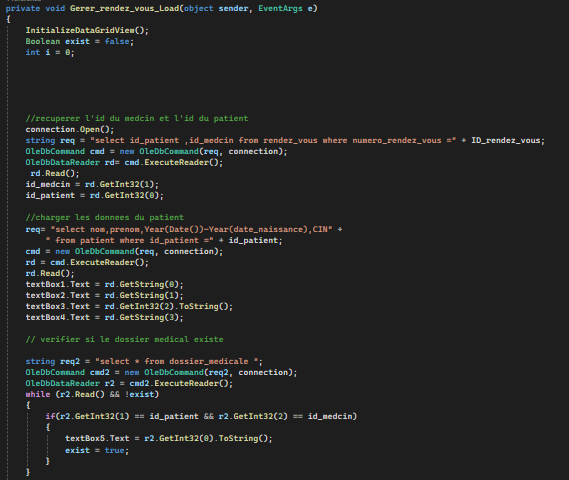
**Scénario :** lors de chargement de la forme les informations du patient seront affichées ainsi qui le numéro de son dossier médical si existe sinon le médecin peut créer un dossier médical pour ce patient. Aussi l’historique de rendez-vous sera affiché et pour ajouter un document médical le médecin peut cliquer sur un Button pour aller vers la page de gestion des dossiers médicale.

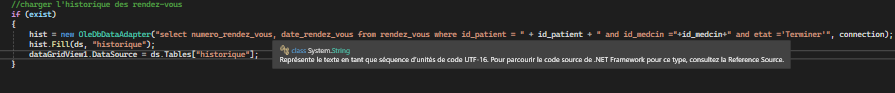
**Implémentation :**

D’abord lors du chargement de la forme on va récupérer l’id de médecin et ID de patient selon le rendez vous choisi ensuite on va récupérer certaine information relative au patient et les charger dans les champs relatifs dans la forme.

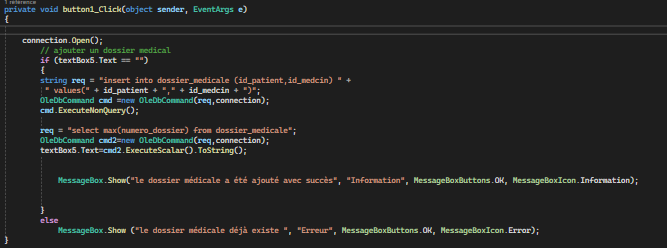
Ensuite on va vérifier si ce patient a un dossier médical si oui on va charger le textbox relative avec le numéro de rendez-vous sinon rien faire

En effet si le dossier médical existe cela signifie qui il y a au moins un rendez-vous archive donc on va récupérer tout les rendez-vous archivé (état = terminer) et les charger dans une datagridview.

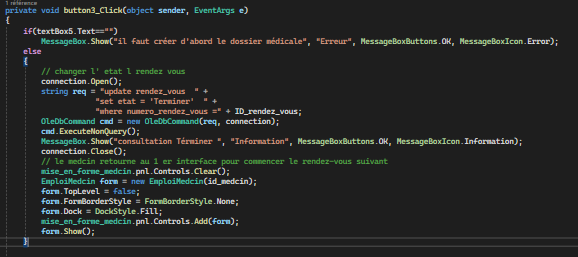




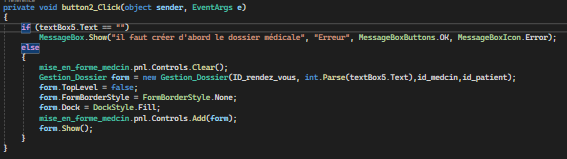
En effet si le dossier médicale n’existe pas (textBox= ‘’) on va le créer ceci à travers une requête insert pour ajouter ce dossier sinon un message sera affiché indiquant que le dossier existe .



Et pour terminer une consultation le médecin doit au moins créer le dossier médicale si il n’existe pas pour la terminer et ensuite on va mettre à jour l’état de ce rendez-vous ( devient terminer) et le médecin retourne à la liste des rendez-vous.



Et enfin pour gérer les document médicale on doit d’abord vérifier si le dossier existe ensuite le médecin sera orienter vers la page de gestion des document.

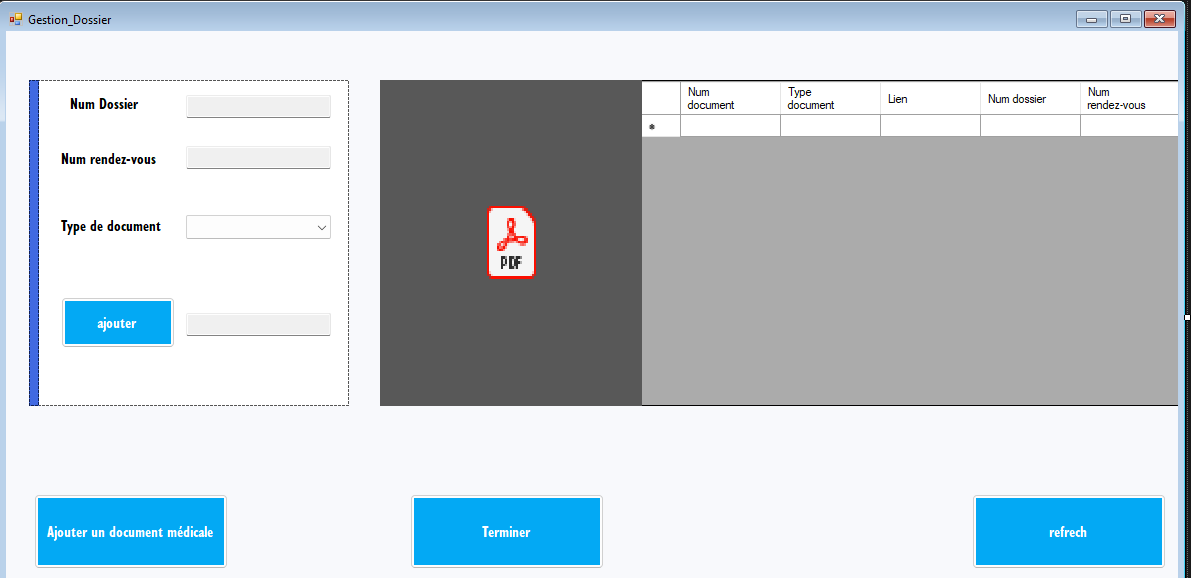


* **Gestion de Dossier médicale (Gérer\_rendez-vous.cs) :**

**Scénario :**  cette interface permet au médecin d’ajouter des documents médicaux au dossier du patient et de lister tous les documents médicaux générer dans des consultations précèdent.

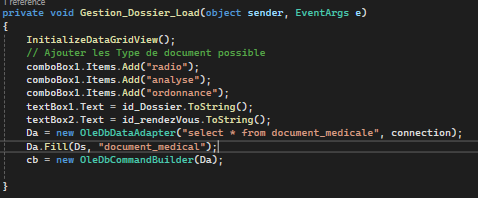
Le Button refrech permet de charger et mise a jour la liste des documents médicaux , le Button ajouter permet de sélectionner un fichier et le Button Ajouer document médicale permet d’ajouter le document sélectionner dans le dossier du patient

On a utiliser un pdfReader pour visualiser les documents sélectionner



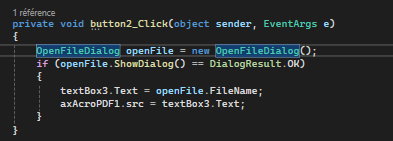
**Implémentation :**

D’bord lors du chargement du form on va afficher le numéro de dossier et le numéro de rendez vous aussi on va récupérer tout le document médical appartient au ce dossier

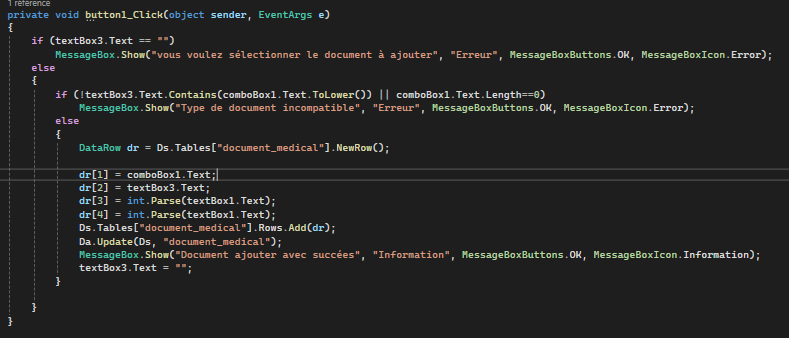


On va utiliser le mode déconnecter pour ajouter des nouveaux documents aussi on a ajouté des attributs pour la form comme id dossier et id rendez-vous vient de la form gestion des rendez-vous

Et on a chargé notre comboBox avec les types des documents possible.

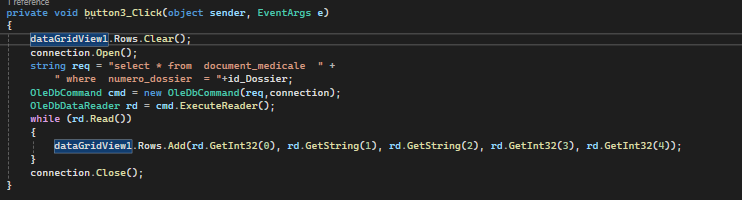


Ensuite pour le Button de sélection des documents on a ouvrir une nouvelle fenêtre de sélection des fichier on a attribuer le schéma de fichier sélectionner au textbox3 et en fin charger notre PDF Reader avec ce fichier pour l’affichage.



Et pour ajouter le document au dossier correspondant on va d’abord vérifier qu’un document est déjà sélectionner si oui on va vérifier que le type et compatible avec le document choisi si oui on va ajouter le document au data Table puis exporter les modifications vers la base de données.

Et en fin pour le Button refrech on va récupérer tous les documents archivés dans le dossier médical et les afficher dans une datagridview



Et lors de la sélection d’un document depuis datagridview il sera affiché dans notre PDF Reader

