

Année 2022-2023

Dossier de projet



# Gestion de Stock

# Sommaire

I. PRESENTATION DE LA PARTIE INDIVIDUELLE .....	4
1. MON ROLE DANS LE PROJET .....	4
2. CAS D'UTILISATION .....	4
3. BASE DE DONNEES .....	5
4. OUTILS UTILISES.....	6
A. Logiciels .....	6
B. Languages de programmation.....	6
C. Composants.....	6
5. ANALYSE UML : DIAGRAMME DE SEQUENCE.....	8
A. Diagramme du cas d'utilisation « Gérer un matériel » : Liste .....	8
B. Diagramme du cas d'utilisation « Gérer un matériel » : Ajouter .....	9
C. Diagramme du cas d'utilisation « Gérer un matériel » : Modifier .....	10
D. Diagramme du cas d'utilisation « Gérer un matériel » : Supprimer .....	11
E. Diagramme du cas d'utilisation « Générer un QrCode » .....	12
F. Diagramme du cas d'utilisation « S'identifier ».....	13
G. Diagramme de classes .....	14
6. CONCEPTION .....	15
A. Classe GestionStockBDD_Connexion et GestionStockBDD .....	15
B. Classe QrCode.....	16
C. Classe UnMateriel, UnEmploye et UnClient.....	17
7. LES DIFFERENTES VUES.....	18
A. A l'ouverture de l'application .....	18
B. Onglet des paramètres .....	18
C. Liste des matériels .....	19
D. Ajouter un matériel .....	19
E. Modifier un matériel .....	20
F. Supprimer un matériel .....	20
G. Génération du QrCode .....	21
H. Liste des clients.....	21
I. Ajouter un client.....	22
J. Liste des employés .....	22
K. Ajouter un employé.....	23
L. Reset password employé .....	23
8. POINT ESSENTIEL DE MON ETUDE: «S'identifier» .....	24

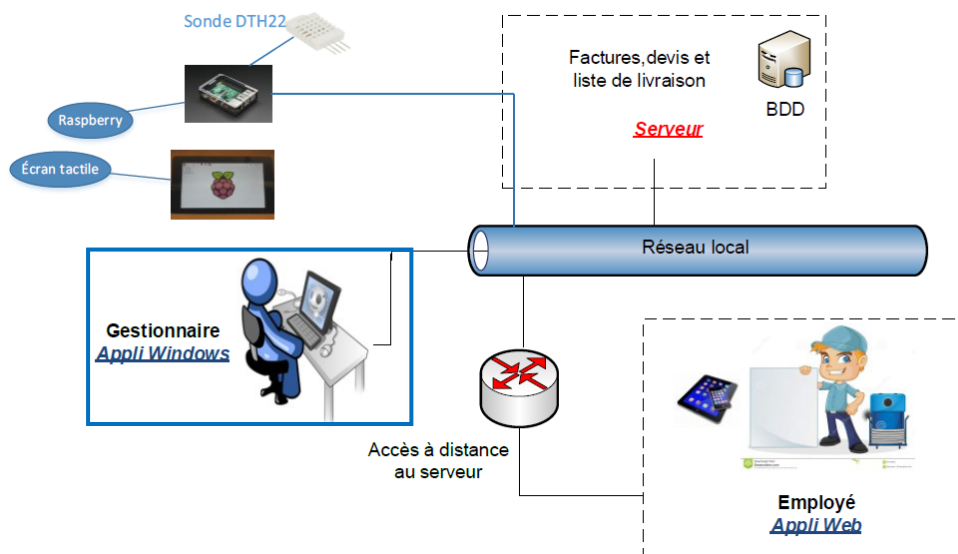
---

A.	Lancement de l'application .....	24
B.	Les différentes méthodes pour la connexion.....	25
C.	Les différentes vues de connexion .....	30
II.	Conclusion .....	31

# I. PRESENTATION DE LA PARTIE INDIVIDUELLE

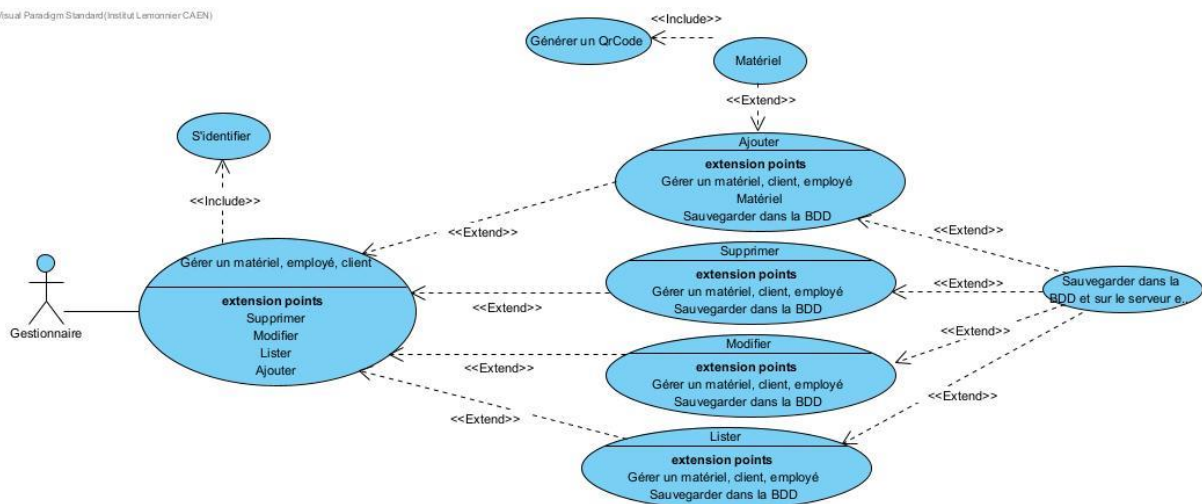
## 1. MON ROLE DANS LE PROJET

Dans ce projet, j'ai pour rôle de réaliser une partie de l'application Windows. Dans cette partie, je gère les matériels, les clients et les employés. Je m'occupe aussi de la création d'un QR Code pour chaque matériel afin de les identifier.



## 2. CAS D'UTILISATION

Visual Paradigm Standard (Institut Lemonnier CAEN)



### Gestionnaire & Employé

**« S'identifier » :**

**Description :** Pour pouvoir gérer un employé, un client ou un matériel, le gestionnaire doit s'identifier à l'application.

**Gestionnaire****« Gérer un matériel » :**

**Description :** Ce cas d'utilisation permet au gestionnaire d'ajouter, modifier ou supprimer un matériel de la base de données. Ce cas inclus le cas générer un QrCode, qui permet d'associer un QRCode à un matériel pour l'identifier.

**« Gérer un client » :**

**Description :** Ce cas d'utilisation permet au gestionnaire d'ajouter, modifier ou supprimer un client de la base de données.

**« Gérer un employé » :**

**Description :** Ce cas d'utilisation permet au gestionnaire d'ajouter, modifier ou supprimer un employé de la base de données.

**« Générer un QrCode » :**

**Description :** Pour pouvoir identifier chaque matériel nous avons besoin de créer un QrCode unique pour chacun.

### 3. BASE DE DONNEES

Pour ma partie dans ce projet, j'utilise principalement ces 3 tables dans la base de données :

materiels	employes	clients
id_materiel INT(11)	id_employe INT(11)	id_client INT(11)
reference VARCHAR(45)	nom_employe VARCHAR(45)	nom_client VARCHAR(45)
categorie VARCHAR(45)	prenom_employe VARCHAR(45)	prenom_client VARCHAR(45)
caracteristique VARCHAR(200)	email_employe VARCHAR(200)	adresse_client VARCHAR(50)
qrcode VARCHAR(45)	telephone_employe CHAR(10)	code_postale_client CHAR(5)
prix_ht DOUBLE	permission TINYINT(1)	ville_client VARCHAR(45)
	identifiant VARCHAR(45)	telephone_client CHAR(10)
	password CHAR(64)	mail_client VARCHAR(45)
Indexes	Indexes	Indexes

## 4. OUTILS UTILISES

### A. Logiciels



**Visual Studio** est un environnement de développement intégré (IDE) de Microsoft pour Windows. Il permet aux développeurs de concevoir, développer, déboguer et publier des applications pour différentes plates-formes, y compris Windows, web, cloud et mobile.



**Visual Paradigm** est un logiciel de modélisation UML (Unified Modeling Language) pour le développement de logiciels. Il permet de concevoir et de documenter les systèmes en utilisant des diagrammes UML tels que les diagrammes de classe, de cas d'utilisation et de séquence.



**GitHub Desktop** est une application de bureau qui facilite la gestion de projet de développement sur la plateforme GitHub. Elle permet de cloner, gérer les branches et les fusions de modifications en utilisant une interface graphique simple et intuitive.

### B. Languages de programmation

**MySQL My Structured Query Language**, est un système de gestion de base de données pour stocker et enregistrer des données que l'on peut associer avec d'autres langages pour effectuer des requêtes SQL.

**C#** est un langage de programmation orienté objet conçu pour le développement d'applications Windows. C# est utilisé pour développer des applications de bureau, des applications web, des jeux et d'autres applications informatiques.

**WPF** est une plateforme de développement d'interface utilisateur pour Windows. Il permet aux développeurs de créer des applications riches en graphiques et en animation en utilisant un modèle de développement basé sur XML. Il permet également une personnalisation avancée de l'interface utilisateur en utilisant des styles et des modèles.

### C. Composants

**MySql.Data** est un connecteur MySQL/.NET fourni par Oracle. Elle permet de se connecter et d'interagir avec une base de données MySQL depuis une application .NET. Avec cette bibliothèque, nous pouvons facilement effectuer des opérations sur une base de données MySQL telles que l'insertion, la mise à jour et la récupération de données.

**iText7** est une bibliothèque de traitement de PDF pour .NET. Elle permet de générer des fichiers PDF, ajouter ou modifier du texte, des images et d'autres éléments dans des documents PDF.

**QRCoder** est une bibliothèque de génération de codes QR pour .NET. Elle permet de générer facilement des codes QR de manière programmatique

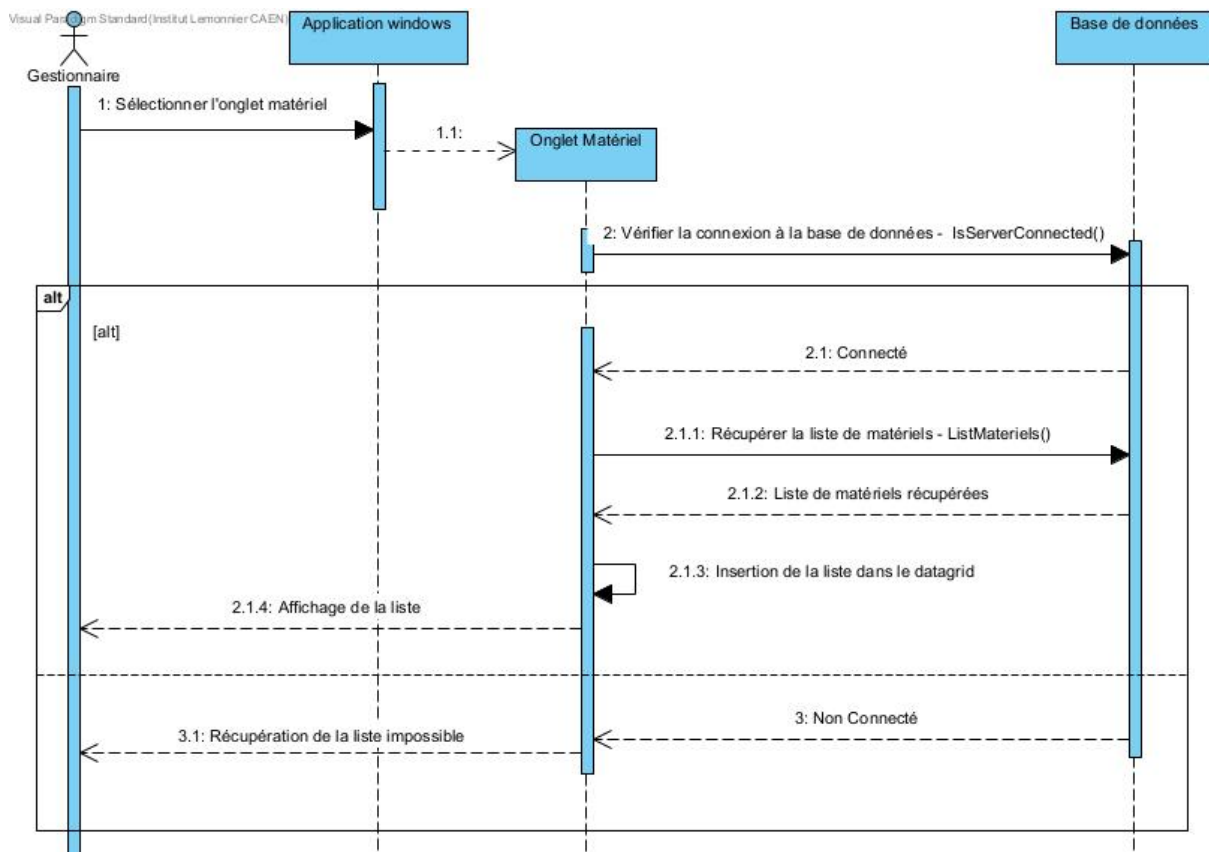
**QrCode** (abréviation de "Quick Response code") est un type de code-barres en deux dimensions qui peut stocker de l'information. Il est constitué de modules carrés noirs et blancs disposés sur un fond blanc. Les QR codes peuvent stocker différents types de données, comme des adresses URL, des numéros de téléphone, des informations de contact, des identifiants de produit, des messages texte, des images, et bien plus encore.

Les QR codes sont scannés simplement avec la caméra d'un smartphone. Les informations contenues dans le code sont ensuite interprétées par l'application pour afficher le contenu correspondant



## 5. ANALYSE UML : DIAGRAMME DE SEQUENCE

### A. Diagramme du cas d'utilisation « Gérer un matériel » : Liste



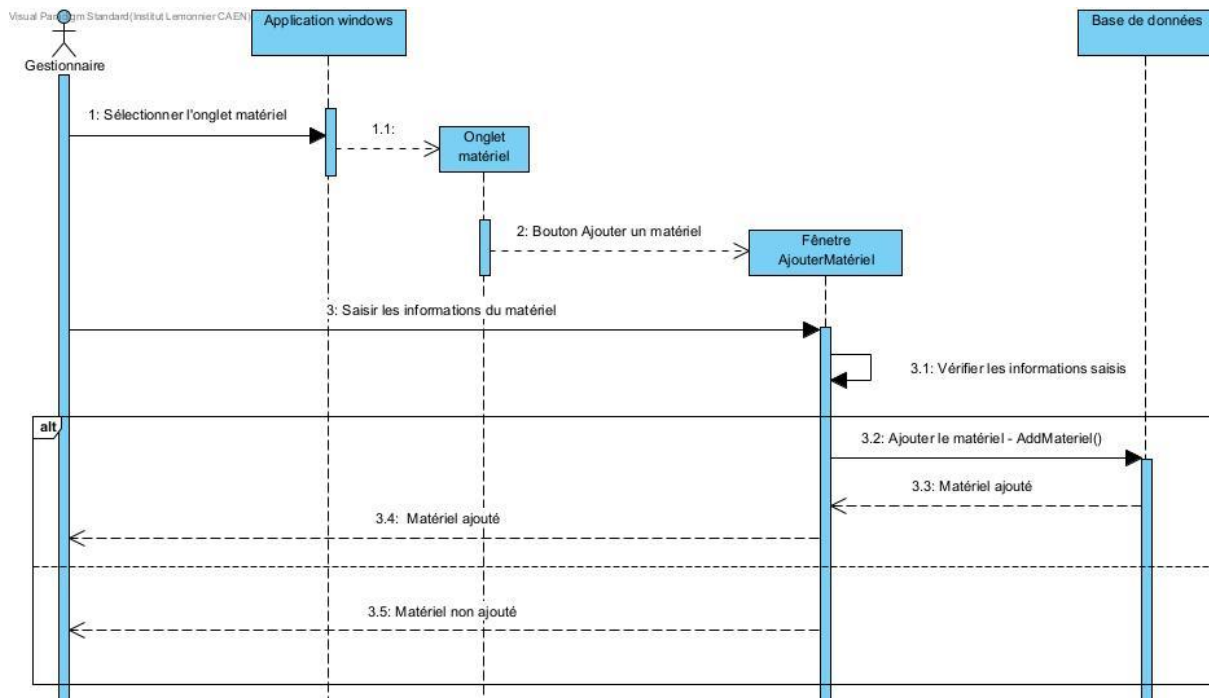
Le gestionnaire clique sur l'onglet « Matériel », ce qui a pour effet de lancer la méthode « **IsServerConnected()** » qui réalise une requête à la base de données pour vérifier la connexion.

- Si la connexion réussit : L'application récupère la liste des matériels avec la méthode « **ListMateriels()** » et l'insère dans un datagrid.
- Si la connexion échoue : L'application affiche que la récupération de la liste a échoué.

**Remarque :** les diagrammes de séquence « Gérer un Client » et « Gérer un Employé » sont similaires au diagramme ci-dessus.



## B. Diagramme du cas d'utilisation « Gérer un matériel » : Ajouter



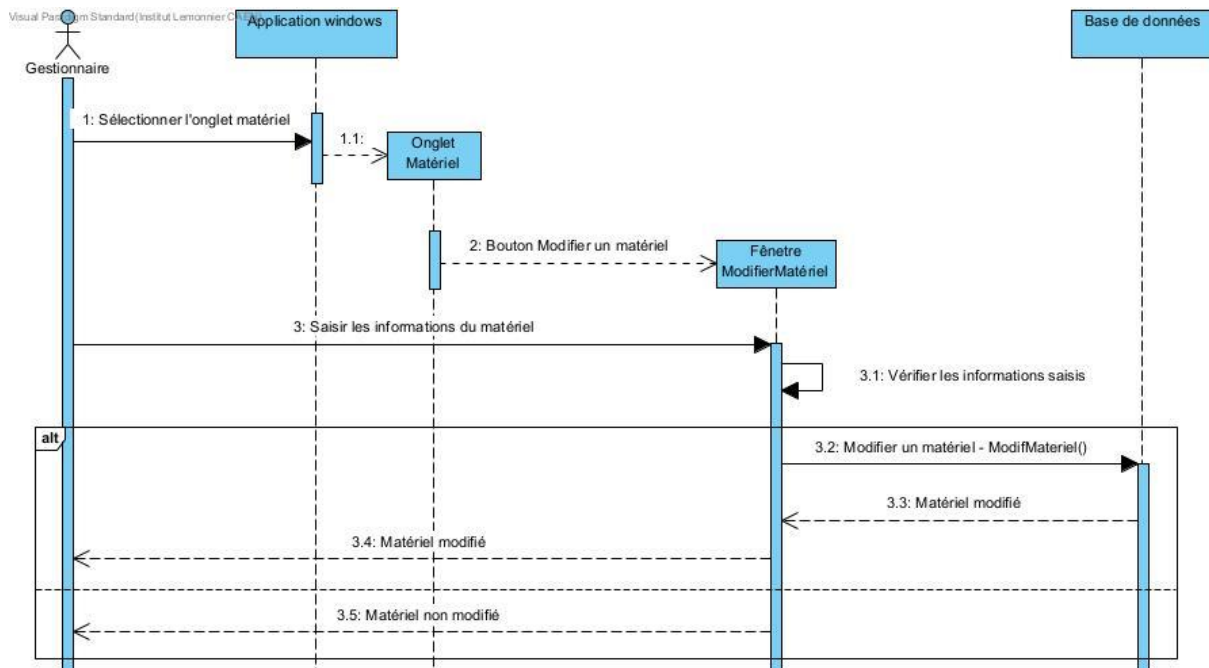
Le gestionnaire peut cliquer sur le bouton « Ajouter » de la page « Matériel », la fenêtre « AjouterMatériel » s'ouvre. L'employé saisit les informations du matériel qui sont vérifiées.

- Si elles sont valides, on passe en paramètre les informations à la méthode « **AddMateriel()** » : le matériel est ajouté.
- Si elles ne sont pas valides : le matériel n'est pas ajouté

Un message est affiché pour informer l'utilisateur si le matériel est ajouté ou pas.

**Remarque :** les diagrammes de séquence « Ajouter un Client » et « Ajouter un Employé » sont similaires au diagramme ci-dessus.

### C. Diagramme du cas d'utilisation « Gérer un matériel » : Modifier



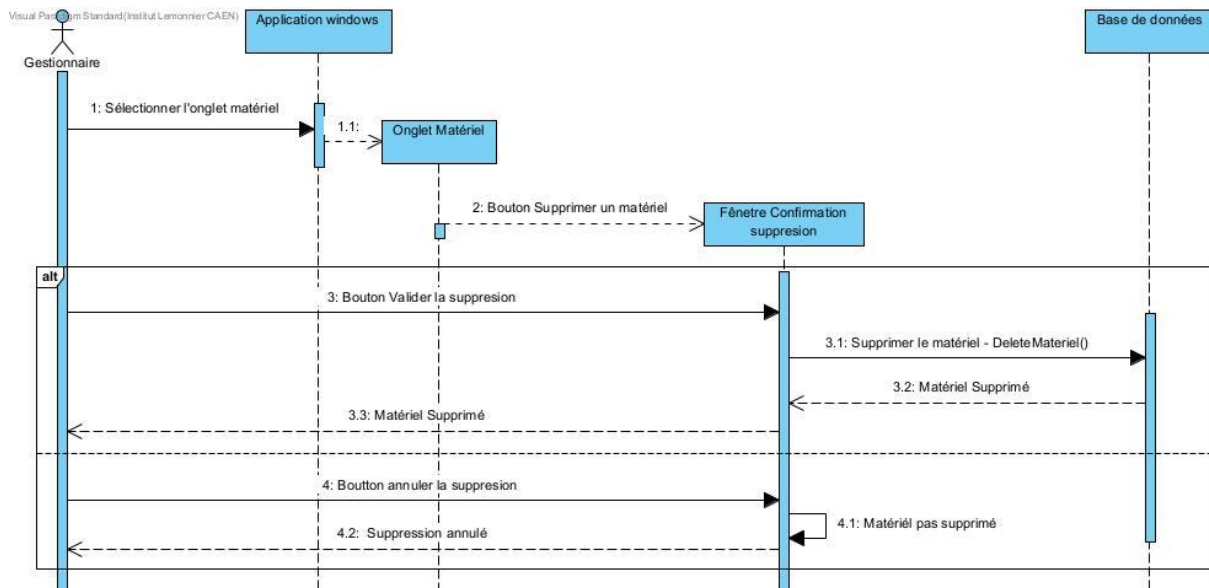
Le gestionnaire peut cliquer sur le bouton « Modifier » d'un matériel dans le datagrid la fenêtre « ModifierMatériel » s'ouvre. Les informations du matériel saisi auparavant sont récupérées et l'employé peut modifier les informations du matériel qui sont vérifiées.

- Si elles sont valides, on passe en paramètre les informations à la méthode « **ModifMatériel()** » : le matériel est modifié.
- Si elles ne sont pas valides : le matériel n'est pas modifié

Un message est affiché pour informer l'utilisateur si le matériel est modifié ou pas.

**Remarque :** les diagrammes de séquence « Modifier un Client » et « Modifier un Employé » sont similaires au diagramme ci-dessus.

## D. Diagramme du cas d'utilisation « Gérer un matériel » : Supprimer



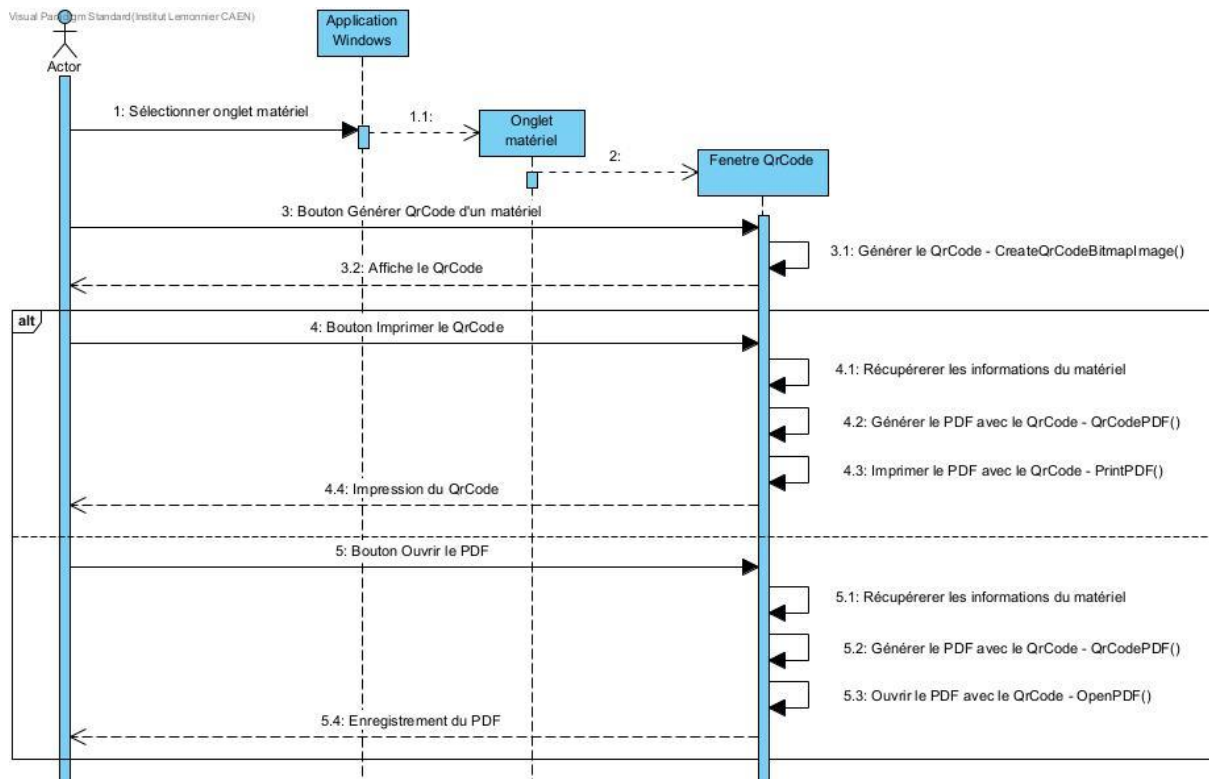
Le gestionnaire peut cliquer sur le bouton « Supprimer » d'un matériel dans le datagrid la fenêtre « SupprimerMatériel » s'ouvre. L'employé décide s'il souhaite supprimer ou non le matériel.

- Si il valide la suppression : On exécute la méthode « **DeleteMatériel()** » : le matériel est supprimé.
- Si il annule la suppression : le matériel n'est pas supprimé.

Un message est affiché pour informer l'utilisateur si le matériel est supprimé ou pas.

**Remarque :** les diagrammes de séquence « Supprimer un Client » et « Supprimer un Employé » sont similaires au diagramme ci-dessus.

## E. Diagramme du cas d'utilisation « Générer un QrCode »

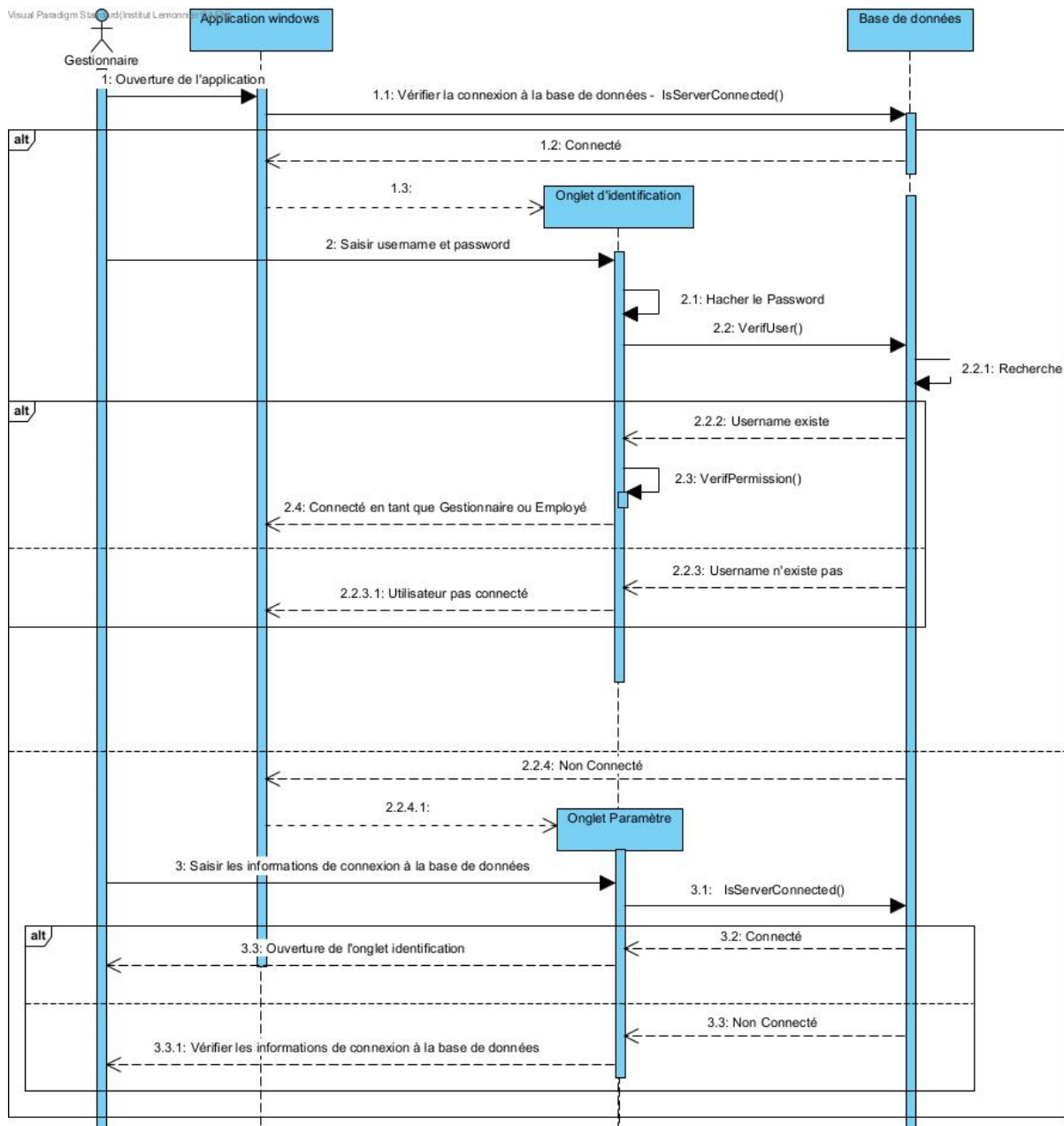


Le gestionnaire peut cliquer sur le bouton « Générer le QrCode » d'un matériel dans le datagrid, la fenêtre s'ouvre

Dans cette fenêtre, on y trouve :

- Le QrCode du matériel
- Un bouton pour imprimer le QrCode : le PDF avec le QrCode est créé et enregistré avec la méthode « **QrCodePDF()** » et on imprime le PDF avec la méthode « **PrintPDF()** »
- Un bouton pour enregistrer le QrCode en format PDF : le PDF avec le QrCode est créé et enregistré avec la méthode « **QrCodePDF()** » et on ouvre le PDF avec la méthode « **OpenPDF()** »

## F. Diagramme du cas d'utilisation « S'identifier »

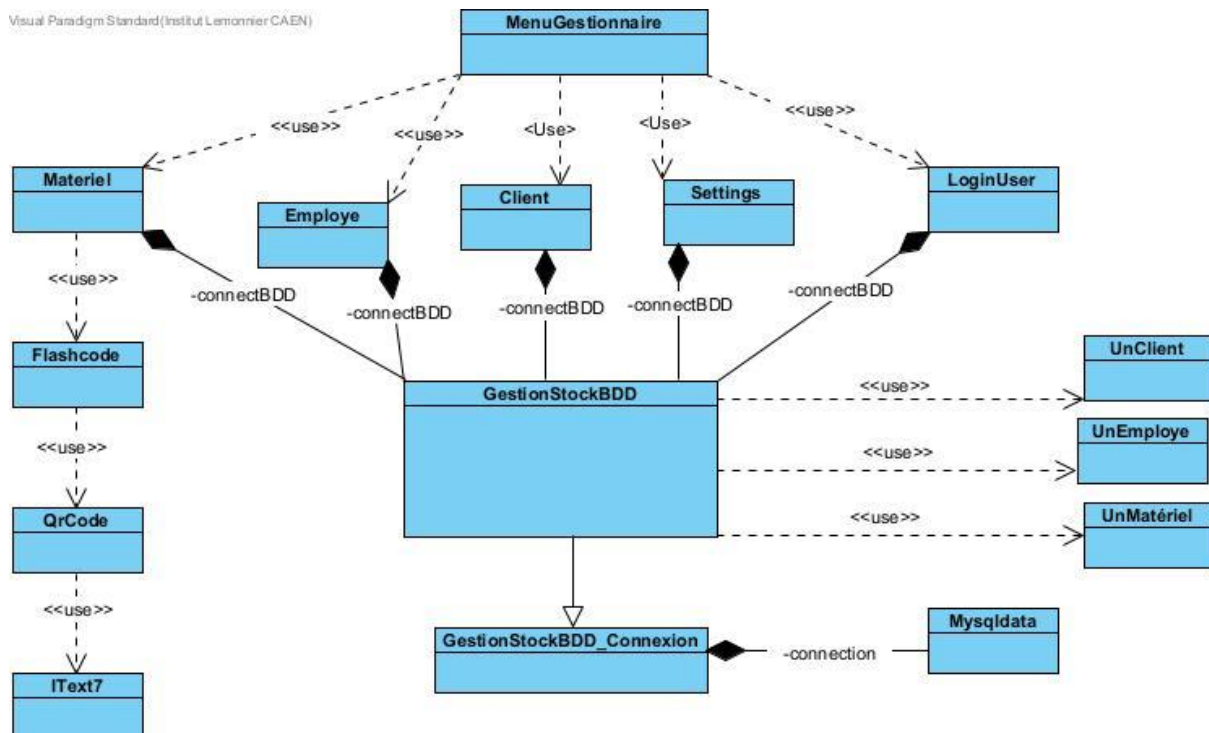


L'application vérifie la connexion à la base de données :

- Si la connexion réussit, le mot de passe est haché et vérifié dans la base de données avec la méthode « **VerifUser()** » :
  - o Si l'utilisateur existe, on récupère la permission de l'utilisateur et est connecté en tant que Gestionnaire ou Employé.
  - o Si l'utilisateur n'existe pas, on affiche que la connexion a échoué.
- Si la connexion n'a pas réussi, on se rend dans les paramètres pour changer les informations de la connexion à la base de données.

## G. Diagramme de classes

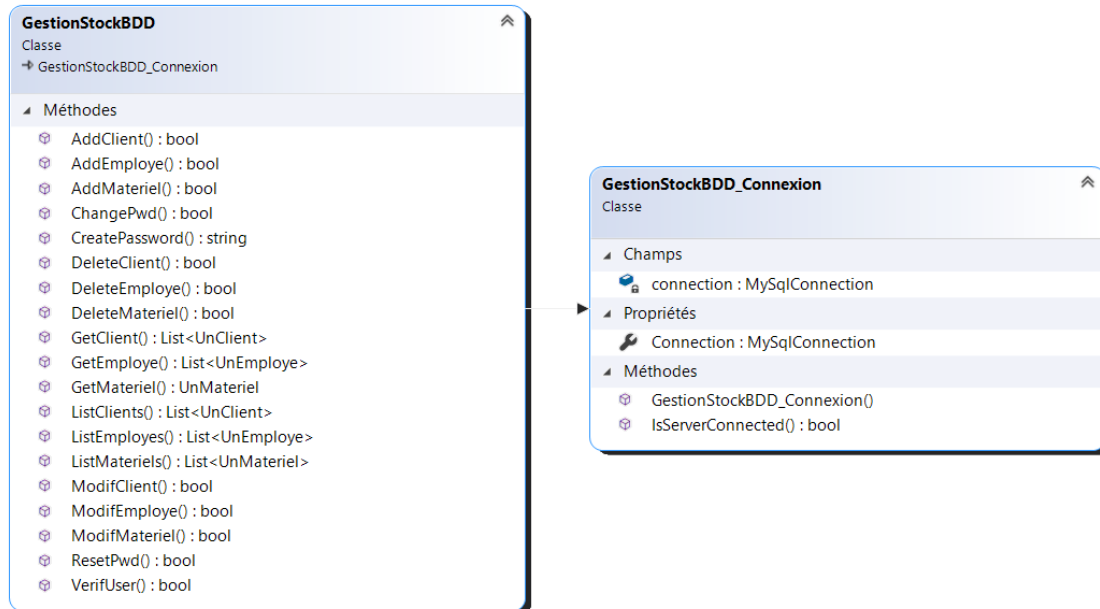
Pour la réalisation de mon application Windows, j'ai créé deux classes pour réaliser les requêtes à la base de données MySQL : Une classe « GestionStockBDD\_Connexion » qui établit la connexion à la base de données et une autre classe « GestionStockBDD » qui hérite de la classe « GestionStockBDD\_Connexion ». Cette autre classe contient toutes les méthodes qui exécutent les requêtes MySQL, elle utilise trois classes qui modélisent : « UnMateriel », « UnEmploye » et « UnClient ».



## 6. CONCEPTION

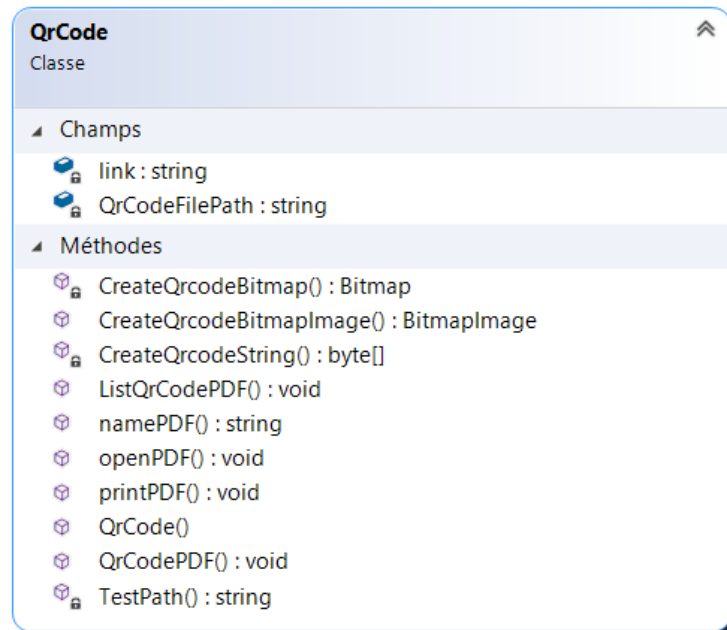
Je vais détailler les principales classes que j'ai développées.

### A. Classe GestionStockBDD\_Connexion et GestionStockBDD



Méthodes	Explication
Bool <b>AddClient()</b>	Ajouter un client.
Bool <b>AddEmploye()</b>	Ajouter un employé.
Bool <b>AddMateriel()</b>	Ajouter un matériel.
Bool <b>ChangePwd()</b>	Changer le mot de passe d'un employé pour se connecter à l'application windows.
String <b>CreatePassword()</b>	Hacher le mot de passe.
Bool <b>DeleteClient()</b>	Supprimer un client.
Bool <b>DeleteEmploye()</b>	Supprimer un employé.
Bool <b>DeleteMateriel()</b>	Supprimer un matériel.
List<UnClient> <b>GetClient()</b>	Récupérer les informations d'un client.
List<UnEmploye> <b>GetEmploye()</b>	Récupérer les informations d'un employé.
UnMateriel <b>GetMateriel()</b>	Récupérer les informations d'un matériel.
List<UnClient> <b>ListClients()</b>	Récupérer la liste des clients.
List<UnEmploye> <b>ListEmployes()</b>	Récupérer la liste des employés.
List<UnMateriel> <b>ListMateriels()</b>	Récupérer la liste des matériels.
Bool <b>ModifClient()</b>	Modifier un client.
Bool <b>ModifEmploye()</b>	Modifier un employé.
Bool <b>ModifMateriel()</b>	Modifier un matériel.
Bool <b>ResetPwd()</b>	Mettre le mot de passe par défaut pour un employé pour la connexion à l'application windows.
Bool <b>VerifUser()</b>	Vérifier si l'username et le mot de passe existe dans la base de données pour se connecter à l'application windows.
Attributs	Explication
<b>GestionStockBDD_Connexion()</b>	Crée le string de connexion à la base de données.
Bool <b>IsServerConnected()</b>	Test si la connexion à la base de données réussi.
MySqlConnection <b>Connection</b>	Contient toutes les informations pour se connecter à la base de données

## B. Classe QrCode



Méthodes	Explication
Bitmap <b>CreateQrcodeBitmap()</b>	Création du QrCode en bitmap.
BitmapImage <b>CreateQrcodeBitmapImage()</b>	Conversion du QrCode de bitmap en BitmapImage pour l'insérer dans le contrôle Image en tant que source.
Byte[] <b>CreateQrcodeString()</b>	Conversion du QrCode de bitmap en tableau d'octets.
Void <b>ListQrCodePDF()</b>	Création du PDF avec plusieurs QrCode pour l'enregistrer ou l'imprimer.
String <b>namePDF()</b>	Nom donnée au PDF quand il y a plusieurs QrCode.
Void <b>openPDF()</b>	Ouvre le PDF crée.
Void <b>printPDF()</b>	Imprime le PDF crée.
<b>QrCode()</b>	Construteur.
Void <b>QrCodePDF()</b>	Création du PDF d'un QrCode pour l'enregistrer ou l'imprimer.
String <b>TestPath()</b>	Vérifie si un dossier est ciblé pour enregistrer les PDF des QrCodes.
Attributs	Explication
String <b>link</b>	Url de ce que contient un QrCode
String <b>QrCodeFilePath</b>	Chemin d'enregistrement des PDFs



## C. Classe UnMateriel, UnEmploye et UnClient

<b>UnMateriel</b> Classe <div> <div>Champs</div> <ul style="list-style-type: none"> <li>caracteristique : string</li> <li>categorie : string</li> <li>flashcode : string</li> <li>id_materiel : int</li> <li>prix_ht : double</li> <li>reference : string</li> </ul> <div>Propriétés</div> <ul style="list-style-type: none"> <li>Caracteristique : string</li> <li>Categorie : string</li> <li>Flashcode : string</li> <li>Id_materiel : int</li> <li>Prix_ht : double</li> <li>Reference : string</li> </ul> <div>Méthodes</div> <ul style="list-style-type: none"> <li>UnMateriel() (+ 1 surcharge)</li> </ul> </div>	<b>UnEmploye</b> Classe <div> <div>Champs</div> <ul style="list-style-type: none"> <li>email : string</li> <li>id_employe : int</li> <li>nom : string</li> <li>password : string</li> <li>permission : bool</li> <li>prenom : string</li> <li>telephone : string</li> <li>user : string</li> </ul> <div>Propriétés</div> <ul style="list-style-type: none"> <li>Email : string</li> <li>Id_employe : int</li> <li>Nom : string</li> <li>Password : string</li> <li>Permission : bool</li> <li>Prenom : string</li> <li>Telephone : string</li> <li>User : string</li> </ul> <div>Méthodes</div> <ul style="list-style-type: none"> <li>UnEmploye()</li> </ul> </div>	<b>UnClient</b> Classe <div> <div>Champs</div> <ul style="list-style-type: none"> <li>adresse_client : string</li> <li>code_postale_client : string</li> <li>email_client : string</li> <li>id_client : int</li> <li>nom_client : string</li> <li>prenom_client : string</li> <li>telephone_client : string</li> <li>ville_client : string</li> </ul> <div>Propriétés</div> <ul style="list-style-type: none"> <li>Adresse_client : string</li> <li>Code_postale_client : string</li> <li>Email_client : string</li> <li>Id_client : int</li> <li>Nom_client : string</li> <li>Prenom_client : string</li> <li>Telephone_client : string</li> <li>Ville_client : string</li> </ul> <div>Méthodes</div> <ul style="list-style-type: none"> <li>UnClient()</li> </ul> </div>
---	---	---

« Un Materiel », « unEmploye » et « unClient » modélisent les tables dans la base de données « Matériels », « Employés » et « Clients ».

## D. Classe Materiel

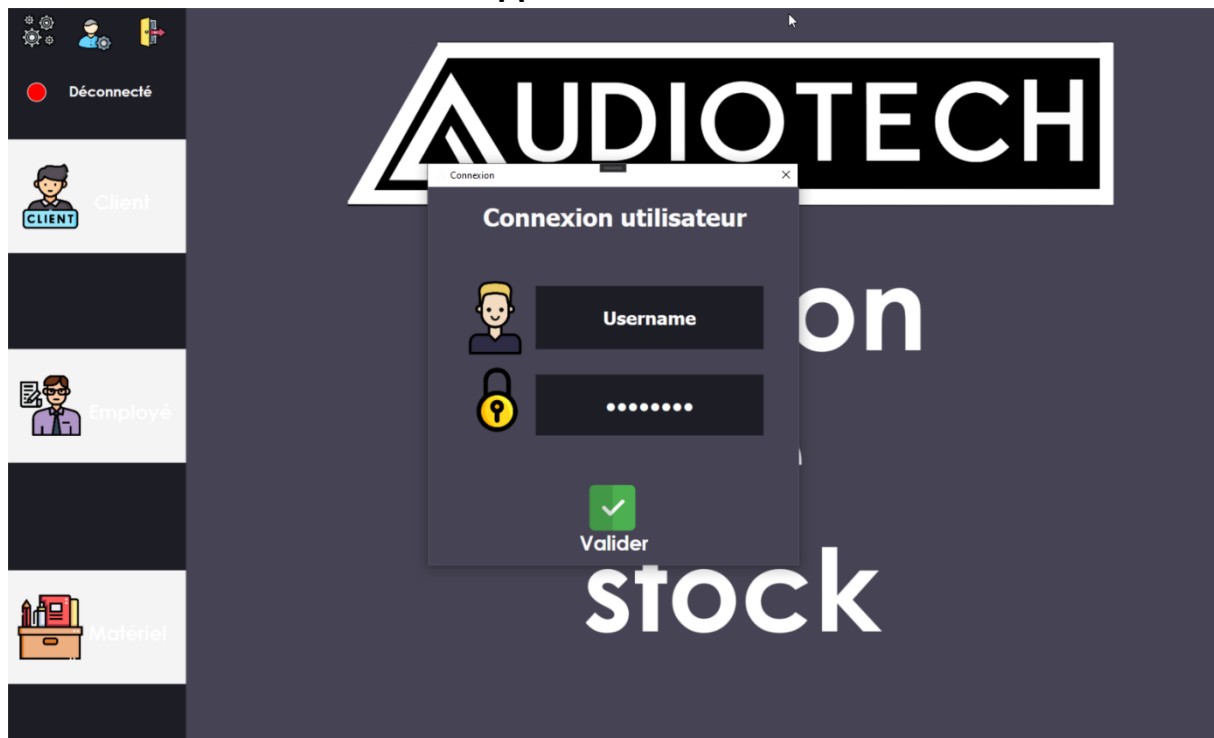
<b>Materiel</b> Classe → UserControl <div> <div>Champs</div> <ul style="list-style-type: none"> <li>connectBDD : GestionStockBDD</li> </ul> <div>Méthodes</div> <ul style="list-style-type: none"> <li>bntDelete_Click(object sender, RoutedEventArgs e) : void</li> <li>btnAjouter_Click(object sender, RoutedEventArgs e) : void</li> <li>btnQrCode_Click(object sender, RoutedEventArgs e) : void</li> <li>btnReload_Click(object sender, RoutedEventArgs e) : void</li> <li>CheckBox_Checked(object sender, RoutedEventArgs e) : void</li> <li>CheckBox_Unchecked(object sender, RoutedEventArgs e) : void</li> <li>DeleteButton_Click(object sender, RoutedEventArgs e) : void</li> <li>EditButton_Click(object sender, RoutedEventArgs e) : void</li> <li>FlashCodeButton_Click(object sender, RoutedEventArgs e) : void</li> <li>IsConnected() : void</li> <li>Materiel()</li> <li>ReloadGrid() : void</li> <li>searchButton_Click(object sender, RoutedEventArgs e) : void</li> <li>searchTextBox_KeyDown(object sender, KeyEventArgs e) : void</li> <li>UserControl_Loaded(object sender, RoutedEventArgs e) : void</li> <li>UserControl_Unloaded(object sender, RoutedEventArgs e) : void</li> </ul> </div>
---

Cette classe modélise la vue pour la gestion des matériels.

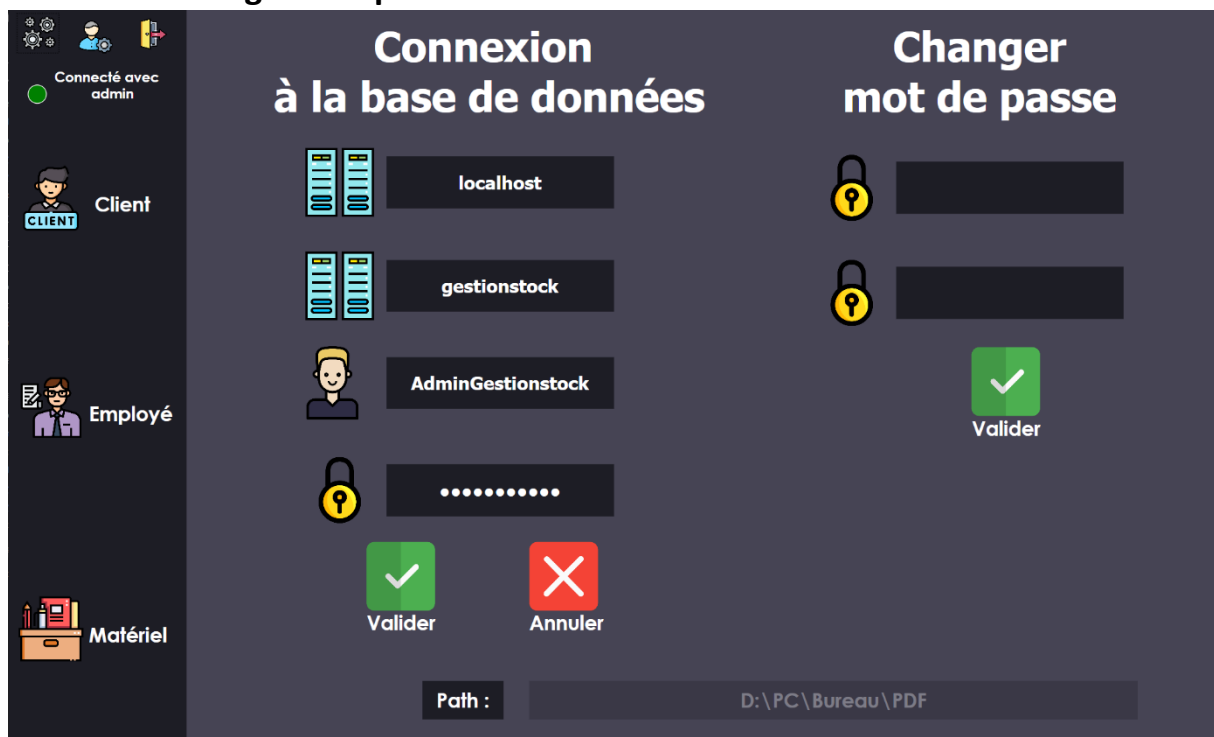
Cette classe a des similitudes avec celle de Employe, Client et Settings.

## 7. LES DIFFERENTES VUES

### A. A l'ouverture de l'application



### B. Onglet des paramètres



## C. Liste des matériels

Connecté avec admin

Client

Employé

Matériel

Ajouter Supprimer QrCode

Reference	Catégorie	Prix HT	FlashCode
4CANAUX	AMP.	€ 60	
ACL	TRAD.	€ 35	
ADAPTXO	STR.	€ 2	
ADAT8	PERI.	€ 21.93	
ALIM3X16A	ALIM.	€ 15	
ALIM50M	ALIM.	€ 15	
ANGLE250	STR.	€ 5	
APEXDBQZERO	PERI.	€ 25	
APEXGX230	PERI.	€ 25	
ARCOM4000		€ 200	
ARCS	ENC.	€ 33	
ARMOIRE125A	ALIM.	€ 70	

## D. Ajouter un matériel

Connecté avec admin

Client

Employé

Matériel

Ajouter Supprimer QrCode

Nouveau Materiel

Reference Catégorie Caractéristique

Prix HT Quantite

Valider

ARCS	ENC.	€ 33	
ARMOIRE125A	ALIM.	€ 70	

## E. Modifier un matériel

Connecté avec admin

Client

Employé

Matériel

Ajouter Supprimer QrCode

**Modifier Matériel**

Reference: 4CANAUX

Categorie: AMP.

Caracteristique: Fp10k - 4x1300W/8 - 4x2000W/4 - 4x2500/2

Prix HT: 60

Valider

Reference	Categorie	Prix HT
ARCS	ENC.	€ 33
ARMOIRE125A	ALIM.	€ 70

## F. Supprimer un matériel

Connecté avec admin

Client

Employé

Matériel

Ajouter Supprimer QrCode

Echec

**Voulez-vous supprimer définitivement ce matériel ?**

Oui Non

Reference	Categorie	Prix HT	FlashCode
ALIM3X16A	ALIM.	€ 15	
ALIM50M	ALIM.	€ 15	
ANGLE250	STR.	€ 5	
APEXDBQZERO	PERI.	€ 25	
APEXGX230	PERI.	€ 25	
ARCOM4000		€ 200	
ARCS	ENC.	€ 33	
ARMOIRE125A	ALIM.	€ 70	

## G. Génération du QrCode

The screenshot shows the FlashCode application interface. The sidebar on the left contains navigation options: 'Connecté avec admin' (top), 'Client', 'Employé', and 'Matériel' (bottom). The main area displays a table of equipment with columns for 'Refe', 'FlashCode', and prices. A modal window titled 'FlashCode' is open, showing a QR code and buttons for 'PDF' and 'Imprimer'.

Refe	FlashCode	Price
AD/		€ 25
ALIM		€ 200
ALIM		€ 33
ANG		€ 70
APEXC		
APEXGX230	PERI.	€ 25
ARCOM4000		€ 200
ARCS	ENC.	€ 33
ARMOIRE125A	ALIM.	€ 70

## H. Liste des clients

The screenshot shows a web application interface for a company database. The interface is in French and features a dark sidebar on the left with navigation icons for Admin, Client, Employé, and Matériel. The main area has a top bar with 'Ajouter' and 'Supprimer' buttons, a search bar, and a table of employees. The table has columns for Nom, Prénom, Email, Téléphone, and Adresse. Each row includes edit and delete icons.

	Nom	Prénom	Email	Téléphone	Adresse
	bernard	michel	michel@gmx.fr	0612345678	rue hérouvill 14000 caen
	Alexandre	Debriou	alex.debrou@gmail.com	0669696969	oooh 140000 caen
	Planchon	Doryan	doryan.planchon@hotmail.fr	0234567898	ecu 14000 caen
	Helleu	Paul	Paul.helleu@gmail.com	0134567899	stmalo 35400 stmalo
	Dupont	Robert	Dupont.Robert@gmail.com	0663442623	6 rue caillem 14320 caen

## I. Ajouter un client

Connecté avec admin

Ajouter Supprimer

AjouterMatériel

### Nouveau Client

Nom	Prénom	Email
<input type="text"/>	<input type="text"/>	<input type="text"/>
Téléphone	Adresse	Code Postale
<input type="text"/>	<input type="text"/>	<input type="text"/>
Ville		
<input type="text"/>		

☒ Valider

## J. Liste des employés

Connecté avec admin

Ajouter Supprimer

Client

Employé

Matériel

	Nom	Prénom	Email	Téléphone	Reset Password
<input type="checkbox"/>	admin	admin	admin@gmail.com	0612345678	<input type="button" value="Reset"/>
<input type="checkbox"/>	Planchon	Doryan	doryan.planchon@hotmail.fr	0123456789	<input type="button" value="Reset"/>

## K. Ajouter un employé

The screenshot shows the 'Ajouter un employé' (Add employee) form. The form is titled 'Nouveau Employé' and contains the following fields:

- Nom
- Prénom
- Email
- Téléphone
- Permission
- User

At the bottom of the form is a green checkmark icon and the label 'Valider'. The form is displayed over a dark background with a sidebar on the left containing icons for 'Connecté avec admin', 'Client', 'Employé', and 'Matériel'. At the top of the main area, there are buttons for 'Ajouter' (Add) and 'Supprimer' (Delete).

## L. Reset password employé

The screenshot shows the 'Reset password employé' (Reset employee password) form. The form is titled 'Voulez-vous reset le mot de passe de cet employé ?' (Do you want to reset the password of this employee?). It contains the following fields:

- Nom
- Prénom
- Email
- Téléphone
- Reset Password

At the bottom of the form are two buttons: 'Oui' (Yes) and 'Non' (No). The form is displayed over a dark background with a sidebar on the left containing icons for 'Connecté avec admin', 'Client', 'Employé', and 'Matériel'. At the top of the main area, there are buttons for 'Ajouter' (Add) and 'Supprimer' (Delete).

## 8. POINT ESSENTIEL DE MON ETUDE: «S'identifier»

J'ai choisi de détailler ce cas d'utilisation car c'est celui qui permet de donner l'accès à l'utilisation de ma partie selon les droits d'accès dont dispose un employé. Sans ce système d'authentification mon application n'est pas utilisable car les cas d'utilisations « Gérer un matériel, un employé et un client » sont accessibles seulement si on sait que l'employé a les droits de gestionnaire.

### A. Lancement de l'application

Mon application Windows se lance grâce à l'application « Employé » de Monsieur DEQUAINDRY, lorsqu'un utilisateur se connecte en tant que « Gestionnaire », il est possible de lancer mon application en cliquant sur un bouton.

Lorsque mon application est lancée, je récupère les informations de connexion à la base de données et d'utilisateur. J'utilise les settings pour sauvegarder ces informations cela permet la persistance des données.

On se retrouve donc avec cette vue :





## B. Les différentes méthodes pour la connexion

### a. Connexion par récupération d'un utilisateur

Mais ce que l'on voit pas quand on lance l'application c'est l'exécution d'une méthode : « **start()** ».

```
public void start()
{
    Accueil accueil = new Accueil();
    PanelAfficher.Children.Clear();
    if (!PanelAfficher.Children.Contains(accueil))
    {
        PanelAfficher.Children.Add(accueil);
        accueil.BringIntoView();
    }
    else
    {
        accueil.BringIntoView();
    }

    string[] args = Environment.GetCommandLineArgs();
    if (args.Length > 1)
    {
        Properties.Settings.Default.IP = args[1];
        Properties.Settings.Default.Database = args[2];
        Properties.Settings.Default.BDDUsername = args[3];
        Properties.Settings.Default.BDDPassword = args[4];
        Properties.Settings.Default.Permission = args[5];
        Properties.Settings.Default.User = args[6];

        Properties.Settings.Default.UrlQrCode = "http://projet.2brou.fr/materiel/";
        Properties.Settings.Default.Save();

        LoginUser.LoginUser loginUser = new LoginUser.LoginUser();

        if (BDD.IsServerConnected() == true)
        {
            if (Properties.Settings.Default.User == "DEFAULT_VALUE")
            {
                txtConnected.Text = "\n Déconnecté";
                Led.Fill = Brushes.Red;

                btnClient.IsEnabled = false;
                btnEmploye.IsEnabled = false;
                btnMateriel.IsEnabled = false;

                loginUser.ShowDialog();
            }
            else
            {
                loginUser.VerifPermission();
            }
        }
        else
        {
            btnSetting_Click(this, new RoutedEventArgs());
        }
    }
}
```

Affichage de la page d'accueil.

Récupération des informations de connexion à la base de données et d'utilisateur. Passer par l'application de Mr DEQUAINDRY.

Définit l'url pour le QrCode.

Ouvre la fenêtre d'identification

Vérifie si la connexion à la base de données est possible

Vérifie le nom d'utilisateur

Définir le statut de connexion en « Déconnecté ».

Désactive les boutons des onglets : Client, Employé et Matériel.

Vérifie les permissions de l'utilisateur.

Ouvre l'onglet des paramètres

On récupère toutes les informations pour se connecter à la base de données et d'utilisateur ainsi que l'initialisation du lien du QRCode.

On vérifie d'abord si la connexion à la base de données est possible

- Si c'est le cas on test l'identification.
- Si ce n'est pas le cas on se retrouve dans l'onglet des paramètres.

On vérifie ensuite s'il y a bien un nom d'utilisateur récupérer.

- Si c'est le cas on exécute, la méthode « **VerifPermission()** »  
Si ce n'est pas le cas on indique un état « Déconnecté », on désactive les boutons et ouvre la page d'identification.

### La méthode « IsServerConnected() »

```
private MySqlConnection connection;
99+ références
public MySqlConnection Connection { get => connection; set => connection = value; }

0 références
public GestionStockBDD_Connexion()
{
    MySqlConnectionStringBuilder builder = new MySqlConnectionStringBuilder();
    builder.Server = Properties.Settings.Default.IP;
    builder.Database = Properties.Settings.Default.Database;
    builder.UserID = Properties.Settings.Default.BDDUsername;
    builder.Password = Properties.Settings.Default.BDDPassword;
    string connectionString = builder.ToString();
    Connection = new MySqlConnection(connectionString);
}
```

Crée le string pour établir la connexion à la base de données.

```
public bool IsServerConnected()
{
    using (Connection)
    {
        try
        {
            Connection.Open();
            return true;
        }
        catch
        {
            return false;
        }
    }
}
```

Test la connexion à la base de données.

## La méthode « VerifPermission() »

```

public void VerifPermission()
{
    Interface.MenuGestionnaire menu = Application.Current.Windows.OfType<Interface.MenuGestionnaire>().FirstOrDefault();

    if (Properties.Settings.Default.Permission == "True")
    {
        menu.btnClient.IsEnabled = true;
        menu.btnEmploye.IsEnabled = true;
        menu.btnMateriel.IsEnabled = true;

        menu.txtConnected.Text = "Connecté avec : \n" + Properties.Settings.Default.User;
        menu.Led.Fill = Brushes.Green;
    }
    else
    {
        menu.btnClient.IsEnabled = false;
        menu.btnEmploye.IsEnabled = false;
        menu.btnMateriel.IsEnabled = false;

        menu.txtConnected.Text = "Connecté avec : \n" + Properties.Settings.Default.User;
        menu.Led.Fill = Brushes.Green;
    }
}

```

Active les boutons des onglets

Désactive les boutons des onglets

Afficher dans un Textbox le nom de l'utilisateur qui est connecté

## b. Déconnexion et connexion via la fenêtre d'identification



## Bouton déconnexion

```

private void btnDisconnect_Click(object sender, RoutedEventArgs e)
{
    if (Properties.Settings.Default.User != "DEFAULT_VALUE")
    {
        txtConnected.Text = "\n Déconnecté";
        Led.Fill = Brushes.Red;

        btnClient.IsEnabled = false;
        btnEmploye.IsEnabled = false;
        btnMateriel.IsEnabled = false;

        Properties.Settings.Default.Permission = "DEFAULT_VALUE";
        Properties.Settings.Default.User = "DEFAULT_VALUE";
        Properties.Settings.Default.Save();

        if (BDD.IsServerConnected() == true)
        {
            LoginUser loginUser = new LoginUser.LoginUser();
            loginUser.ShowDialog();
        }
        else
        {
            btnSetting_Click(this, new RoutedEventArgs());
        }
    }
}

```

Vérifie si un employé est connecté.

Définir le statut de connexion en « Déconnecté ».


Désactive les boutons des onglets : Client, Employé et Matériel.


Vérifie si la connexion à la base de données est possible


Ouvre la fenêtre d'identification

Ouvre l'onglet des paramètres

### Connexion utilisateur





  
**Valider**

#### Bouton de validation de connexion

```
private void btnConnexion_Click(object sender, RoutedEventArgs e)
{
    BDD.GestionStockBDD BDD = new BDD.GestionStockBDD();

    if (BDD.IsServerConnected() == true)
    {
        if (Verification() == true)
        {
            if (BDD.VerifUser(Settings_txtUser.Text, Settings_txtPassword.Password))
            {
                SuccessDialog success = new SuccessDialog();
                success.ShowDialog();
                VerifPermission();
                this.Close();
            }
            else
            {
                FailedDialog failed = new FailedDialog();
                failed.ShowDialog();
            }
        }
        else
        {
            FailedDialog failed = new FailedDialog();
            failed.lblNom.Text = "Définir les paramètres de connexion \n à la base de données";
            failed.ShowDialog();
        }
    }
}
```

Vérifie si la connexion à la base de données est possible

Vérifie le contenu des champs

Vérifie le nom d'utilisateur et mot de passe dans la base de données

Vérifie les permissions de l'utilisateur

### La méthode « Verification() »

```
private bool Verification()
{
    if(Settings_txtUser.Text == "" || Settings_txtUser.Text == "Username")
    {
        FailedDialog failed = new FailedDialog();
        failed.lblNom.Text = "Insérer un username";
        failed.ShowDialog();
        return false;
    }

    if(Settings_txtPassword.Password == "" || Settings_txtPassword.Password == "Password")
    {
        FailedDialog failed = new FailedDialog();
        failed.lblNom.Text = "Insérer un mot de passe";
        failed.ShowDialog();
        return false;
    }

    return true;
}
```

Vérifie si les champs « username » et « mot de passe » ne soit pas vide ou avec des valeurs par défaut

### VerifUser()

```
public bool VerifUser(string username, string password)
{
    try
    {
        string Hashpassword = CreatePassword(password);

        using (Connection)
        {
            Connection.Open();
            MySqlConnection transaction = Connection.BeginTransaction();

            string sql = "select permission, identifiant from employees where password = @Password AND identifiant = @Username";

            using (MySQLCommand command = new MySQLCommand(sql, Connection))
            {
                command.Parameters.AddWithValue("@Username", username);
                command.Parameters.AddWithValue("@Password", Hashpassword);

                using (MySQLDataReader reader = command.ExecuteReader())
                {
                    if (reader.HasRows)
                    {
                        reader.Read();
                        Properties.Settings.Default.Permission = Convert.ToString(reader["permission"]);
                        Properties.Settings.Default.User = Convert.ToString(reader["identifiant"]);
                        return true;
                    }
                    return false;
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        return false;
    }
}
```

### C. Les différentes vues de connexion

Voilà la fenêtre lorsqu'un employé est connecté avec les permissions de gestionnaire.



Voilà la fenêtre lorsqu'un employé n'est pas connecté avec les permissions de gestionnaire.



## II. Conclusion

Ce projet m'a beaucoup apporté, j'ai approfondi mes connaissances acquises pendant les deux années de BTS ce qui m'a permis de concevoir une application Windows en WPF C#.

Grâce à ce projet, cela m'a permis d'apprendre à respecter un cahier des charges, de travailler en équipe.

Je suis fier de l'application que j'ai pu concevoir, elle est pleinement fonctionnelle et esthétique. Elle répond aux attentes demandées par l'entreprise qui est de gérer les matériels, client et matériel selon l'utilisateur connecté.