

PROGRAMME PRINCIPAL PROJET C#

SIO2 Sofiane

Programme principal:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using MySql.Data.MySqlClient;

namespace Projet_C__1

{

    public class BD

    {

        private static string connectionString = "Server=127.0.0.1 ; Database=laboratoire; Uid=root

; Password=";

        public static void AjouterTechnicien(Technicien technicien)

        {

            MySqlConnection con = new MySqlConnection(connectionString);

            con.Open();

            MySqlCommand mySqlCommand = con.CreateCommand();

            mySqlCommand.CommandText="INSERT INTO TECHNICIEN VALUES (@matricule,

@nom, @prenom, @competences";

            mySqlCommand.Parameters.AddWithValue("@matricule", technicien.getMatricule());
```

```

mySqlCommand.Parameters.AddWithValue("@nom", technicien.getNom());

mySqlCommand.Parameters.AddWithValue("@prenom", technicien.getPrenom());

mySqlCommand.Parameters.AddWithValue("@competences",
technicien.getCompetences());

mySqlCommand.ExecuteNonQuery();

con.Close();

}

public static List<Technicien> getLesTechniciens()

{

Technicien unTechnicien;

List<Technicien> lesTechniciens = new List<Technicien>();

MySqlConnection con = new MySqlConnection(connectionString);

con.Open();

MySqlCommand mySqlCommand = con.CreateCommand();

mySqlCommand.CommandText = "SELECT * FROM Technicien";

MySqlDataReader reader = mySqlCommand.ExecuteReader();

reader=mySqlCommand.ExecuteReader();

while (reader.Read())

{

unTechnicien = new Technicien(Convert.ToInt32(reader["matricule"]),

reader["nom"].ToString(), reader["prenom"].ToString(), reader["competences"].ToString(),

reader["motDePasse"].ToString());

lesTechniciens.Add(unTechnicien);

}

con.Close();

return lesTechniciens;

```

```

}

public static List<Utilisateur> getLesUtilisateurs()
{
    Utilisateur unUtilisateur;

    List<Utilisateur> lesUtilisateurs = new List<Utilisateur>();

    MySqlConnection con = new MySqlConnection(connectionString);

    con.Open();

    MySqlCommand mySqlCommand = con.CreateCommand();

    mySqlCommand.CommandText = "SELECT * FROM utilisateur";

    MySqlDataReader reader = mySqlCommand.ExecuteReader();

    reader = mySqlCommand.ExecuteReader();

    while (reader.Read())
    {
        unUtilisateur = new Utilisateur(Convert.ToInt32(reader["id"]), reader["nom"].ToString(),
        reader["prenom"].ToString(), reader["region"].ToString(), reader["email"].ToString(),
        Convert.ToInt32(reader["responsable"]), reader["motDePasse"].ToString());

        lesUtilisateurs.Add(unUtilisateur);
    }

    con.Close();

    return lesUtilisateurs;
}

public static List<Materiel> getLesMateriaux()
{
    Materiel unMateriel;

    List<Materiel> lesMateriaux = new List<Materiel>();

    MySqlConnection con = new MySqlConnection(connectionString);

```

```

con.Open();

MySQLCommand mySqlCommand = con.CreateCommand();

mySqlCommand.CommandText = "SELECT * FROM Materiel";

MySQLDataReader reader = mySqlCommand.ExecuteReader();

reader = mySqlCommand.ExecuteReader();

while (reader.Read())

{

unMateriel = new Materiel(Convert.ToInt32(reader["id"]), reader["type"].ToString(),

reader["modele"].ToString(), Convert.ToDateTime(reader["date_acquisition"]),

reader["etat"].ToString());

lesMateriaux.Add(unMateriel);

}

con.Close();

return lesMateriaux;

}

public static List<Intervention> getLesInterventions()

{

Intervention uneIntervention;

List<Intervention> lesInterventions = new List<Intervention>();

MySQLConnection con = new MySQLConnection(connectionString);

con.Open();

MySQLCommand mySqlCommand = con.CreateCommand();

mySqlCommand.CommandText = "SELECT * FROM intervention";

MySQLDataReader reader = mySqlCommand.ExecuteReader();

reader = mySqlCommand.ExecuteReader();

while (reader.Read())

```

```

{
    uneIntervention = new Intervention(Convert.ToInt32(reader["id_demande"]),
    Convert.ToDateTime(reader["date_ouverture"]), reader["objet"].ToString(),
    reader["description"].ToString(), Convert.ToInt32(reader["niveau_urgence"]),
    Convert.ToDateTime(reader["date_resolution"]), reader["statut"].ToString());
    lesInterventions.Add(uneIntervention);
}

con.Close();

return lesInterventions;

}

public static void AjouterIntervention(Intervention intervention)
{
    MySqlConnection con = new MySqlConnection(connectionString);
    con.Open();

    MySqlCommand mySqlCommand = con.CreateCommand();

    mySqlCommand.CommandText = "INSERT INTO INTERVENTION VALUES
    (@id_demande, @date_Ouverture, @objet, @description, @Niveau_Urgence,
    @date_Resolution, @statut";

    mySqlCommand.Parameters.AddWithValue("@id_demande",
    intervention.getIdDemande());

    mySqlCommand.Parameters.AddWithValue("@date_Ouverture",
    intervention.getDateOuverture());

    mySqlCommand.Parameters.AddWithValue("@objet", intervention.getObjet());
    mySqlCommand.Parameters.AddWithValue("@description",
    intervention.getDescription());

    mySqlCommand.Parameters.AddWithValue("@Niveau_Urgence",

```

```

intervention.getNiveauUrgence());

mySqlCommand.Parameters.AddWithValue("@date_Resolution",
intervention.getDateResolution());

mySqlCommand.Parameters.AddWithValue("@statut", intervention.getStatut());

mySqlCommand.ExecuteNonQuery();

con.Close();

}

public static void AjouterMateriel(Materiel materiel)
{
    MySqlConnection con = new MySqlConnection(connectionString);

    con.Open();

    MySqlCommand mySqlCommand = con.CreateCommand();

    mySqlCommand.CommandText = "INSERT INTO matériel VALUES (@type, @modele,
    @date_acquisition, @etat";

    mySqlCommand.Parameters.AddWithValue("@type", materiel.GetType());

    mySqlCommand.Parameters.AddWithValue("@modele", materiel.getModele());

    mySqlCommand.Parameters.AddWithValue("@date_acquisition",
    materiel.getDateAcquisition());

    mySqlCommand.Parameters.AddWithValue("@etat", materiel.getEtat());

    mySqlCommand.ExecuteNonQuery();

    con.Close();

}

public static void AjouterUtilisateur(Utilisateur utilisateur)
{
    MySqlConnection con = new MySqlConnection(connectionString);

    con.Open();

```

```

MySqlCommand mySqlCommand = con.CreateCommand();

mySqlCommand.CommandText = "INSERT INTO utilisateur VALUES (@nom,
@prenom, @région, @email, @responsable";

mySqlCommand.Parameters.AddWithValue("@type", utilisateur.getNom());
mySqlCommand.Parameters.AddWithValue("@modele", utilisateur.getPrenom());
mySqlCommand.Parameters.AddWithValue("@date_acquisition",
utilisateur.getRegion());

mySqlCommand.Parameters.AddWithValue("@etat", utilisateur.getEmail());
mySqlCommand.Parameters.AddWithValue("@etat", utilisateur.getResponsable());

mySqlCommand.ExecuteNonQuery();

con.Close();

}

}

}

```

Programme Principale page Technicien :

```

private void btnValiderIncident_Click(object sender, EventArgs e)
{
    Intervention intervention = new Intervention(tbDéclatIncident.Text, "",
textBoxIdPerso.Text, dateTimePickerDateDecla.Value, Convert.ToInt32(textBoxIdMateriel.Text),
textBoxIdTech.Text);

    BD.AjouterIntervention(intervention);

    MessageBox.Show("Exécuté");
}

/// <summary>

/// Événement déclenché lorsqu'un technicien ajoute un matériel.

/// </summary>

```

```

private void btnConfirmerMatériel_Click(object sender, EventArgs e)
{
    bool garantie = rbGarantieTrue.Checked;

    // Correction : les contrôles de type TextBox contiennent des textes, donc utilisez leurs
    valeurs Text.

    Matériel nouveauMatériel = new Matériel(textBoxIdPerso.Text, textBoxIdTech.Text,
    Convert.ToDateTime(textBoxIdMatériel.Text), tbEtatInci.Text);

    BD.AjouterMatériel(nouveauMatériel); // Méthode renommée pour correspondre à celle
    de la classe BD.

    ActualiserMatériel();
}

/// <summary>
/// Fonction pour actualiser les ComboBox de matériel.
/// </summary>

public void ActualiserMatériel()
{
    // Effacez d'abord tous les éléments de la ComboBox

    cbConsultMatériel.Items.Clear();

    cbSuppMater.Items.Clear();

    // Correction : il est préférable de vérifier la validité de la liste obtenue.

    List<int> idMatériel = BD.getLesMateriaux();

    if (idMatériel != null)
    {
        foreach (int id in idMatériel)
        {
            cbConsultMatériel.Items.Add(id);

```



```

cbSuppMater.Items.Add(id);

}

}

}

/// <summary>

/// Fonction pour actualiser les ComboBox des techniciens.

/// </summary>

public void ActualiserTechnicien()

{

    cbSuppTech.Items.Clear();

    List<string> matriculesTechnicien = BD.GetMatriculeTechnicien();

    if (matriculesTechnicien != null)

    {

        foreach (string matricule in matriculesTechnicien)

        {

            cbSuppTech.Items.Add(matricule);

        }

    }

}

/// <summary>

/// Fonction pour actualiser les ComboBox des utilisateurs.

/// </summary>

public void ActualiserUtilisateur()

{

    cbSuppU.Items.Clear();

    List<string> idUtilisateur = BD.GetIdUtilisateur();

```

```

if (idUtilisateur != null)
{
    foreach (string id in idUtilisateur)
    {
        cbSuppU.Items.Add(id);
    }
}
}

/// <summary>
/// Événement déclenché lors de l'enregistrement d'un utilisateur.
/// </summary>

private void btnEnregU_Click(object sender, EventArgs e)
{
    // Correction : Assurez-vous que le constructeur Utilisateur correspond bien aux
    paramètres fournis.

    Utilisateur unUtilisateur = new Utilisateur(tbMatriculeU.Text, tbNomU.Text,
    dateTimePicker1.Value, tbRegion.Text, 0, Convert.ToInt32(numericUpDown1.Value),
    Convert.ToInt32(numericUpDown2.Value), tbAvan.Text,
    Convert.ToInt32(numericUpDown3.Value));

    BD.AjouterUtilisateur(unUtilisateur); // Méthode renommée pour correspondre à celle de
    la classe BD.

    BD.AjouterUtilisateur(unUtilisateur); // Méthode renommée pour correspondre à celle de
    la classe BD.

    ActualiserUtilisateur();
}

/// <summary>

```

```
/// Événement de chargement de la fenêtre principale.

/// </summary>

private void Form1_Load(object sender, EventArgs e)

{

    ActualiserMateriel();

    ActualiserTechnicien();

    ActualiserUtilisateur();

    // Correction : Vérifiez que la méthode affichIncident renvoie bien une liste d'objets
    Incident.

    List<Intervention> interventions = BD.affichIntervention();

    if (interventions != null)

    {

        cbConsInci.Items.Clear();

    }

}

}
```