

INTRODUCTION

Le présent travail est un travail individuel donnée dans le cadre d'une mise en pratique des connaissances reçu durant notre cours de langage de programmation « C# ».

De ce fait, il nous a été demandé de concevoir une application desktop au choix et qui concernerait une certaine gestion, et pour notre cas, nous avons jugé bon de mettre en place une petite application d'enregistrement ou d'inscription des étudiants, cette application n'est en réalité petit qu'un module d'une gestion bien plus costaud.

A travers cette modeste application nous espérons balayer, et aussi refléter la consistance de connaissances reçues pendant notre cours de langage de programmation.

PRESENTATION DE L'APPLICATION

Tout application de gestion digne de ce nom, à son lancement commence toujours par une interface de connexion ou si l'on veut d'authentification de l'utilisateur, dite interface que l'on peut précéder d'un formulaire de chargement, que nous n'avons pas utilisé dans notre cas, ainsi notre modeste application ou module d'application se résume en deux interfaces principales :

- ❖ **Interface** de Connexion à l'application
- ❖ **Interface** de Mise à jour des étudiants



Interface de Connexion à l'application

Dans le cadre de notre application, c'est la première interface sur laquelle tombera l'utilisateur au lancement, celle-ci permet simplement à l'utilisateur de se connecter éventuellement à l'application, en demandant à ce dernier de fournir un **nom d'utilisateur** et un **mot de passe** valide, si tant est que ces deux informations s'avère valide, il pourra accéder à l'application. Comme l'indique la capture ci-dessous :

The screenshot shows a login window with the following elements:

- Window title: **Connection à l'application**
- User icon: A placeholder image of a person with an orange head and blue body.
- Username label: **Saisissez votre nom d'utilisateur**
- Username input field: Contains the text **Marie**.
- Password label: **Saisissez votre mot passe**
- Password input field: Contains four dots (**....**) to mask the password.
- Login button: **Connectez-vous**

Callouts from the image:

- Nom d'utilisateur valide** (points to the username field)
- Mot de passe valide, mais masqué !!** (points to the password field)



Interface de Mise à jour des étudiants

Dans le cadre de notre application, c'est la deuxième et la seule interface de gestion proprement dite, celle-ci permet simplement de faire la mise à jour de étudiant en trois étapes majeurs (Ajout, modification et suppression), elle permet évidemment d'effectuer des recherches, et fournis aussi le compte de l'ensemble des étudiants inscrits. Comme l'indique la capture ci-dessous :

Mise à jour des étudiants

Ajouter

Modifier

Supprimer

Actualiser

Informations Etudiant

Matricule E006

Nom : SUAREZ

Post-nom : PACO

Prenom : LUIS

Adresse : Uruguay

Promotion G3 INFO

Sexe : M

Identification Incorrect

Insertion Reussi !!

OK

Matricule	Nom	Post-nom	Prenom	Sexe	Adresse
E005	SUAREZ	PACO	LUIS	M	Uruguay
E001	MESSI	Padré	Lionel	M	Buenos air
E002	SUAREZ	PACO	LUIS	M	Uruguay
E003	SUAREZ	PACO	LUIS	M	Uruguay
E004	NYEMBO	HORNY	Hornella	F	Espagne, t

Nombre d'enregistrement :
5 Etudiant(s)

dimanche 23 juin 2019



Code source Connexion à la Base de données

```

class Connexion
{
    //----- Déclaration des variables pour la connexion à la base de donnée
    //----- Avec le pilote OleDb
    public OleDbConnection Mycx ;
    public OleDbCommand Mycmd;
    public OleDbDataReader Lire;

    //---- Initialisation de la variable de connexion et Connexion à la base
    de données
    public void getConnection()
    {
        //Mycx = new OleDbConnection(@"provider=Microsoft.ace.oledb.12.0;data
        source=Gestion_Etudiant.accdb");
        Mycx = new OleDbConnection("Provider = Microsoft.Jet.Oledb.4.0;Data
        source=Gestion_Etudiant.mdb");

        Mycx.Open();
    }

    public void Mise_jour_(string requete)
    {
        try
        {
            getConnection();
            Mycmd = new OleDbCommand(requete, Mycx);
            Mycmd.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
            MessageBox.Show("ERREUR : \n" + ex.Message, "Erreur detectée",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    public void Affichage(string requete)
    {
        try
        {
            getConnection();
            Mycmd = new OleDbCommand(requete, Mycx);
            Lire = Mycmd.ExecuteReader();
        }
        catch (Exception ex)
        {
            MessageBox.Show("ERREUR : \n" + ex.Message, "Erreur detectée",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

{...} Code source Connexion à l'application

1.

Bouton Se connecter

```
private void BtnValider_Click(object sender, EventArgs e)
{
    if (txtnom.Text == "")
    {
        MessageBox.Show("Veuillez fournir le nom d'utilisateur", "Zone vide détectée", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else if (txtpass.Text == "")
    {
        MessageBox.Show("Veuillez fournir le mot de passe", "Zone vide détectée", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        String req = "select * from utilisateur where nom='" + txtnom.Text + "' AND pass='" + txtpass.Text + "'";
        cnx.Affichage(req);
        if (cnx.Lire.Read())
        {
            Etudiant frmEt = new Etudiant();
            this.Hide();
            frmEt.Show();
        }
        else
        {
            MessageBox.Show("Nom d'utilisateur ou mot de passe incorrect !!", "Identification Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtnom.Text = "";
            txtpass.Text = "";
        }
    }
}
```

{...} Code source Mise à jour etudiants

1.

Bouton Ajouter

```
private void btnClient_Click(object sender, EventArgs e)
{
    try
    {
        if (txtmat.Text == "" || txtnom.Text == "" || txtpost.Text
        == "" || txtpre.Text == "" || txtadres.Text == "" || CboSexe.Text == "" ||
        lblpromo.Text == "")
        {
            MessageBox.Show("Zone vide detectée \n Veuillez à
            remmplir toutes les infos.", "Vide", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        }
        else
        {
            String req = "Insert into Etudiant
            (nom,postn,prenom,sexe,adresse,matricule,promot) values ('" +
            txtnom.Text.ToString() + "','" + txtpost.Text.ToString() + "','" +
            txtpre.Text.ToString() + "','" + CboSexe.Text.ToString() + "','" +
            txtadres.Text.ToString() + "','" + txtmat.Text.ToString() + "','" +
            lblpromo.Text.ToString() + "')";
            cnx.Mise_jour_(req);
            MessageBox.Show("Insertion Reussi !! ", "Identification
            Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information);
            Nettoyer();
            Remplir_Grille();
            compter();
        }
    }
    catch (Exception){}
}
```

2.

Bouton Modifier

```
private void btn_modif_Click(object sender, EventArgs e)
{
    try
    {
        if (txtmat.Text == "")
        {
            MessageBox.Show("Veuillez fournir au moin le Matricule
            etudiant !! ", "Identification Incorrect", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        }
    }
}
```

```

    }
    else
    {
        String req = "Update Etudiant Set nom='" + txtnom.Text
+ "',postn='" + txtpost.Text + "',prenom='" + txtpre.Text + "',adresse='" +
txtadres.Text + "',sexe='" + CboSexe.Text.ToString() + "',promot='" +
lblpromo.Text.ToString() + "' where matricule='" + txtmat.Text + "'";
        cnx.Mise_jour_(req);
        MessageBox.Show("Modification reussi !! ",
"Identification Incorrect", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        Nettoyer();
        Remplir_Grille();
        compter();
    }
}
catch (Exception ex)
{
}
}
}

```

Bouton Supprimer

3.

```

private void btnSupp_Click(object sender, EventArgs e)
{
    try {
        if (txtmat.Text=="") {
            MessageBox.Show("Veuillez fournir au moin le Matricule
etudiant !! ", "Identification Incorrect", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
        else {
            String req = "Delete from Etudiant where
matricule='"+txtmat.Text+"'";
            cnx.Mise_jour_(req);
            MessageBox.Show("Suppression reussi !! ", "Identification
Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information);
            Nettoyer();
            Remplir_Grille();
            compter();
        }
    }
    catch (Exception ex) {
    }
}
}

```

CONCLUSION

Nous voici donc arrivés à la fin de notre travail, nous tenons à préciser que le contenu de ce travail est une **gestion d'inscriptions basique**, à travers de laquelle nous avons essayé d'appliquer et refléter dans la mesure du possible la consistance des connaissances reçu durant notre cours de langage de programmation , l'expérience a été belle et nous continuerons de l'approfondir au-delà de ce travail.