

Модуль № 3: Проектирование и разработка информационных систем

Текст задания: Для выполнения задания рекомендуется создать в базе данных таблицу "Пользователи". Если такая таблица уже существует, необходимо внести некоторые изменения для реализации дальнейшего функционала приложения.

Разработайте форму для авторизации зарегистрированных пользователей с ролями "Администратор" и "Пользователь".

Форма должна содержать поля текстовые поля логин, пароль и кнопку "Войти".

Поля "Логин" и "Пароль" должны быть обязательными для заполнения.

При неверно введенных данных, пользователь должен получить сообщение об ошибке "Вы ввели неверный логин или пароль. Пожалуйста проверьте ещё раз введенные данные".

После успешной авторизации пользователь должен получить сообщение "Вы успешно авторизовались".

При аутентификации связка «логин/пароль» должна совпадать с одной из записей в таблице "Пользователи".

При первой успешной авторизации по выданному паролю администратором должна выводиться форма для смены пароля.

Форма должна включать текущий пароль, новый пароль, подтверждение нового пароля. Все поля обязательные для заполнения. После заполнения формы и нажатия кнопки "Изменить пароль", система должна проверить правильность введенного текущего пароля и совпадение нового пароля с подтверждением.

В случае ошибок при заполнении формы пользователю должно выводиться сообщение об ошибке. В случае успешного изменения пароля, пользователю должно выводиться сообщение об успешной смене пароля.

Если в течении 3-х раз подряд был неверно введен логин/пароль, то учетная запись блокируется и при повторном авторизации должно появляться сообщение "Вы заблокированы. Обратитесь к администратору".

Так же учетная запись должна блокироваться если пользователь не авторизовался в течении 1 месяца. – для ГЛАВНОГО ОКНА

На рабочем столе пользователя с ролью "Администратор" предусмотрите функционал для добавления новых пользователей, изменения данных текущих пользователей (включая снятие блокировки).

При добавлении нового пользователя следует проверять его наличие в базе данных.

В случае, если пользователь с указанным логином уже существует, должно выводиться соответствующее сообщение.

Графический интерфейс необходимо разработать в соответствии с требованиями к разработке.

Разработайте проектную документацию на разработанный функционал.

Включите описание функционального назначения, используемые методы с указанием параметров.

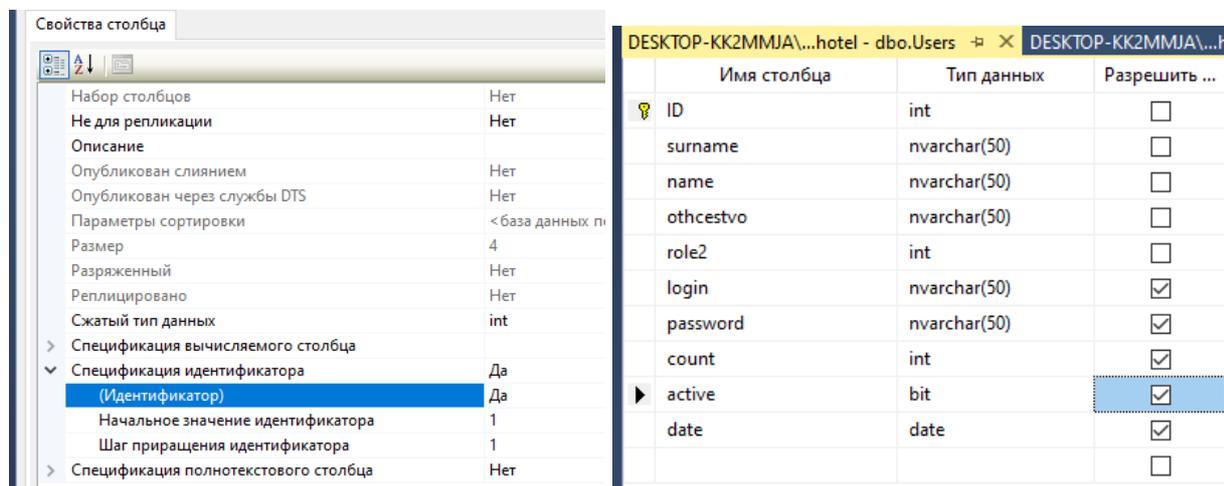
Необходимые приложения: Требования к разработке.pdf

На рабочем столе пользователя с ролью "Администратор" предусмотрите функционал для добавления новых пользователей, изменения данных текущих пользователей (включая снятие блокировки).

При добавлении нового пользователя следует проверять его наличие в базе данных.

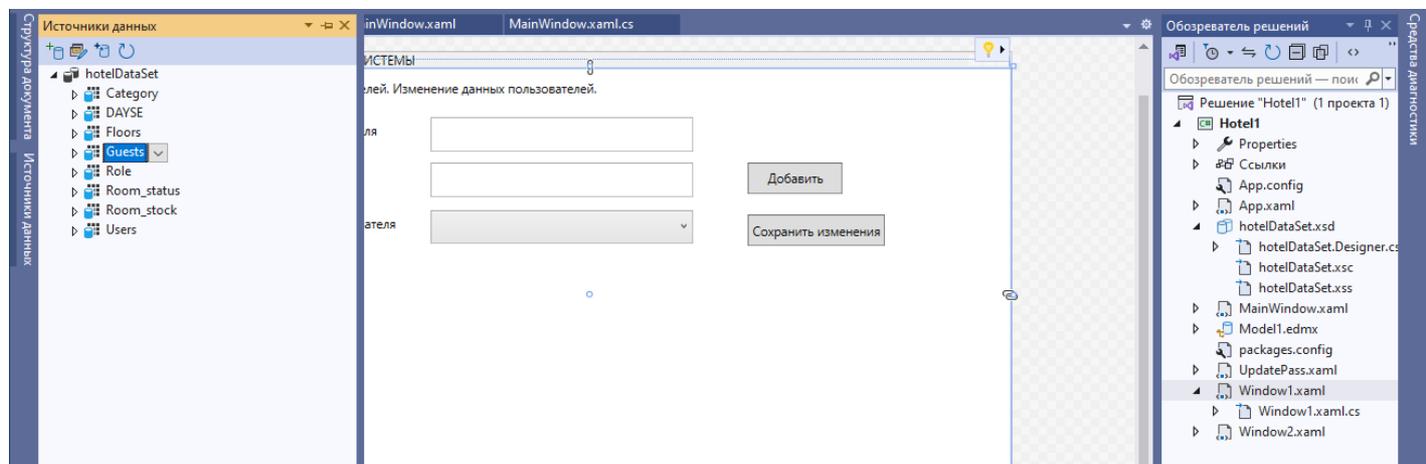
В случае, если пользователь с указанным логином уже существует, должно выводиться соответствующее сообщение.

В базе данных необходимо сделать поле ИД счётчиком! (Если не сделали этого ранее)



После внесения изменения базу нужно переподключить

Для привязки данных добавим нашу базу данных в Источники данных



Выполним настройку внешнего вида окна Администратора. - **Window1.xaml**

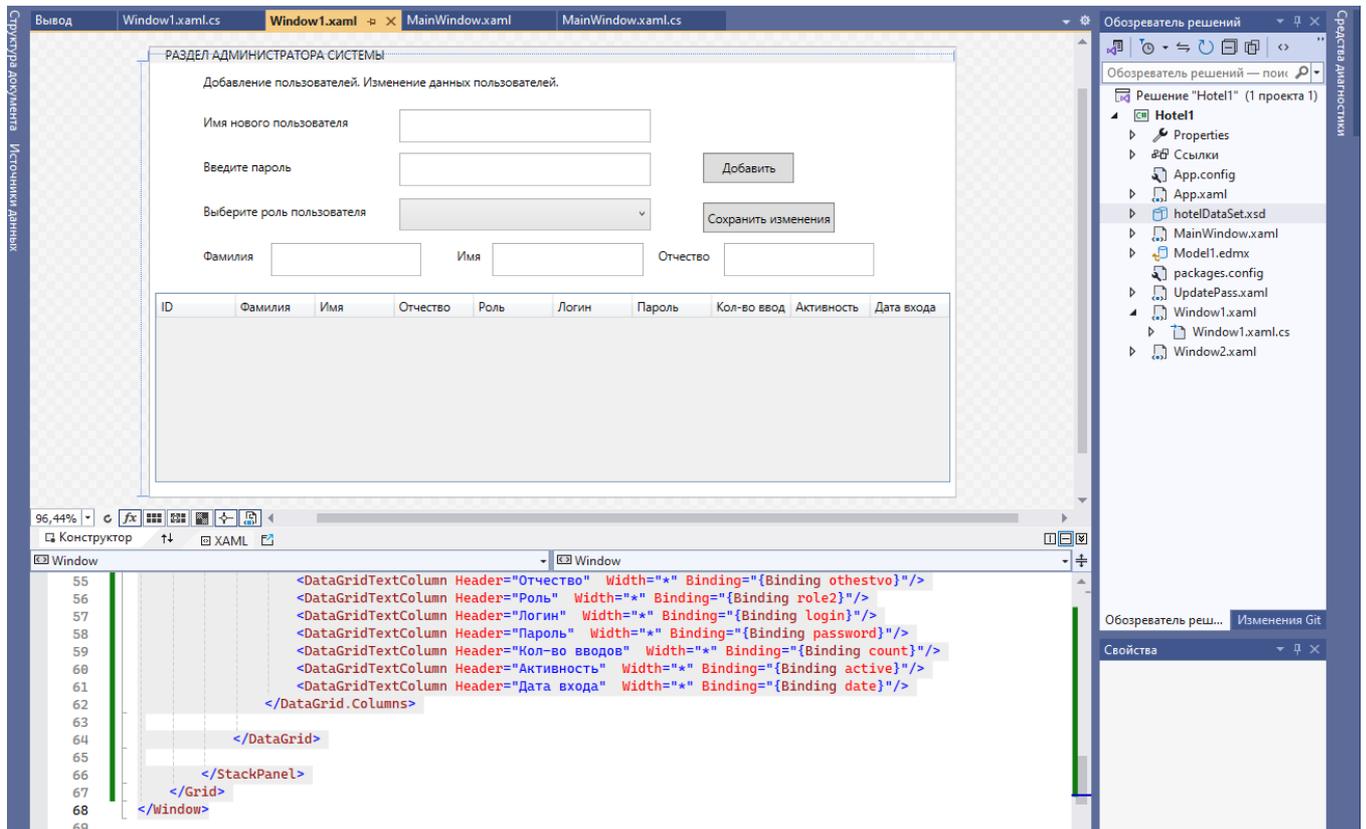
```
<Window x:Class="Hotel1.Window1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:Hotel1"
    mc:Ignorable="d"
    Title="РАЗДЕЛ АДМИНИСТРАТОРА СИСТЕМЫ" Height="450" Width="800">
    <Grid>
        <Label x:Name="Label1" Content="Добавление пользователей. Изменение данных
пользователей."
            HorizontalAlignment="Left" Height="30" Margin="48,6,0,0"
            VerticalAlignment="Top"></Label>
```

```

        <Label x:Name="Label2" Content="Имя нового пользователя" HorizontalAlignment="Left"
Height="40" Margin="48,46,0,0"
        VerticalAlignment="Top"></Label>
        <Label x:Name="Label3" Content="Введите пароль" HorizontalAlignment="Left" Height="30"
Margin="48,90,0,0"
        VerticalAlignment="Top"> </Label>
        <Label x:Name="Label4" Content="Выберите роль пользователя" HorizontalAlignment="Left"
Height="30" Margin="48,135,0,0"
        VerticalAlignment="Top"> </Label>
        <Label x:Name="Label5" Content="Фамилия" HorizontalAlignment="Left" Height="30"
Margin="48,180,0,0"
        VerticalAlignment="Top"></Label>
        <Label x:Name="Label6" Content="Имя" HorizontalAlignment="Left" Height="30"
Margin="300,180,0,0"
        VerticalAlignment="Top"></Label>
        <Label x:Name="Label7" Content="Отчество" HorizontalAlignment="Left" Height="30"
Margin="500,180,0,0"
        VerticalAlignment="Top"></Label>
        <TextBox x:Name="TextBox1" HorizontalAlignment="Left" Height="33" Width="250"
Margin="248,46,0,0"
        TextWrapping="Wrap" VerticalAlignment="Top"></TextBox>
        <TextBox x:Name="TextBox2" HorizontalAlignment="Left" Height="33" Width="250"
Margin="248,90,0,0"
        TextWrapping="Wrap" VerticalAlignment="Top"></TextBox>
        <TextBox x:Name="TextBox3" HorizontalAlignment="Left" Height="33" Width="150"
Margin="120,180,0,0"
        TextWrapping="Wrap" VerticalAlignment="Top"></TextBox>
        <TextBox x:Name="TextBox4" HorizontalAlignment="Left" Height="33" Width="150"
Margin="340,180,0,0"
        TextWrapping="Wrap" VerticalAlignment="Top"></TextBox>
        <TextBox x:Name="TextBox5" HorizontalAlignment="Left" Height="33" Width="150"
Margin="570,180,0,0"
        TextWrapping="Wrap" VerticalAlignment="Top"></TextBox>
        <ComboBox x:Name="typeRole" FontSize="14" HorizontalAlignment="Left" Height="33"
Width="250"
        VerticalAlignment="Top" Margin="248,135,0,0" >
        </ComboBox>
        <Button x:Name="button3" Content="Добавить" HorizontalAlignment="Left" Height="30"
Width="90"
        Margin="550,90,0,0" VerticalAlignment="Top" >
        </Button>
        <Button x:Name="button4" Content="Сохранить изменения" HorizontalAlignment="Left"
Height="30" Width="130"
        Margin="550,140,0,0" VerticalAlignment="Top" >
        </Button>

        <StackPanel Orientation="Vertical" HorizontalAlignment="Center" VerticalAlignment="Top"
Margin="5,230,5,5">
        <DataGrid Name="GridUser" AutoGenerateColumns="False"
Height="190" MaxWidth="750" ScrollViewer.CanContentScroll="True" >
<DataGrid.Columns>
        <DataGridTextColumn Header="ID" Width="*" Binding="{Binding ID}"/>
<DataGridTextColumn Header="Фамилия" Width="*" Binding="{Binding surname}"/>
        <DataGridTextColumn Header="Имя" Width="*" Binding="{Binding name}"/>
        <DataGridTextColumn Header="Отчество" Width="*" Binding="{Binding othcestvo}"/>
        <DataGridTextColumn Header="Роль" Width="*" Binding="{Binding role2}"/>
        <DataGridTextColumn Header="Логин" Width="*" Binding="{Binding login}"/>
        <DataGridTextColumn Header="Пароль" Width="*" Binding="{Binding password}"/>
        <DataGridTextColumn Header="Кол-во вводов" Width="*" Binding="{Binding count}"/>
<DataGridTextColumn Header="Активность" Width="*" Binding="{Binding active}"/>
        <DataGridTextColumn Header="Дата входа" Width="*" Binding="{Binding date}"/>
        </DataGrid.Columns>
        </DataGrid>
    </StackPanel>
</Grid>
</Window>

```



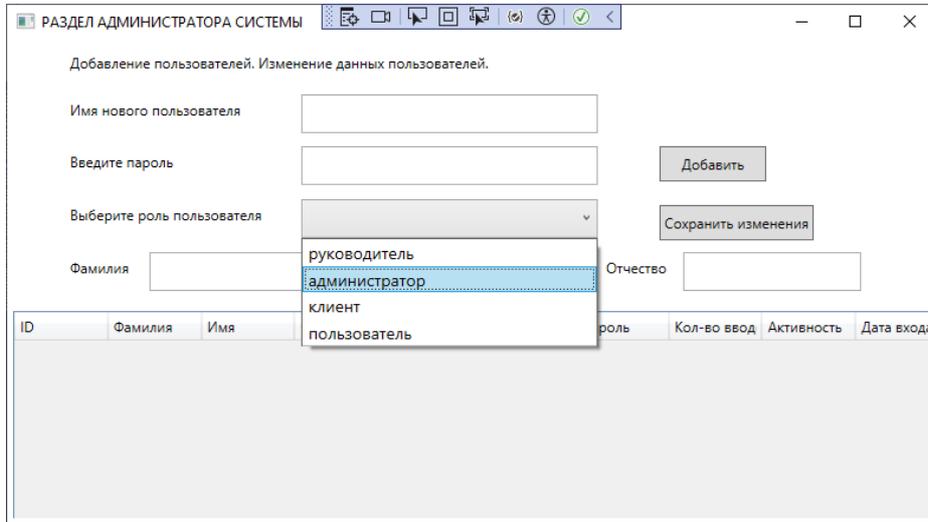
Выполним настройку программной части

```
// подключаемся к базе данных
hotelEntities db = new hotelEntities();
```

Передаём роли пользователей из базы данных в выпадающий список

```
var rol1 = (from x in db.Role select x).ToArray();
typeRole.DisplayMemberPath = "role1";
typeRole.SelectedValuePath = "ID";
typeRole.ItemsSource = rol1 ;
```

```
15 namespace Hotel1
16 {
17     /// <summary>
18     /// Логика взаимодействия для Window1.xaml
19     /// </summary>
20     — ссылки
21     public partial class Window1 : Window
22     {
23         // подключаемся к базе данных
24         hotelEntities db = new hotelEntities();
25         — ссылки
26         public Window1()
27         {
28             InitializeComponent();
29             var rol1 = (from x in db.Role select x).ToArray();
30             typeRole.DisplayMemberPath = "role1";
31             typeRole.SelectedValuePath = "ID";
32             typeRole.ItemsSource = rol1 ;
33         }
34     }
35 }
```



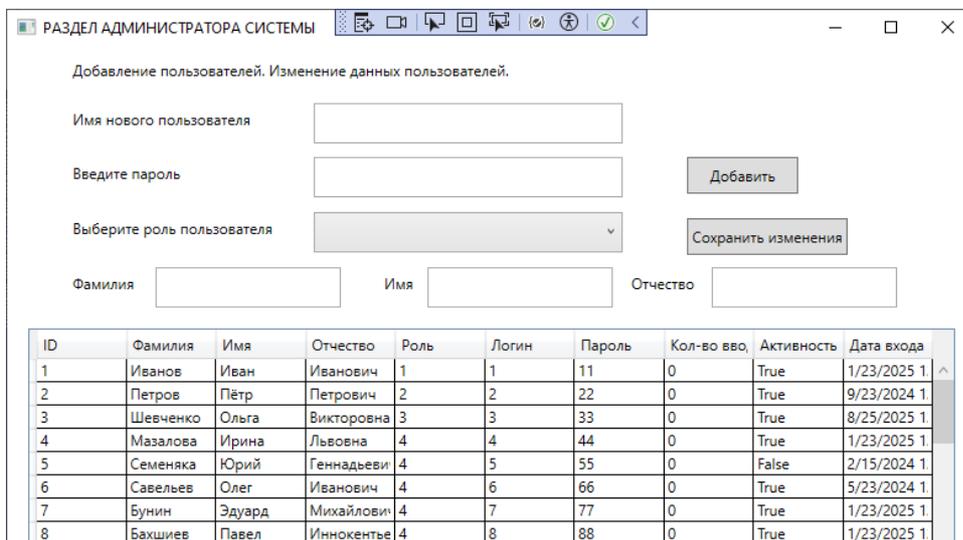
Выведем данные из таблицы Пользователь на форму

```
var user1 = (from u in db.Users select u).ToArray();
GridUser.ItemsSource = user1;
```

```

13  L  using System.Windows.Shapes;
14
15  namespace Hotel1
16  {
17      /// <summary>
18      /// Логика взаимодействия для Window1.xaml
19      /// </summary>
20      — ссылки
21      public partial class Window1 : Window
22      {
23          // подключаемся к базе данных
24          hotelEntities db = new hotelEntities();
25
26          — ссылки
27          public Window1()
28          {
29              InitializeComponent();
30              var rol1 = (from x in db.Role select x).ToArray();
31              typeRole.DisplayMemberPath = "role1";
32              typeRole.SelectedValuePath = "ID";
33              typeRole.ItemsSource = rol1 ;
34
35              var user1 = (from u in db.Users select u).ToArray();
36              GridUser.ItemsSource = user1;
37          }
38      }
39  }

```



При добавлении нового пользователя система должна проверять его на наличие и, если пользователь в системе уже существует, должно выводиться соответствующее сообщение.

Опишем обработчик события кнопки Добавить

```
private void button3_Click(object sender, RoutedEventArgs e)
{
    Users Users1 = new Users();
    var user2 = (from u in db.Users where u.login == TextBox1.Text
                select u).ToArray();
    // проверяем наличие пользователя
    if (user2.Length == 0)
    {
        // записываем данные из полей в таблицу бд
        Users1.login = TextBox1.Text;
        Users1.password = TextBox2.Text;
        Users1.role2 = (typeRole.SelectedItem as Role).ID;
        Users1.surname = TextBox3.Text;
        Users1.name = TextBox4.Text;
        Users1.othcestvo = TextBox5.Text;
        Users1.count = 0;
        Users1.active = true;
        Users1.date = null;
        db.Users.Add(Users1);
        db.SaveChanges();
        // обновление таблицы для вывода из базы
        var user1 = (from u in db.Users select u).ToArray();
        GridUser.ItemsSource = user1;
        MessageBox.Show("Данные успешно добавлены", "Уведомление",
            MessageBoxButton.OK, MessageBoxImage.Information);
    }
    else {
        MessageBox.Show("Такой пользователь уже существует",
            "!", MessageBoxButton.OK, MessageBoxImage.Error);
    }
}
```

```
30 var rol1 = (from x in db.Role select x).ToArray();
31 typeRole.DisplayMemberPath = "role1";
32 typeRole.SelectedValuePath = "ID";
33 typeRole.ItemsSource = rol1;
34
35 // создаём массив пользователей, передаём значения из бд в таблицу
36 var user1 = (from u in db.Users select u).ToArray();
37 GridUser.ItemsSource = user1;
38 }
39
40
41
42 private void button3_Click(object sender, RoutedEventArgs e)
43 {
44     Users Users1 = new Users();
45     var user2 = (from u in db.Users where u.login == TextBox1.Text
46                 select u).ToArray();
47     // проверяем наличие пользователя
48     if (user2.Length == 0)
49     {
50         // записываем данные из полей в таблицу бд
51         Users1.login = TextBox1.Text;
52         Users1.password = TextBox2.Text;
53         Users1.role2 = (typeRole.SelectedItem as Role).ID;
54         Users1.surname = TextBox3.Text;
55         Users1.name = TextBox4.Text;
56         Users1.othcestvo = TextBox5.Text;
57         Users1.count = 0;
58         Users1.active = true;
59         Users1.date = null;
60         db.Users.Add(Users1);
61         db.SaveChanges();
62         // обновление таблицы для вывода из базы
63         var user1 = (from u in db.Users select u).ToArray();
64         GridUser.ItemsSource = user1;
65         MessageBox.Show("Данные успешно добавлены", "Уведомление",
66             MessageBoxButton.OK, MessageBoxImage.Information);
67     }
68     else {
69         MessageBox.Show("Такой пользователь уже существует",
70             "!", MessageBoxButton.OK, MessageBoxImage.Error);
71     }
72 }
```

РАЗДЕЛ АДМИНИСТРАТОРА СИСТЕМЫ

Добавление пользователей. Изменение данных пользователей.

Имя нового пользователя

Введите пароль

Выберите роль пользователя

Фамилия Имя Отчество

ID	Фамилия	Имя	Отчество	Роль	Логин	Пароль	Кол-во вво	Активность	Дата входа
14	Гусев	Семен	Петрович	4	14	104	0	True	
15	Гладилина	Вера	Михайловна	4	15	105	0	True	
16	Масюк	Динара	Викторовна	4	16	106	0	True	
17	Лукин	Илья	Федорович	4	17	107	0	True	
18	Петров	Станислав	Игоревич	4	18	108	0	True	
19	Филь	Марина	Федоровна	4	19	109	0	True	
20	Михайлов	Игорь	Вадимович	4	20	110	0	True	
21	Ivanov	Ivan	Ivan	4	234	345	0	True	

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace Hotel1
{
    /// <summary>
    /// Логика взаимодействия для Window1.xaml
    /// </summary>
    public partial class Window1 : Window
    {
        // подключаемся к базе данных
        hotelEntities db = new hotelEntities();

        public Window1()
        {
            InitializeComponent();
            // создаём массив ролей, передаём значения из бд в выпадающий список
            var rol1 = (from x in db.Role select x).ToArray();
            typeRole.DisplayMemberPath = "role1";
            typeRole.SelectedValuePath = "ID";
            typeRole.ItemsSource = rol1;

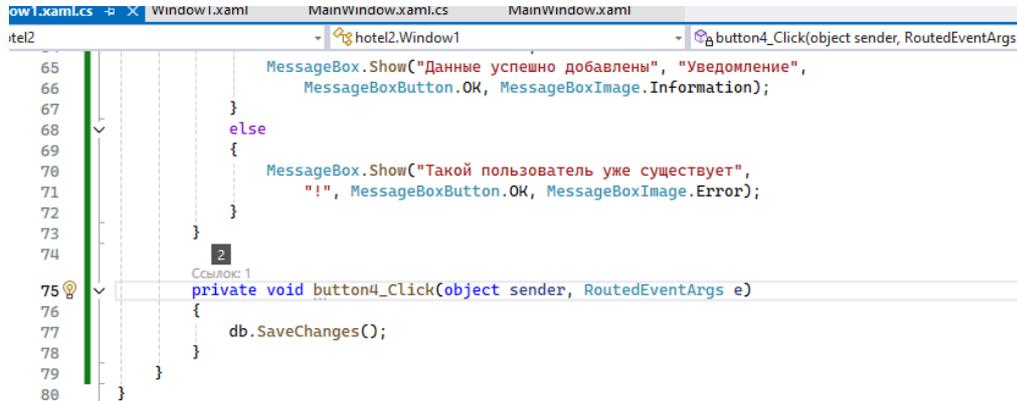
            // создаём массив пользователей, передаём значения из бд в таблицу
            var user1 = (from u in db.Users select u).ToArray();
            GridUser.ItemsSource = user1;
        }

        private void button3_Click(object sender, RoutedEventArgs e)
        {
            Users Users1 = new Users();

```


Сохраняем все изменения в таблице в базу данных

```
private void button4_Click(object sender, RoutedEventArgs e)
{
    db.SaveChanges();
}
```



КОД страницы администратора

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using System.Data.Entity;

namespace hotel2
{
    /// <summary>
    /// Логика взаимодействия для Window1.xaml
    /// </summary>
    public partial class Window1 : Window
    {
        // подключаемся к базе данных
        hotelEntities db = new hotelEntities();

        public Window1()
        {
            InitializeComponent();
            var rol1 = (from x in db.Role select x).ToArray();
            typeRole.DisplayMemberPath = "role1";
            typeRole.SelectedValuePath = "ID";
            typeRole.ItemsSource = rol1;
            db.Users.Load();
            typeRole.ItemsSource = db.Users.Local.ToBindingList();

            var user1 = (from u in db.Users select u).ToArray();
            GridUser.ItemsSource = user1;
        }

        private void button3_Click(object sender, RoutedEventArgs e)
        {
            Users Users1 = new Users();
        }
    }
}
```

```

var user2 = (from u in db.Users
             where u.login == TextBox1.Text
             select u).ToArray();
// проверяем наличие пользователя
if (user2.Length == 0)
{
    // записываем данные из полей в таблицу бд
    Users1.login = TextBox1.Text;
    Users1.password = TextBox2.Text;
    Users1.role2 = (typeRole.SelectedItem as Role).ID;
    Users1.surname = TextBox3.Text;
    Users1.name = TextBox4.Text;
    Users1.otchestvo = TextBox5.Text;
    Users1.count = 0;
    Users1.active = true;
    Users1.date = null;
    db.Users.Add(Users1);
    db.SaveChanges();
    // обновление таблицы для вывода из базы
    var user1 = (from u in db.Users select u).ToArray();
    GridUser.ItemsSource = user1;
    MessageBox.Show("Данные успешно добавлены", "Уведомление",
                    MessageBoxButton.OK, MessageBoxImage.Information);
}
else
{
    MessageBox.Show("Такой пользователь уже существует",
                    "!", MessageBoxButton.OK, MessageBoxImage.Error);
}
}

private void button4_Click(object sender, RoutedEventArgs e)
{
    db.SaveChanges();
}
}
}

```