

### 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.

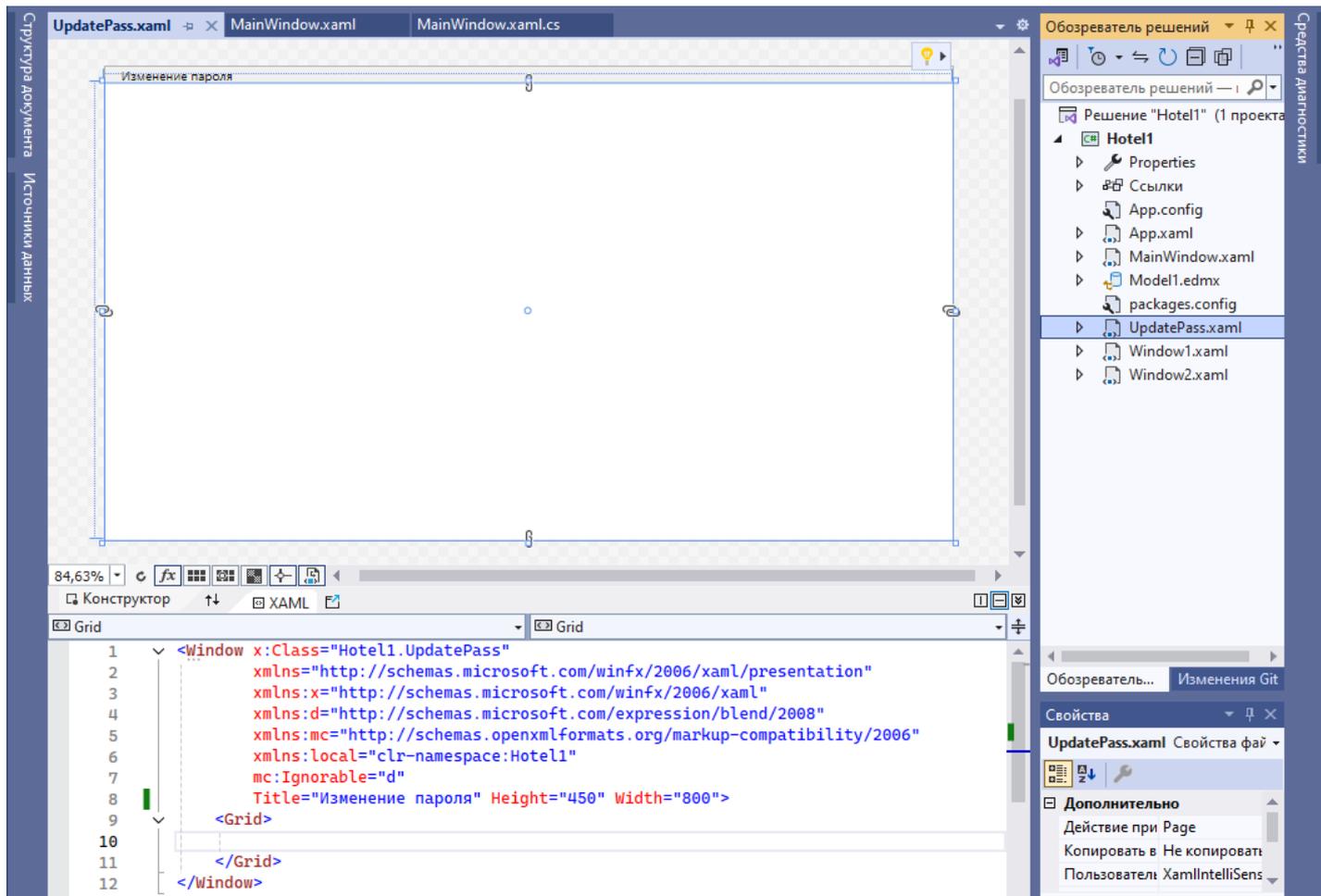
Необходимые приложения: Требования к разработке.pdf

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.

Создадим новое окно для формы смены пароля - **UpdatePass.xaml**



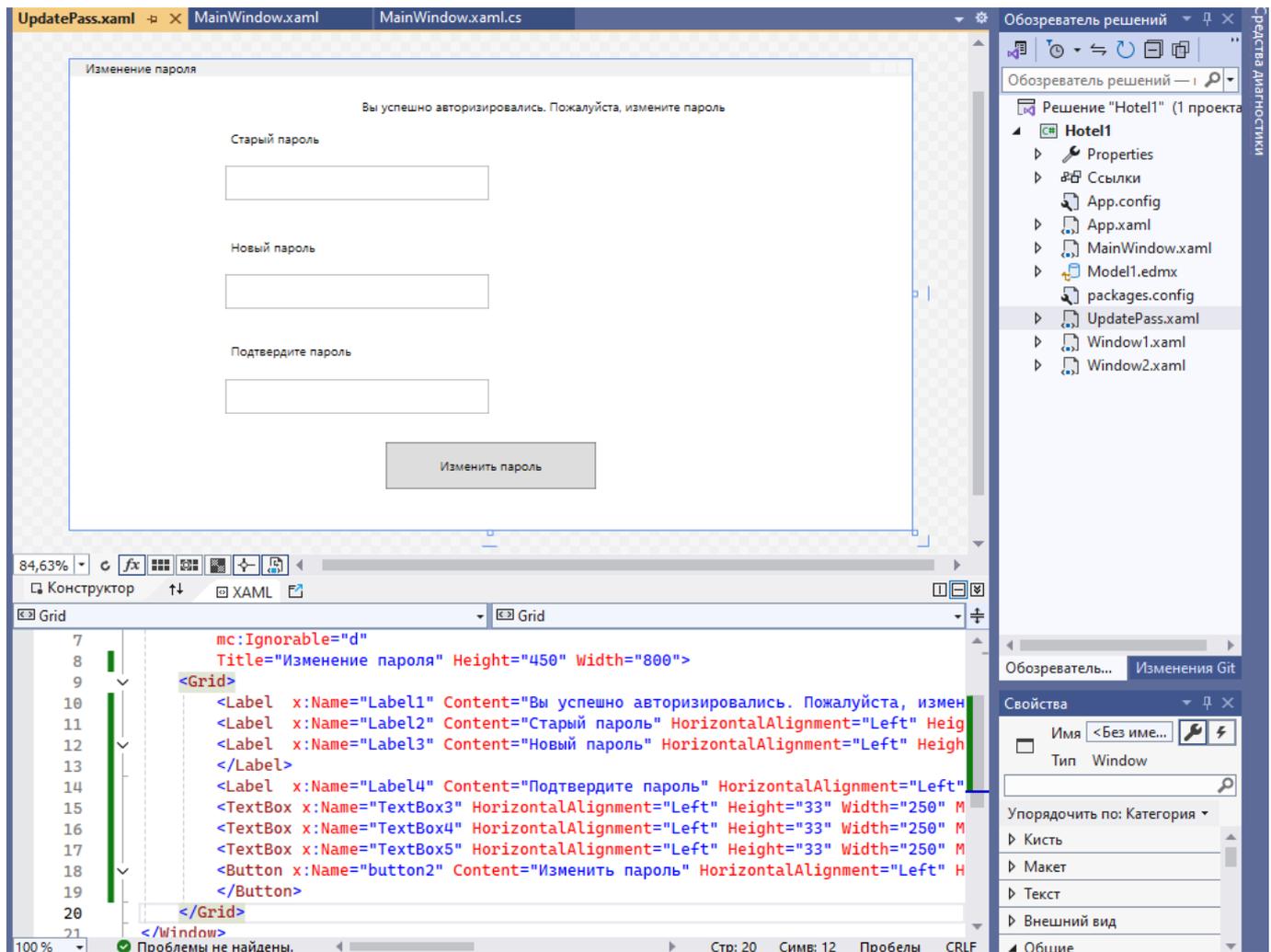
Поместим на форму 4 элемента «Label», 3 элемента «TextBox» и один элемент «Button»

```
<Label x:Name="Label1" Content="Вы успешно авторизировались. Пожалуйста, измените
пароль" HorizontalAlignment="Center" Height="40" Margin="100,16,0,0"
VerticalAlignment="Top"></Label>
<Label x:Name="Label2" Content="Старый пароль" HorizontalAlignment="Left" Height="40"
Margin="148,46,0,0" VerticalAlignment="Top"></Label>
<Label x:Name="Label3" Content="Новый пароль" HorizontalAlignment="Left" Height="43"
Margin="148,150,0,0" VerticalAlignment="Top">
</Label>
<Label x:Name="Label4" Content="Подтвердите пароль" HorizontalAlignment="Left"
Height="40" Margin="148,250,0,0" VerticalAlignment="Top"></Label>
<TextBox x:Name="TextBox3" HorizontalAlignment="Left" Height="33" Width="250"
Margin="148,86,0,0" TextWrapping="Wrap" VerticalAlignment="Top"></TextBox>
<TextBox x:Name="TextBox4" HorizontalAlignment="Left" Height="33" Width="250"
Margin="148,190,0,0" TextWrapping="Wrap" VerticalAlignment="Top"></TextBox>
```

```

<TextBox x:Name="TextBox5" HorizontalAlignment="Left" Height="33" Width="250"
Margin="148,290,0,0" TextWrapping="Wrap" VerticalAlignment="Top"></TextBox>
<Button x:Name="button2" Content="Изменить пароль" HorizontalAlignment="Left"
Height="45" Width="200" Margin="300,350,0,0" VerticalAlignment="Top" >
</Button>

```



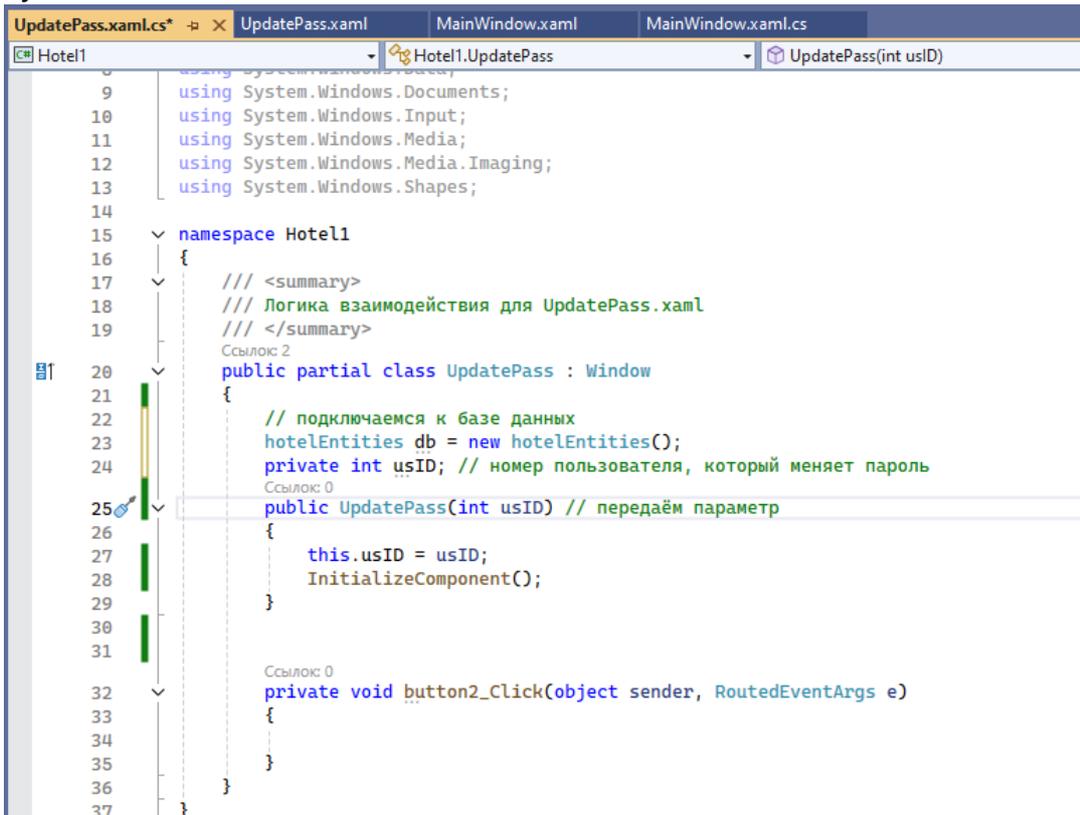
Для того, чтобы между формами была передача данных о текущем пользователе, создадим на главной форме глобальную переменную - номер редактируемого пользователя.

```
private int usID;
```



На форме смены пароля загружаем номер пользователя, который меняет пароль, подключаемся к базе данных

```
public partial class UpdatePass : Window
{
    // подключаемся к базе данных
    hotelEntities db = new hotelEntities();
    private int usID; // номер пользователя, который меняет пароль
    public UpdatePass(int usID) // передаём параметр
    {
        this.usID = usID;
        InitializeComponent();
    }
}
```



```
private void button2_Click(object sender, RoutedEventArgs e)
{
    // создаём массив пользователей из базы данных
    // загружаем данные пользователя, у которого ид равен передаваемому с формы авторизации
    var user = (from x in db.Users
                where x.ID == usID select x).ToArray();
    // если старый и новый пароли совпадают
    if (user[0].password == TextBox3.Text)
    {
        // если новый и подтверждение пароля совпадают
        if (TextBox4.Text == TextBox5.Text)
        {
            // меняем старый пароль на новый в базе данных
            user[0].password = TextBox4.Text;
            db.SaveChanges();
            MessageBox.Show("Пароль успешно изменён",
                "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);
            // сохраняем новую дату входа
            user[0].date = DateTime.Now;
            // обнуляем счётчик неправильных попыток входа
            user[0].count = 0;
            db.SaveChanges();
        }
    }
}
```

```

        // переходим на окно пользователя
        Window2 window1 = new Window2();
        window1.Show();
    }
    else
    {
        MessageBox.Show("Введённые пароли не совпадают!", "Пожалуйста проверьте ещё раз
введённые данные",
            MessageBoxButton.OK, MessageBoxImage.Error);
    }
}
else
{
    MessageBox.Show("Старый пароль не совпадает!", "Пожалуйста проверьте ещё раз введённые
данные",
        MessageBoxButton.OK, MessageBoxImage.Error);
}
}
}

```

```

33
34
35
36
37 // создаём массив пользователей из базы данных
38 // загружаем данные пользователя, у которого ид равен передаваемому с формы авторизации
39 var user = (from x in db.Users
40             where x.ID == usID select x).ToArray();
41 // если старый и новый пароли совпадают
42 if (user[0].password == TextBox3.Text)
43 {
44     // если новый и подтверждение пароля совпадают
45     if (TextBox4.Text == TextBox5.Text)
46     {
47         // меняем старый пароль на новый в базе данных
48         user[0].password = TextBox3.Text;
49         db.SaveChanges();
50         MessageBox.Show("Пароль успешно изменён",
51             "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);
52         // сохраняем новую дату входа
53         user[0].date = DateTime.Now;
54         // обнуляем счётчик неправильных попыток входа
55         user[0].count = 0;
56         db.SaveChanges();
57         // переходим на окно пользователя
58         Window2 window1 = new Window2();
59         window1.Show();
60     }
61     else
62     {
63         MessageBox.Show("Введённые пароли не совпадают!", "Пожалуйста проверьте ещё раз введённые данные",
64             MessageBoxButton.OK, MessageBoxImage.Error);
65     }
66 }
67 else
68 {
69     MessageBox.Show("Старый пароль не совпадает!", "Пожалуйста проверьте ещё раз введённые данные",
70         MessageBoxButton.OK, MessageBoxImage.Error);
71 }
72 }
73 }
74 }
75

```

Организуем переход пользователя при первом входе из окна авторизации на окно смены пароля

Проверяем дату последнего входа – если она равна нулю, то пользователь заходит в систему первый раз

```

if (user[0].date == null)
{
    usID = user[0].ID;
    MessageBox.Show("Добро пожаловать. При первом входе в систему необходимо изменить пароль",
        "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);
    UpdatePass window3 = new UpdatePass(usID);
    window3.Show();
}
}

```

```

UpdatePass.xaml.cs | UpdatePass.xaml | MainWindow.xaml | MainWindow.xaml.cs
Hotel1.MainWindow
Button1_Click(object sender, RoutedEventArgs e)
53     if (TextBox2.Text == user[0].password)
54     {
55         if (user[0].date != null)
56         {
57             // вычисляем количество дней, прошедших после последнего входа пользователя
58             DateTime d1 = DateTime.Now;
59             DateTime d2 = Convert.ToDateTime(user[0].date);
60             TimeSpan d = d1 - d2;
61
62             // если прошло более 30 дней, блокируем запись
63             if (Convert.ToInt32(d.ToString("dd")) > 30)
64             {
65                 user[0].active = false;
66                 db.SaveChanges();
67             }
68         }
69
70         // проверяем статус активности пользователя
71         if (user[0].active == true)
72         {
73             // организуем вход в соответствии с ролью пользователя
74             switch (user[0].role2)
75             {
76             case 1:
77                 MessageBox.Show("Добро пожаловать. Вы успешно авторизировались как администратор");
78                 Window1 window = new Window1();
79                 window.Show();
80                 break;
81             case 4:
82                 if (user[0].date == null)
83                 {
84                     usID = user[0].ID;
85                     MessageBox.Show("Добро пожаловать. При первом входе в систему необходимо ввести пароль");
86                     UpdatePass window3 = new UpdatePass(usID);
87                     window3.Show();
88                 }
89                 else
90                 {
91                     MessageBox.Show("Добро пожаловать. Вы успешно авторизировались как пользователь");
92                 }
93             }
94         }
95     }
96 }

```

### Код окна ИЗМЕНЕНИЯ ПАРОЛЯ

```

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;

namespace Hotel1
{
    /// <summary>
    /// Логика взаимодействия для UpdatePass.xaml
    /// </summary>
    public partial class UpdatePass : Window
    {
        // подключаемся к базе данных
        hotelEntities db = new hotelEntities();
        private int usID; // номер пользователя, который меняет пароль
        public UpdatePass(int usID) // передаём параметр
        {
            InitializeComponent();
            this.usID = usID;
        }

        private void button2_Click(object sender, RoutedEventArgs e)
        {
            // создаём массив пользователей из базы данных
            // загружаем данные пользователя, у которого ид равен передаваемому с формы
            // авторизации

```

```

var user = (from x in db.Users
            where x.ID == usID select x).ToArray();
// если старый и новый пароли совпадают
if (user[0].password == TextBox3.Text)
{
    // если новый и подтверждение пароля совпадают
    if (TextBox4.Text == TextBox5.Text)
    {
        // меняем старый пароль на новый в базе данных
        user[0].password = TextBox4.Text;
        db.SaveChanges();
        MessageBox.Show("Пароль успешно изменён",
            "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);
        // сохраняем новую дату входа
        user[0].date = DateTime.Now;
        // обнуляем счётчик неправильных попыток входа
        user[0].count = 0;
        db.SaveChanges();
        // переходим на окно пользователя
        Window2 window1 = new Window2();
        window1.Show();
    }
    else
    {
        MessageBox.Show("Введённые пароли не совпадают!", "Пожалуйста проверьте ещё
раз введённые данные",
            MessageBoxButton.OK, MessageBoxImage.Error);
    }
}
else
{
    MessageBox.Show("Старый пароль не совпадает!", "Пожалуйста проверьте ещё раз
введённые данные",
        MessageBoxButton.OK, MessageBoxImage.Error);
}
}
}
}

```

### Код окна АВТОРИЗАЦИЯ

```

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.Navigation;
using System.Windows.Shapes;

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

```

```

public MainWindow()
{
    InitializeComponent();
}
private int usID;
private void Button1_Click(object sender, RoutedEventArgs e)
{
    // организуем обработчик ошибок кода
    try
    {
        // создаём массив пользователей из базы данных
        var user = (from x in db.Users
                    where x.login == TextBox1.Text
                    select x).ToArray();

        // проверяем условие входа – поля логин и пароль должны быть не пустыми
        if (TextBox1.Text.Length == 0 || TextBox2.Text.Length == 0)
        {
            // если одно из полей пустое – выводим сообщение об ошибке
            MessageBox.Show("Поля Логин и Пароль обязательны для заполнения",
                "!", MessageBoxButton.OK, MessageBoxImage.Error);
        }
        else
        {
            // проверяем правильность введённого логина
            if (TextBox1.Text == user[0].login)
            {
                // проверяем правильность введённого пароля для выбранного пользователя
                if (TextBox2.Text == user[0].password)
                {
                    if (user[0].date != null)
                    {
                        // вычисляем количество дней, прошедших после последнего входа
                        // пользователя
                        DateTime d1 = DateTime.Now;
                        DateTime d2 = Convert.ToDateTime(user[0].date);
                        TimeSpan d = d1 - d2;

                        // если прошло более 30 дней, блокируем запись
                        if (Convert.ToInt32(d.ToString("dd")) > 30)
                        {
                            user[0].active = false;
                            db.SaveChanges();
                        }
                    }

                    // проверяем статус активности пользователя
                    if (user[0].active == true)
                    {
                        // организуем вход в соответствии с ролью пользователя
                        switch (user[0].role2)
                        {
                            case 1:
                                MessageBox.Show("Добро пожаловать. Вы успешно
авторизировались как администратор " + user[0].surname, "Уведомление", MessageBoxButton.OK,
MessageBoxImage.Information);

                                Window1 window = new Window1();
                                window.Show();
                                break;
                            case 4:
                                if (user[0].date == null)
                                {
                                    usID = user[0].ID;
                                    MessageBox.Show("Добро пожаловать. При первом входе
в систему необходимо изменить пароль" ,
                                        "Уведомление", MessageBoxButton.OK,
                                        MessageBoxImage.Information);

                                    UpdatePass window3 = new UpdatePass(usID);
                                    window3.Show();
                                }
                            default:
                                break;
                        }
                    }
                }
            }
        }
    }
    catch { }
}

```

```

        }
        else
        {
            MessageBox.Show("Добро пожаловать. Вы успешно
авторизировались как пользователь " + user[0].surname, "Уведомление", MessageBoxButton.OK,
MessageBoxImage.Information);
            Window2 window1 = new Window2();
            window1.Show();
        }
        break;
    default:
        MessageBox.Show("Данные не обнаружены", "Уведомление",
MessageBoxButton.OK, MessageBoxImage.Information);
        break;
    }
    // записываем текущую дату входа
    user[0].date = DateTime.Now;
    user[0].count = 0;
    db.SaveChanges();
}
else
{
    MessageBox.Show("Вы заблокированы. Обратитесь к администратору
системы.", "Уведомление",
        MessageBoxButton.OK, MessageBoxImage.Information);
}
}
else
{
    // проверяем количество неправильных вводов пароля
    if (user[0].count > 2)
    {
        user[0].active = false;
        db.SaveChanges();
        MessageBox.Show("Вы заблокированы. Обратитесь к администратору
системы.",
            "Уведомление", MessageBoxButton.OK,
MessageBoxImage.Information);
    }
    else
    {
        MessageBox.Show("Вы ввели неверный логин или пароль",
"Пожалуйста проверьте ещё раз введенные данные",
        MessageBoxButton.OK, MessageBoxImage.Error);
        user[0].count++;
        db.SaveChanges();
    }
}
}
}
}

catch (SystemException)
{
    MessageBox.Show("Ошибка системы", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
}
}
}
}

```