# **18 Работа с базами данных. Технологии доступа к данным**

Задание 1. CRUD–операции: ClientRepository с GetAllClientsAsync(), AddClientAsync(Client), UpdateClientAsync(Client), DeleteClientAsync(Client). Привязка: Clients в CRMViewModel как ObservableCollection<Client>, в XAML <ListView ItemsSource=“{Binding Clients}” />. Сохранение: в SaveClientsCommand выполнять await\_context.SaveChangesAsync().

Листинг программы:

using System;

using System.Windows;

using X.Views;

using X.Services;

using Microsoft.EntityFrameworkCore;

using X.ViewModels;

namespace \_1

{ public partial class App : Application

{ private readonly UserManager userManager;

public App()

{ userManager = new UserManager(); }

[STAThread]

public static void Main()

{ var app = new App();

app.InitializeComponent();

app.Run(); }

protected override void OnStartup(StartupEventArgs e)

{ base.OnStartup(e);

try

{ var loginWindow = new LoginWindow(userManager);

loginWindow.ShowDialog();

if (loginWindow.IsAuthenticated)

{ var clientRepository = new ClientRepository(new CRMContext(new DbContextOptions<CRMContext>()));

var crmViewModel = new CRMViewModel(clientRepository);

var mainWindow = new MainWindow(crmViewModel);

Current.MainWindow = mainWindow;

mainWindow.Show();

loginWindow.Close(); }

else

{ Shutdown(); } }

catch (Exception ex)

{

MessageBox.Show($"Ошибка запуска: {ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

Shutdown();

}

}

}

}

Анализ результатов:

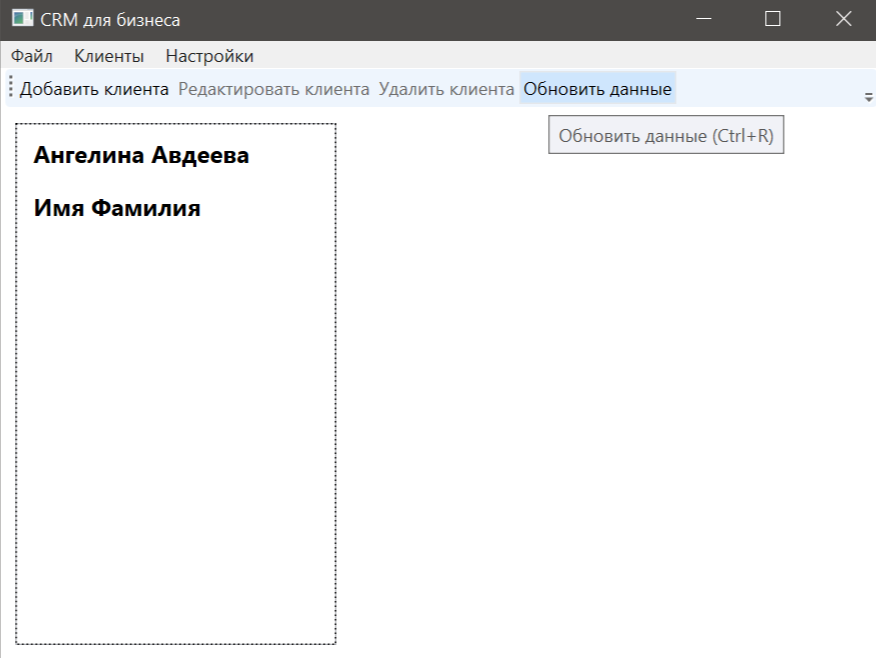


Рисунок 18.1 – Результат работы программы