TP .Net Maui n° 4 : Utilisation d'une base de données pour gérer des contacts

AppShell.xaml

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Shell
    x:Class="TPMauiApp4.AppShell"
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:TPMauiApp4.Views"
    Shell.FlyoutBehavior="Disabled">

    <ShellContent
        Title="Home"
        ContentTemplate="{DataTemplate local:vListe}"
        Route="MainPage" />

</Shell>
```

vListe.xaml

```xml
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="TPMauiApp4.Views.vListe"
             Title="Contacts">
    <ContentPage.ToolbarItems>
        <ToolbarItem Text="Ajouter" Clicked="Add_Clicked"
                     IconImageSource="{FontImage Glyph='A', Color=White, Size=22}" />
        <ToolbarItem Text="Supprimer" Clicked="Del_Clicked"
                     IconImageSource="{FontImage Glyph='D', Color=White, Size=22}" />
    </ContentPage.ToolbarItems>
    <VerticalStackLayout Spacing="20" Margin="10" >
        <!--<Label
            Text="Welcome to .NET MAUI!"
            VerticalOptions="Center"
            HorizontalOptions="Center" />-->
        <Label
            Text="?"
            x:Name="lbContacts"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Large" TextColor="{StaticResource Primary}"/>
        <CollectionView x:Name="cvListeContact" SelectionMode="Single">
            <CollectionView.ItemTemplate>
                <DataTemplate>
                    <VerticalStackLayout>
                        <HorizontalStackLayout Spacing="20" Margin="10" >
                            <Label Text="{Binding Nom}" FontSize="Large"
TextColor="{StaticResource Secondary}"/>
                            <Label Text="{Binding Email}" FontSize="Large"
TextColor="{StaticResource Primary}"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </DataTemplate>
            </CollectionView.ItemTemplate>
        </CollectionView>
```

```
        </VerticalStackLayout>
</ContentPage>
```

vListe.xaml.cs

```csharp
using Microsoft.Maui.Controls;
using TPMauiApp4.Models;

namespace TPMauiApp4.Views;

[QueryProperty(nameof(vListe.Name), nameof(vListe.Name))]
public partial class vListe : ContentPage
{
    public string Name
    {
        set
        {
            String sTmp = value;
            cContact NewCt = new cContact();
            NewCt.Nom = sTmp.Substring(0, sTmp.IndexOf("$$"));
            NewCt.Email = sTmp.Substring(sTmp.IndexOf("$$") + 2);
            _ListeContact.CreateContact(NewCt);
        }
    }

    public readonly cListeContact _ListeContact;
    public vListe()
      {
        Routing.RegisterRoute($"{nameof(vContact)}", typeof(vContact));
        _ListeContact = new cListeContact("MyBD");
        InitializeComponent();
      }

    protected override void OnAppearing()
    {
        base.OnAppearing();
        lbContacts.Text = "Nombre: " + _ListeContact.GetList().Count.ToString();
        cvListeContact.ItemsSource = _ListeContact.GetList();
    }

    private async void Add_Clicked(object sender, EventArgs e)
      {
        await Shell.Current.GoToAsync(nameof(vContact));

        OnAppearing();

        //var _contact1 = new cContact()
        //{
        //    Nom = "Chervy",
        //    Email = "chervy@3il.fr"
        //};
        //_ListeContact.CreateContact(_contact1);
        //OnAppearing();
    }
    private void Del_Clicked(object sender, EventArgs e)
    {

        if (cvListeContact.SelectedItem is null)
            return;
        var myContact = cvListeContact.SelectedItem as cContact;
        if (myContact is null)
            return;
        _ListeContact.DeleteContact(myContact);
        OnAppearing();
    }
}
```

vContact.xaml

```xml
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="TPMauiApp4.Views.vContact"
             Title="Contact">
    <VerticalStackLayout Spacing="20" Margin="10" >

        <Label
            Text="Ajouter un contact"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Large" TextColor="{StaticResource Primary}"/>
        <Label
            Text="Nom"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Small" TextColor="{StaticResource Primary}"/>
        <Editor x:Name="edtNom"
            />
        <Label
            Text="Mail"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Small" TextColor="{StaticResource Primary}"/>
        <Editor x:Name="edtEmail"
            />
        <Button
                x:Name="btSave"
                Text="Enregistrer"
                Clicked="btSave_Clicked"
                HorizontalOptions="Center" />
    </VerticalStackLayout>
</ContentPage>
```

vContact.xaml.cs

```csharp
namespace TPMauiApp4.Views;
using TPMauiApp4.Models;

public partial class vContact : ContentPage
{
    public vContact()
    {
      Routing.RegisterRoute($"{nameof(vListe)}", typeof(vListe));
      InitializeComponent();
    }

    private async void btSave_Clicked(object sender, EventArgs e)
    {
        //String sVal = $"test$$chervi2@3il.fr";
        String sVal = edtNom.Text + "$$" + edtEmail.Text;
        await Shell.Current.GoToAsync($"{nameof(vListe)}?{nameof(vListe.Name)}={sVal}");
    }
}
```

cContact.cs

```csharp
using SQLite;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```csharp
namespace TPMauiApp4.Models
{
    public class cContact
    {
        [PrimaryKey, AutoIncrement]
        public int Id { get; set; }

        public string Email { get; set; } = string.Empty;

        public string Nom { get; set; } = string.Empty;
    }
    public class cListeContact
    {
        private readonly SQLiteConnection _database;

        public SQLiteConnection GetSqlConnexion()
        {
            return _database;
        }
        public cListeContact(string dbName)
        {
            var dbPath =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData), dbName);
            _database = new SQLiteConnection(dbPath);
            _database.CreateTable<cContact>();
        }

        public List<cContact> GetList()
        {
            return _database.Table<cContact>().ToList();
        }
        public int CreateContact(cContact myContact)
        {
            return _database.Insert(myContact);
        }

        public int DeleteContact(cContact myContac)
        {
            return _database.Delete(myContac);
        }
    }
}
```