**Лабораторная работа 2. Разработка представления данных по моделям**

**Цель работы:** Создание представления моделей и отображение данных

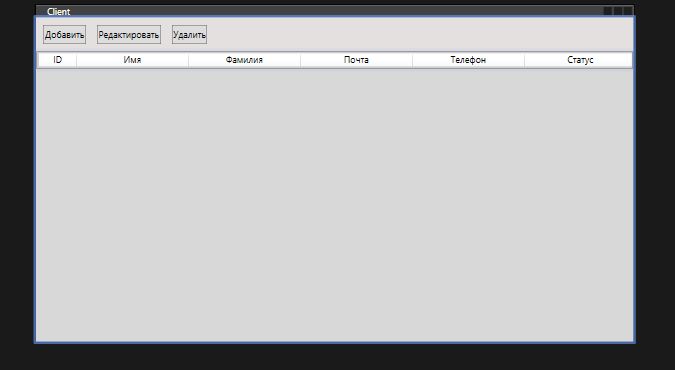


Рисунок 1 – Экранная форма клиент

<Window x:Class="WpfApp2.View.WindowClient"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp2.View"

mc:Ignorable="d"

Title="Client" Height="450" Width="800">

<StackPanel Background="#FFD8D8D8">

<StackPanel Orientation="Horizontal" Background="#FFE4E0E0">

<Button x:Name="btnAdd" Margin="10,10,5,10" Content="Добавить"

Height="25"

Click="btnAdd\_Click"/>

<Button x:Name="btnEdit" Margin="10,10,5,10" Content="Редактировать"

Height="25" Click="btnEdit\_Click"/>

<Button x:Name="btnDelete" Margin="10,10,5,10" Content="Удалить"

Height="25"

Click="btnDelete\_Click"/>

</StackPanel>

<ListView x:Name="WNPclient" Background="#FFD3CDCD">

<ListView.View>

<GridView>

<GridView.Columns>

<GridViewColumn Header="ID" Width="50"

DisplayMemberBinding="{Binding ID}" />

<GridViewColumn Header="Имя" Width="150"

DisplayMemberBinding="{Binding FirstName}" />

<GridViewColumn Header="Фамилия" Width="150"

DisplayMemberBinding="{Binding LastName}" />

<GridViewColumn Header="Почта" Width="150"

DisplayMemberBinding="{Binding Email}" />

<GridViewColumn Header="Телефон" Width="150"

DisplayMemberBinding="{Binding Phone}" />

<GridViewColumn Header="Статус" Width="150"

DisplayMemberBinding="{Binding Status}" />

</GridView.Columns>

</GridView>

</ListView.View>

</ListView>

</StackPanel>

</Window>

**C# код для клиента**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WpfApp2.Model

{

//Класс, содержит поля необходимые по заданию

class Client

{

public int ID { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Email { get; set; }

public long Phone { get; set; }

public string Status { get; set; }

public Client() { }

//Конструктор чтобы воспользоваться полями

public Client(int id, string firstname, string lastname, string email, long phone, string status)

{

this.ID = id;

this.FirstName = firstname;

this.LastName = lastname;

this.Email = email;

this.Phone = phone;

this.Status = status;

}

//Клон для тестов(сам по себе ничего не делает)

public Client ShallowCopy()

{

return (Client)this.MemberwiseClone();

}

}

}