

Documentation technique – Gestion d'appartements

Ce projet a une architecture MVC (Modèle – Vue – Contrôleur).

Modèle :

Il existe 2 types de classe modèle différentes, les classes métier ainsi que les classes DAO.

Les classes DAO contiennent toutes les requêtes SQL qui sont liées à la base de donnée.

La classe « connexionDB » est très importante car c'est elle qui permet au programme de se connecter à la base de donnée.

```
0.4 références
class ConnexionDB
{
    public static string connexion = @"server=localhost;userid=Lucas;password=;database=gestiondeappartement";
    public static MySqlConnection conn = null;

    35 références
    public static bool ConnexionBdd()
    {
        conn = new MySqlConnection(connexion);
        conn.Open();

        return (conn.State == System.Data.ConnectionState.Open);
    }
}
```

Il faut ensuite utiliser ses fonctions au début de chaque fonction de requête SQL.

Cette fonction contient une requête permettant de récupérer les informations de tous les propriétaires présents dans la base de donnée.

```

public static List<Proprietaire_modele> GetLesProprietaires()
{
    ConnexionDB.ConnexionBdd();
    List<Proprietaire_modele> LesProprietaires = new List<Proprietaire_modele>();
    Proprietaire_modele proprietaire;
    MySqlCommand objCmd;
    objCmd = ConnexionDB.conn.CreateCommand();
    string reqQuery;
    reqQuery = "SELECT * FROM proprietaires";
    objCmd.CommandText = reqQuery;
    MySqlDataReader rdr = objCmd.ExecuteReader();
    int numeroprop;
    string nom;
    string prenom;
    string adresse;
    string code_ville;
    string tel;

    while (rdr.Read())
    {
        numeroprop = (int)rdr["NUMEROPROP"];
        nom = (string)rdr["NOM"];
        prenom = (string)rdr["PRENOM"];
        adresse = (string)rdr["ADRESSE"];
        code_ville = (string)rdr["CODE_VILLE"];
        tel = (string)rdr["TEL"];
        proprietaire = new Proprietaire_modele(numeroprop, nom, prenom, adresse, code_ville, tel);
        LesProprietaires.Add(proprietaire);
    }

    rdr.Close();

    return LesProprietaires;
}

```

Contrôleur :

Les contrôleurs sert à synchroniser les modèles et la vue.

Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer.

Il peut donc appeler des fonctions des différents modèle et avertir la vue que des données ont changé pour que celle-ci se mette à jour.

```

public static AdherentDAO.Adherent UtilisateurConnecte { get; set; }

```

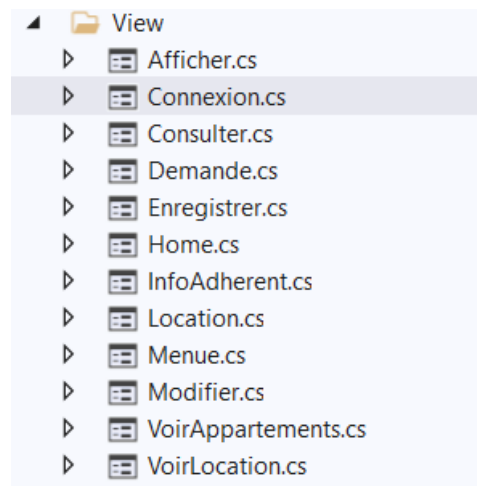
Vue :

La vue est le visuel de l'application, chaque vue est donc un form en c#.

Nous pouvons voir dans « home_view » des boutons qui ont chacun un nom permettant d'être appelé par les fonctions DAO dans le modele. Chaque form possède une `public partial class ConnexionFRM : Form`

Pour pouvoir coder chaque bouton,label, list qui se trouve dans une form

Il existe un fichier de vue pour chaque page, ou partie de page que nous souhaitons afficher.



Chaque nom de fichier termine par la fonction dont il est destiné :

- _controller,
- _modele ou DAO,
- _view.