**Приложение «Почтовое отправление»**

Разработать приложение «Почтовое отправление» с помощью Visual Studio 2019, приложнения Wpf .NET Framework.

1.Создаем проект в Visual Studio 2019, приложнение Wpf .NET Framework.

2. В форме настроить дизайн по примеру ниже.

Для создания формы авторизации и регистрации требуется открыть Visual Studio 2019 и создать проект WpfApp1(.NET Framework.), как показано на рисунке 1.

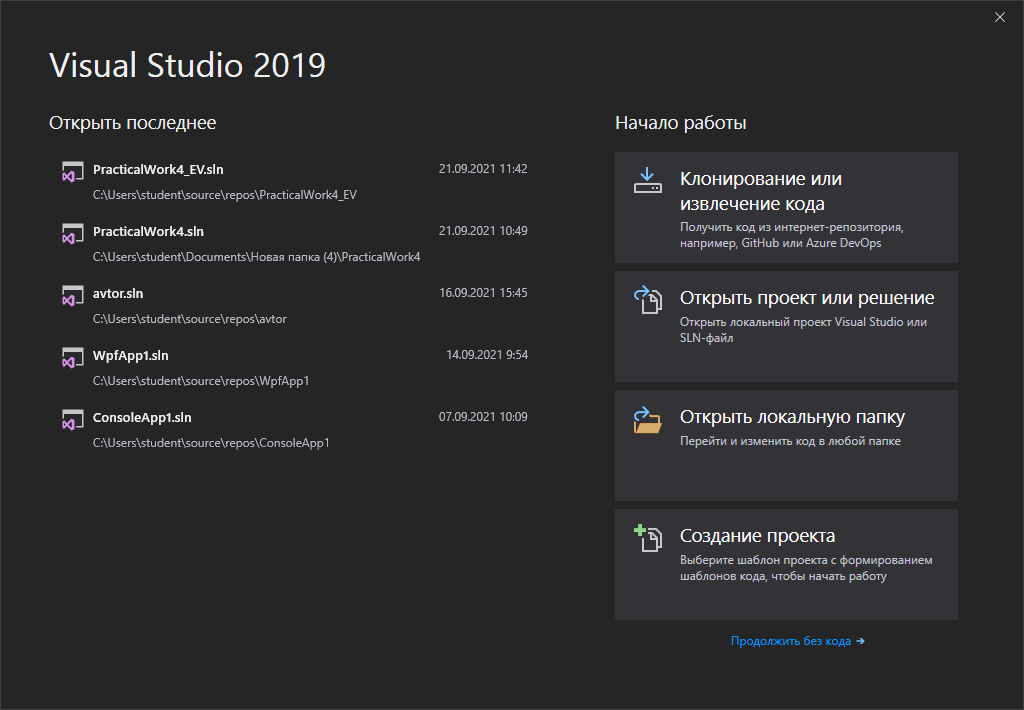


Рисунок 1 – Создание проекта

Открывается пустая форма, для создания окна авторизации, нужно задать размеры окна тэгами Height (Длина окна) Weight (Ширина окна).

Все манипуляции для дизайна выполняем в MainWindows.xaml

Title="MainWindow" Height="400" Width="300" WindowStartupLocation="CenterScreen">

Форма изменилась в размерах, как показано на рисунке 2.

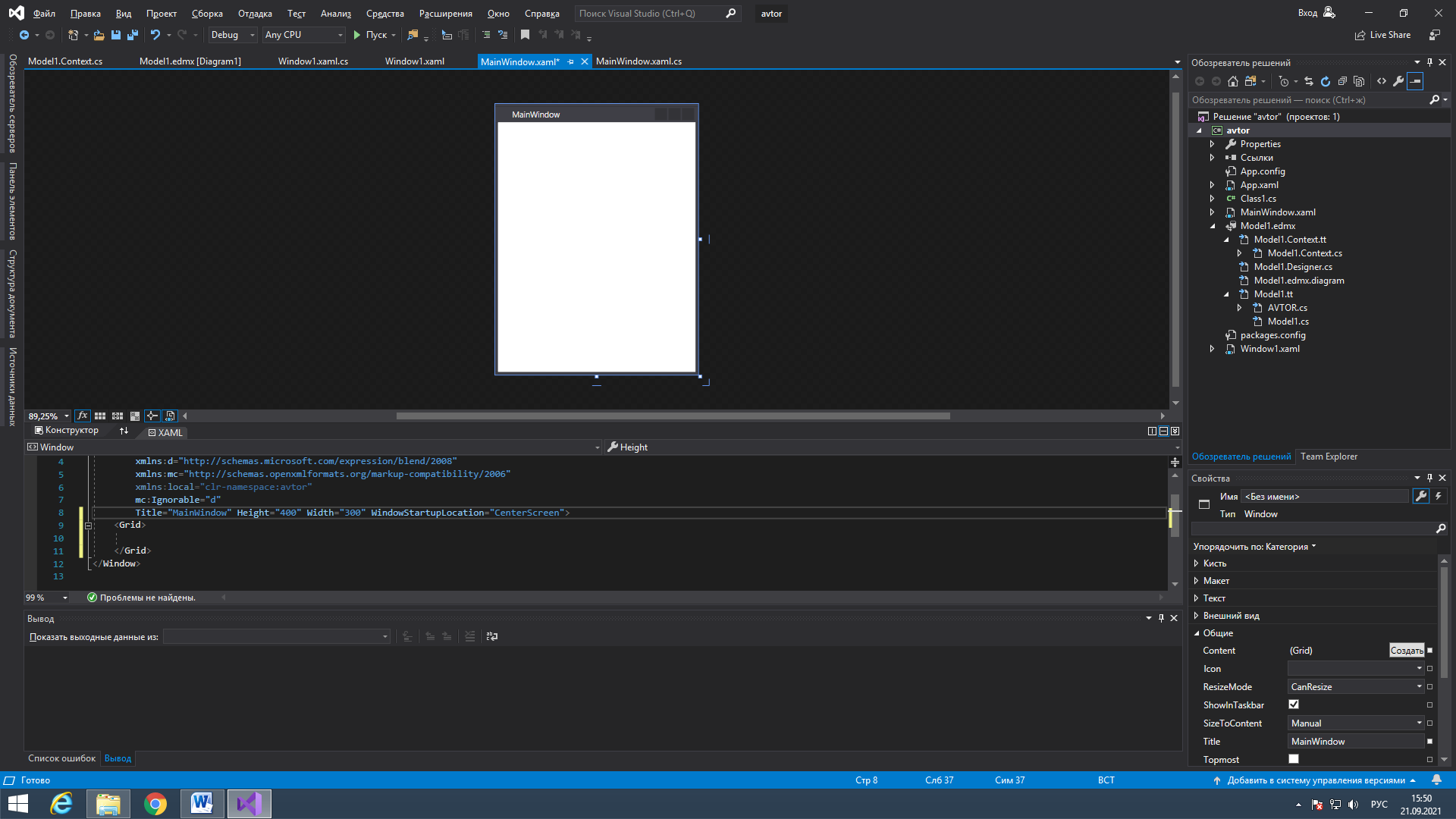


Рисунок 2 – Изменение размеров формы

Задаем задний фон формы прописав в тэге “Grid”, команду Background="#FFA3D8A3",

<Grid Background="#FFA3D8A3">

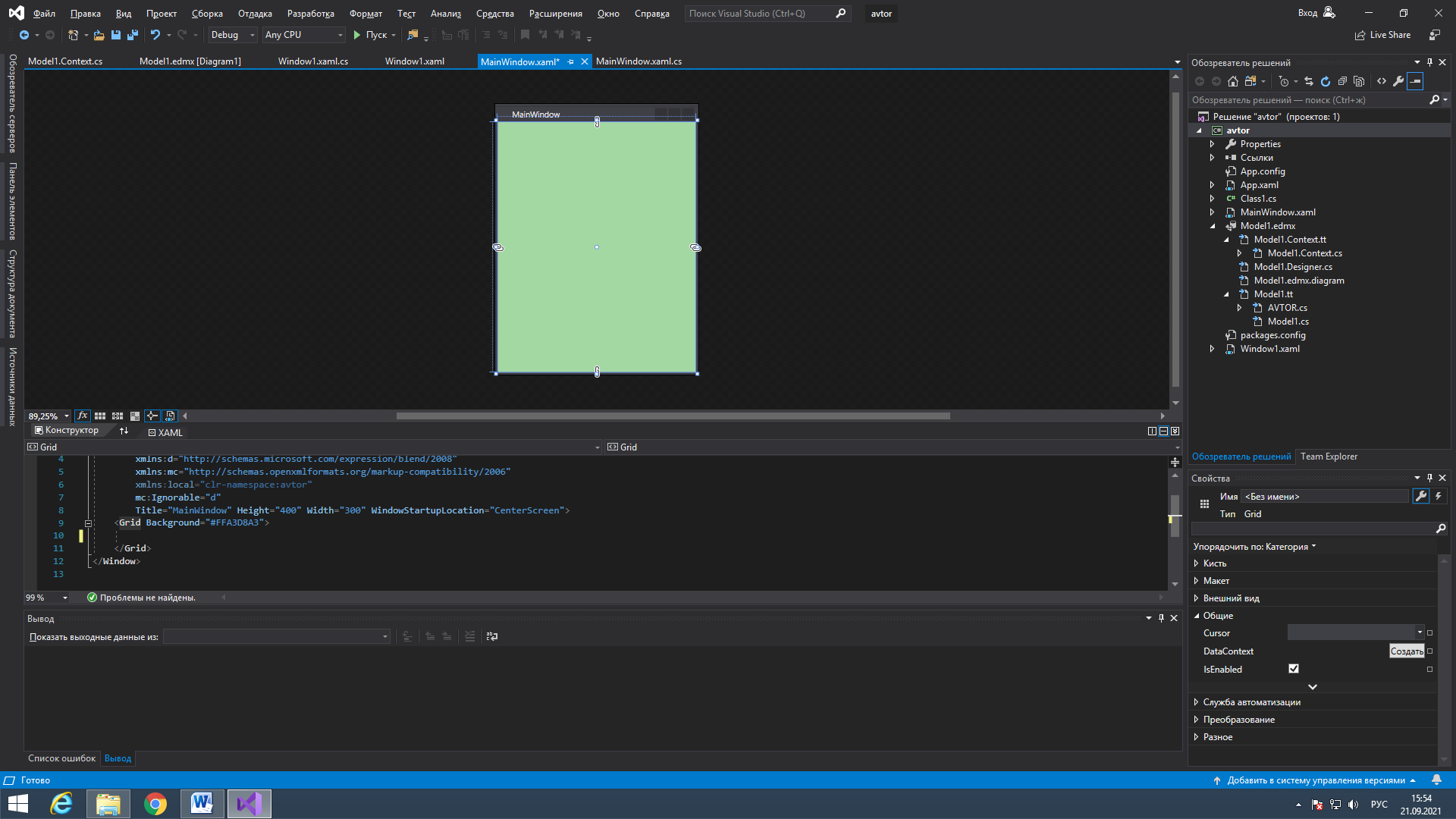


Рисунок 3 – Изменение цвета формы

- Тэгом <Label Content="Авторизация" FontSize="25" Margin="63,45,57,276" /> задаем текстовую метку на форме, тэг “FontSize” отвечает за размер шрифта, в “Content” вписывается желаемое слово или сочетание, как представлено на рисунке 4.

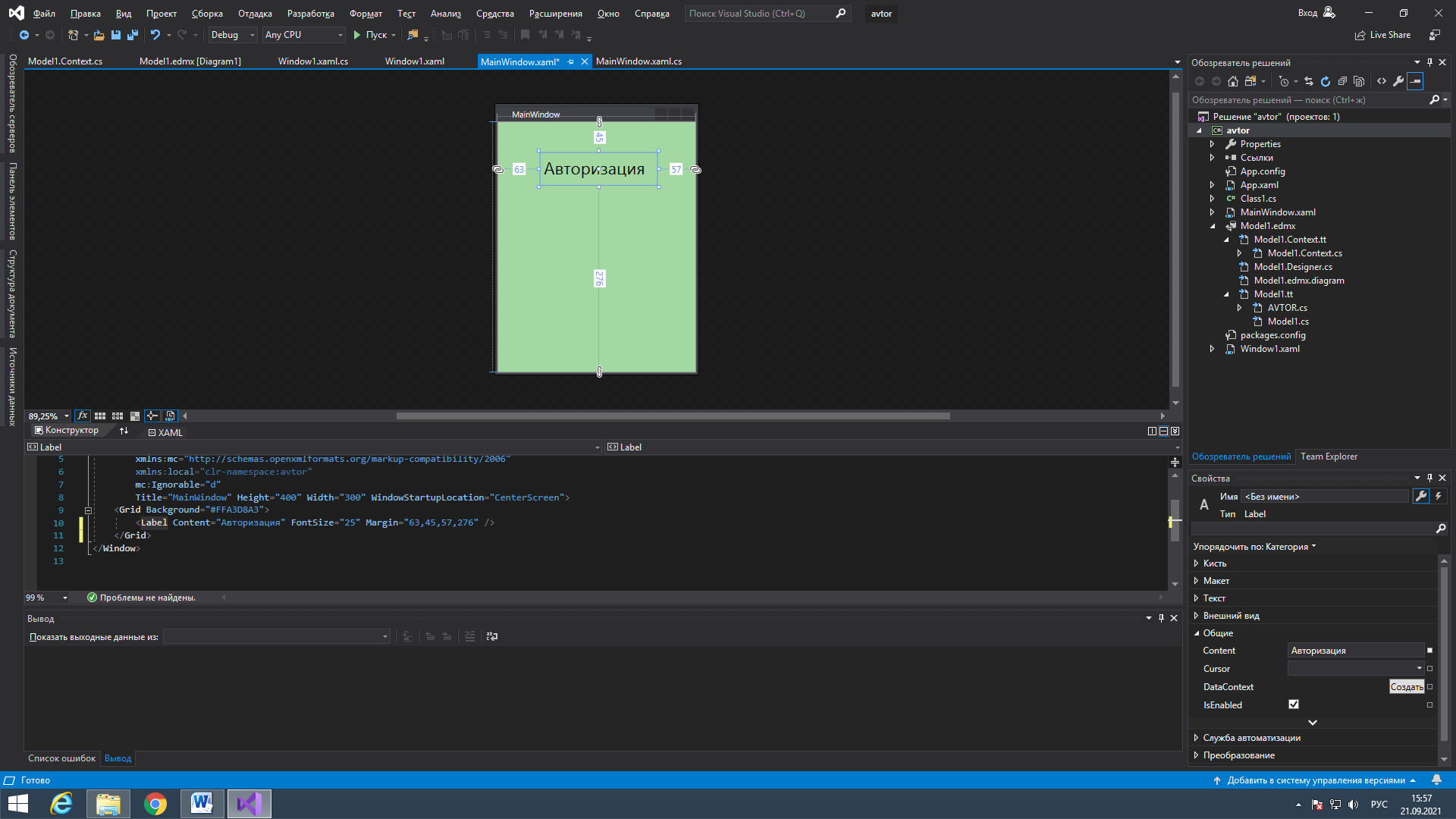


Рисунок 4 – Тэг “Label”

Добавляем 2 текстовых поля для ввода данных с помощью тэга “TextBox”, пример кода:

<TextBox x:Name="login" Text="Введите логин" FontStyle="Oblique" Background="#FFE2DBDB"/>. Тэг “X:Name”, задает имя текстовому полю, в последующем применяющийся для вызова поля. Тэг “FontStyle” необходим для изменения шрифта используемого текста. В тэг “Text” заносятся желаемые фразы, которые будут видны внутри текстового поля, как показано на рисунке 5.

*<TextBox x:Name="logi" PreviewMouseDown="Logi\_PreviewMouseDown" Text="Введите логин" FontStyle="Oblique" FontStretch="Normal" Background="#FFE2DBDB" Margin="83,122,89,215"/>*

*<TextBox x:Name="par" PreviewMouseDown="TextBox\_PreviewMouseDown" Text="Введите пароль" FontStyle="Oblique" FontStretch="Normal" Background="#FFE2DBDB" Margin="83,181,89,156"/>*

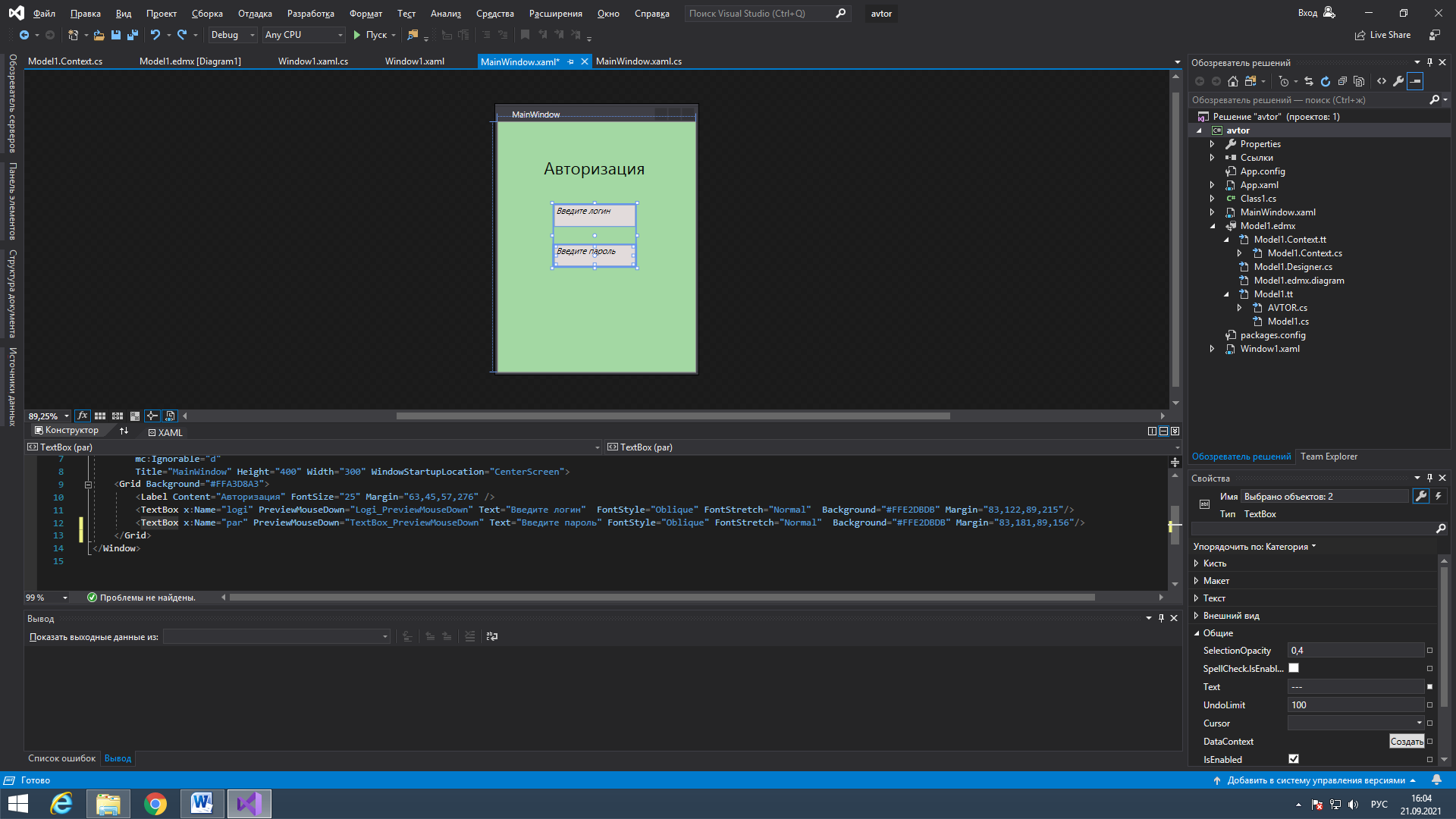


Рисунок 5 – Тэг “TextBox”

Для добавления на форму кнопки “Войти” требуется использовать тэг “Button”, пример кода: <Button Content="Войти" FontFamily="Sitka Text"/>, рисунок 6.

<Button Content="Войти" Margin="94,238,101,100" FontFamily="Sitka Text" Click="Button\_Click"/>

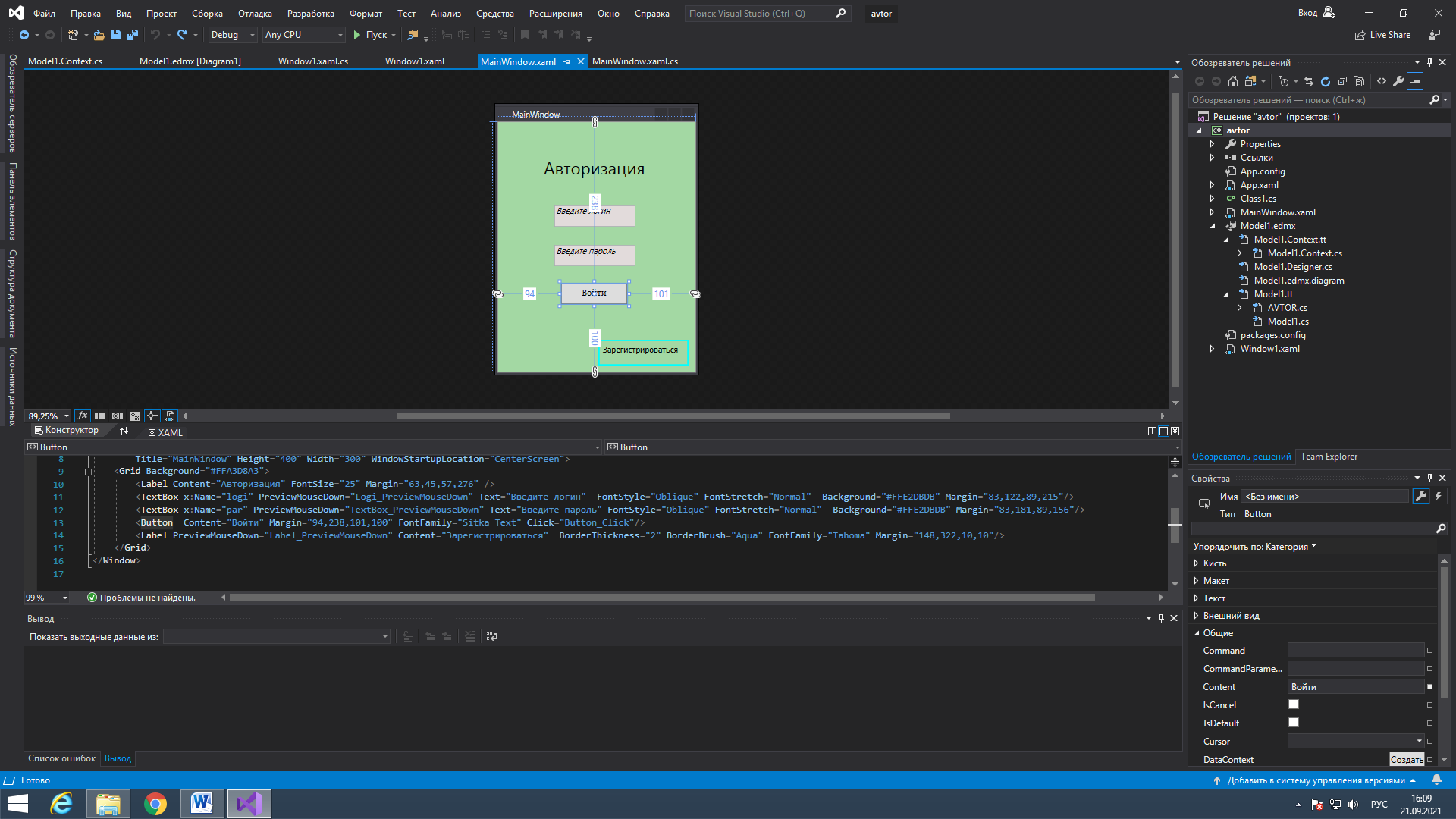


Рисунок 6 – Тэг “Button”

Ниже задаем текстовое поле с помощью тэга “Label”, которые был описан выше, с помощью метода “PreviewMouseDown” слово “Зарегистрироваться” становится кликабельным и при нажатие реализует переход на форму регистрации.

Готовая форма авторизации изображена на рисунке 7.

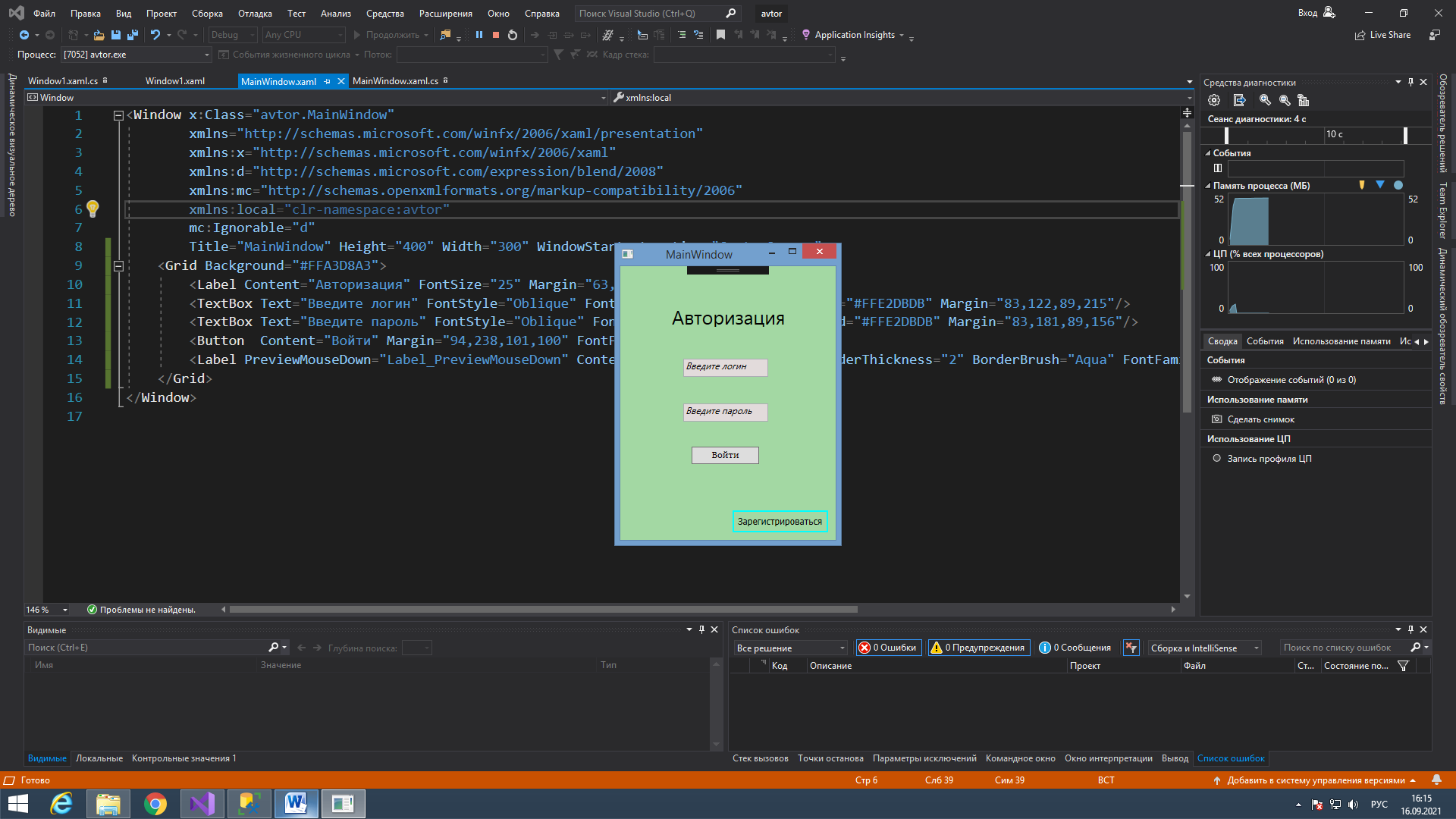


Рисунок 7 – Форма авторизации

Полный листинг для формы авторизации представлен ниже в Xaml:

Проверьте написания кода:

*<Grid Background="#FFA3D8A3">*

*<Label Content="Авторизация" FontSize="25" Margin="63,45,57,276" />*

*<TextBox x:Name="logi" PreviewMouseDown="Logi\_PreviewMouseDown" Text="Введите логин" FontStyle="Oblique" FontStretch="Normal" Background="#FFE2DBDB" Margin="83,122,89,215"/>*

*<TextBox x:Name="par" PreviewMouseDown="TextBox\_PreviewMouseDown" Text="Введите пароль" FontStyle="Oblique" FontStretch="Normal" Background="#FFE2DBDB" Margin="83,181,89,156"/>*

*<Button Content="Войти" Margin="94,238,101,100" FontFamily="Sitka Text" Click="Button\_Click"/>*

*<Label PreviewMouseDown="Label\_PreviewMouseDown" Content="Зарегистрироваться" BorderThickness="2" BorderBrush="Aqua" FontFamily="Tahoma" Margin="148,322,10,10"/>*

*</Grid>*

3. Создайте класс - через добавление элементов – класс (правая клавиша мыши по названию проекта в окне обозревателя – добавить –создать элемент - класс, внутри класса пропишите следующий код:

namespace avtor

{

class Class1

{

public static AutorEntities context = new AutorEntities();

}

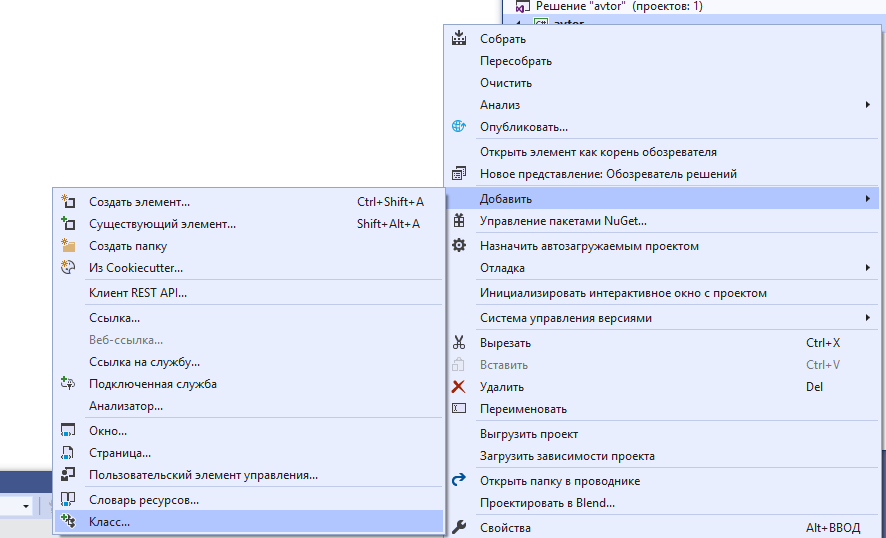
}

***AutorEntities название модели подключения базы данных, изменить при подключении.***

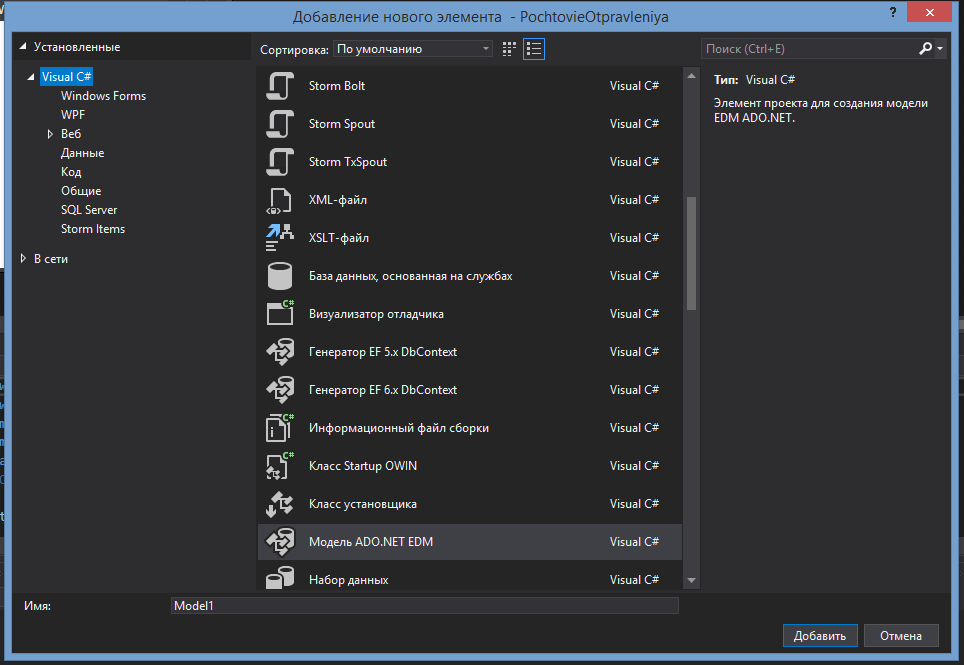
4. Настройте подключение к базе данных созданной с помощью скрипта в MS SQL либо с нуля.

Для подключения базы данных необходимо добавить модель ADO.NET.

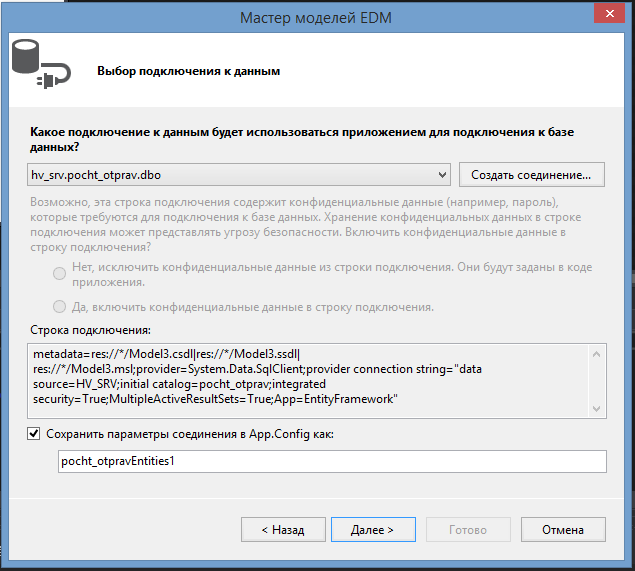
- правая клавиша мыши по названию проекта в окне обозревателя выбрать добавить-элемент- класс



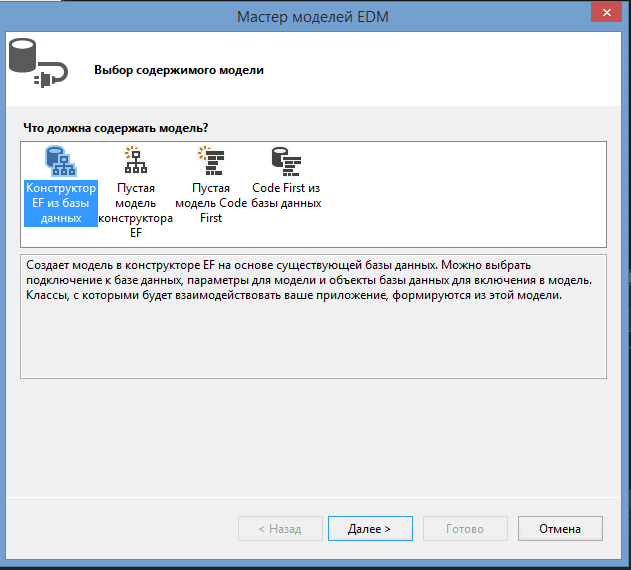
-выбрать модель ADO.NET.



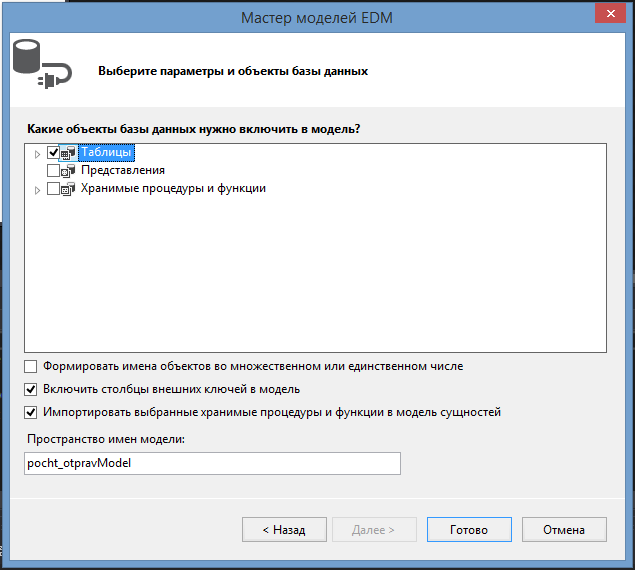
- выбрать сервер, установить подлинность если это необходимо для подключения, логин, пароль, затем выбрать имя базы данных;



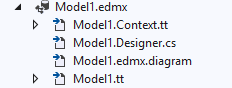
Далее:



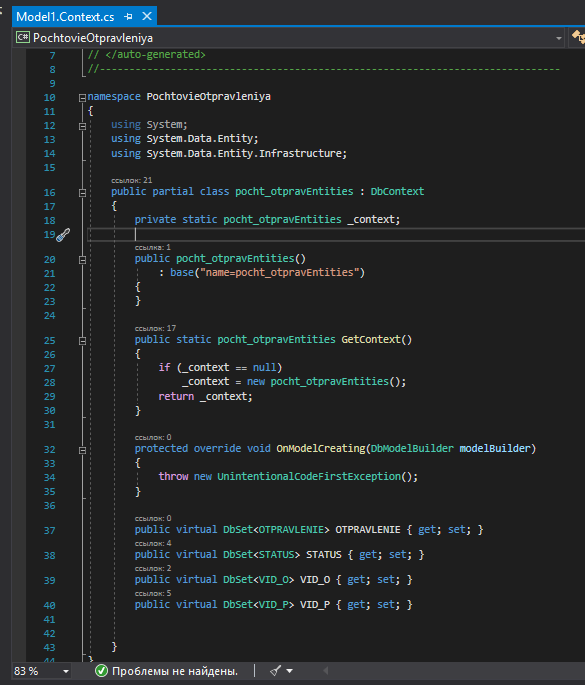
Далее:



В обозревателе решений при успешном подключении должна появиться модель подключения базы данных:



- разверните модель .Context.tt, далее в Model1.Context.cs, добавьте недостающий код:



***Конкретно следующие строки:***

private static pocht\_otpravEntities \_context;

public static pocht\_otpravEntities GetContext()

{

if (\_context == null)

\_context = new pocht\_otpravEntities();

return \_context;

}

*Примечание: данный код позволяет обращаться к вашей базе данных.*

Листинг кода CS для окна авторизации, вставляем в кнопку ОК

namespace avtor

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Label\_PreviewMouseDown(object sender, MouseButtonEventArgs e)

{

Window1 window1 = new Window1();

window1.Show();

this.Close();

}

private void TextBox\_PreviewMouseDown(object sender, MouseButtonEventArgs e)

{

par.Clear();

}

private void Logi\_PreviewMouseDown(object sender, MouseButtonEventArgs e)

{

logi.Clear();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

var dir = from s in Class1.context.AVTOR

where s.LOGIN == logi.Text && s.PASSWORD == par.Text

select new

{

logi = s.LOGIN,

par = s.PASSWORD

};

if (dir.Count() !=0)

{

MessageBox.Show("Добро пожаловать!");

}

else

{

MessageBox.Show("Данные введены неправильно");

}

}

}

}

При нажатии на кнопку «Зарегистрироваться» должно появится окно регистрации, но для этого его необходимо добавить:

Для того чтобы добавить окно необходимо добавить элемент «Окно», и в кнопке «зарегистроваться» прописать следующий код перехода.

***Примечание: данный код уже добавлен из предыдущего листинга***

private void Label\_PreviewMouseDown(object sender, MouseButtonEventArgs e)

{

Window1 window1 = new Window1();

window1.Show();

this.Close();

}

Готовая форма регистрации изображена на рисунке 8.

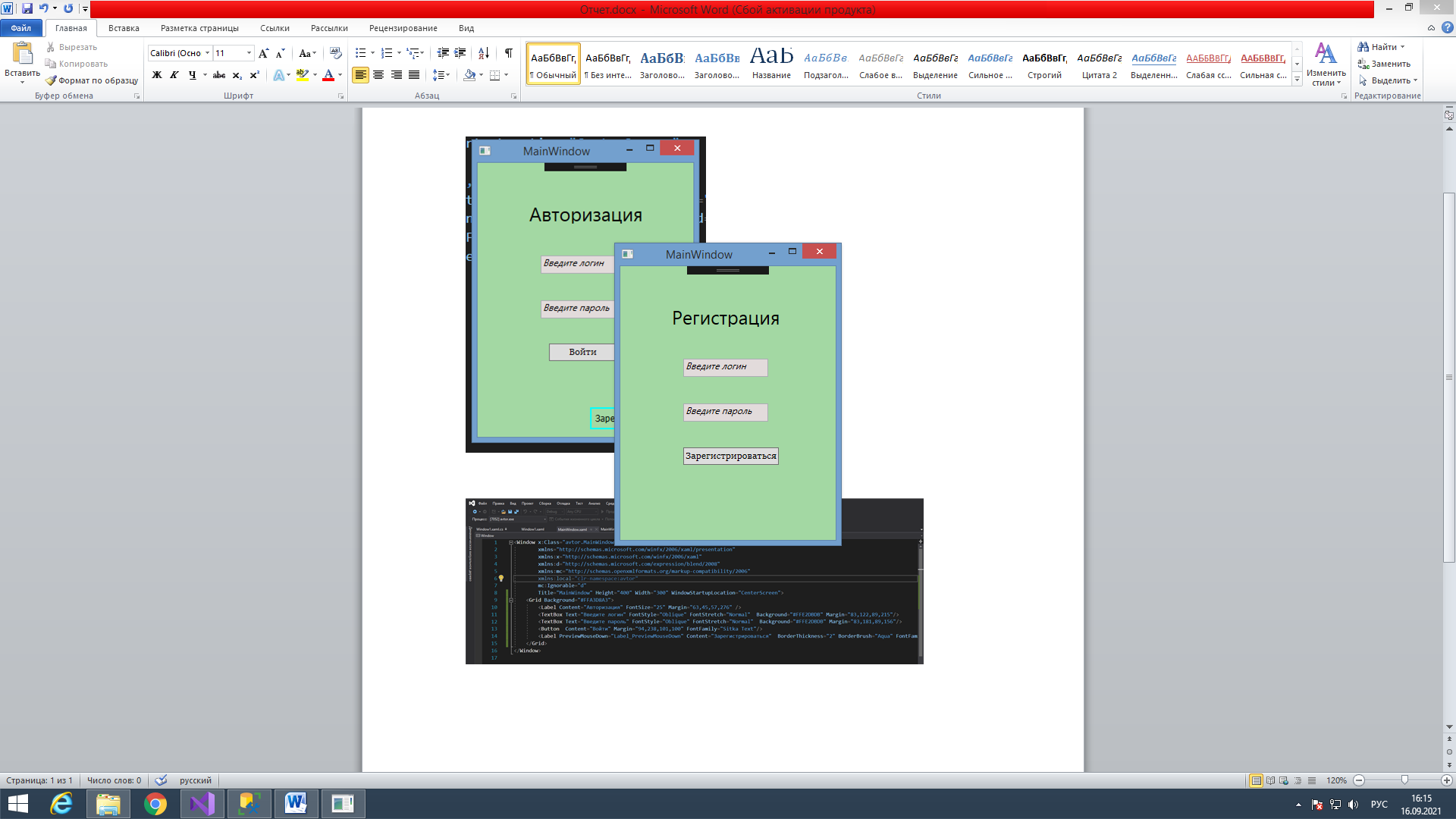


Рисунок 8 – Форма регистрации

Листинг для формы регистрации представлен ниже:

<Grid Background="#FFA3D8A3">

<Label Content="Регистрация" FontSize="25" Margin="63,45,57,276" />

<TextBox PreviewMouseDown="Log\_PreviewMouseDown" x:Name="log" Text="Введите логин" FontStyle="Oblique" FontStretch="Normal" Background="#FFE2DBDB" Margin="83,122,89,215"/>

<TextBox PreviewMouseDown="Pas\_PreviewMouseDown" x:Name="pas" Text="Введите пароль" FontStyle="Oblique" FontStretch="Normal" Background="#FFE2DBDB" Margin="83,181,89,156"/>

<Button Content="Зарегистрироваться" Margin="83,239,75,99" FontFamily="Sitka Text" Click="Button\_Click"/>

</Grid>

Листинг кода для выполнения регистрации представлен ниже:

namespace avtor

{

/// <summary>

/// Логика взаимодействия для Window1.xaml

/// </summary>

public partial class Window1 : Window

{

public Window1()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

AVTOR con = new AVTOR()

{

LOGIN = log.Text,

PASSWORD = pas.Text

};

try

{

Class1.context.AVTOR.Add(con);

Class1.context.SaveChanges();

MessageBox.Show("Вы успешно зарегистрировались");

MainWindow ax = new MainWindow();

ax.Show();

this.Close();

}

catch

{

}

}

private void Log\_PreviewMouseDown(object sender, MouseButtonEventArgs e)

{

log.Clear();

}

private void Pas\_PreviewMouseDown(object sender, MouseButtonEventArgs e)

{

pas.Clear();

}

}

}

После добавления кода для реализации регистрации, запускаем приложение и жмем на поле “Зарегистрироваться”, далее вводим желаемый логин и пароль, регистрация прошла успешно, как показано на рисунке 10

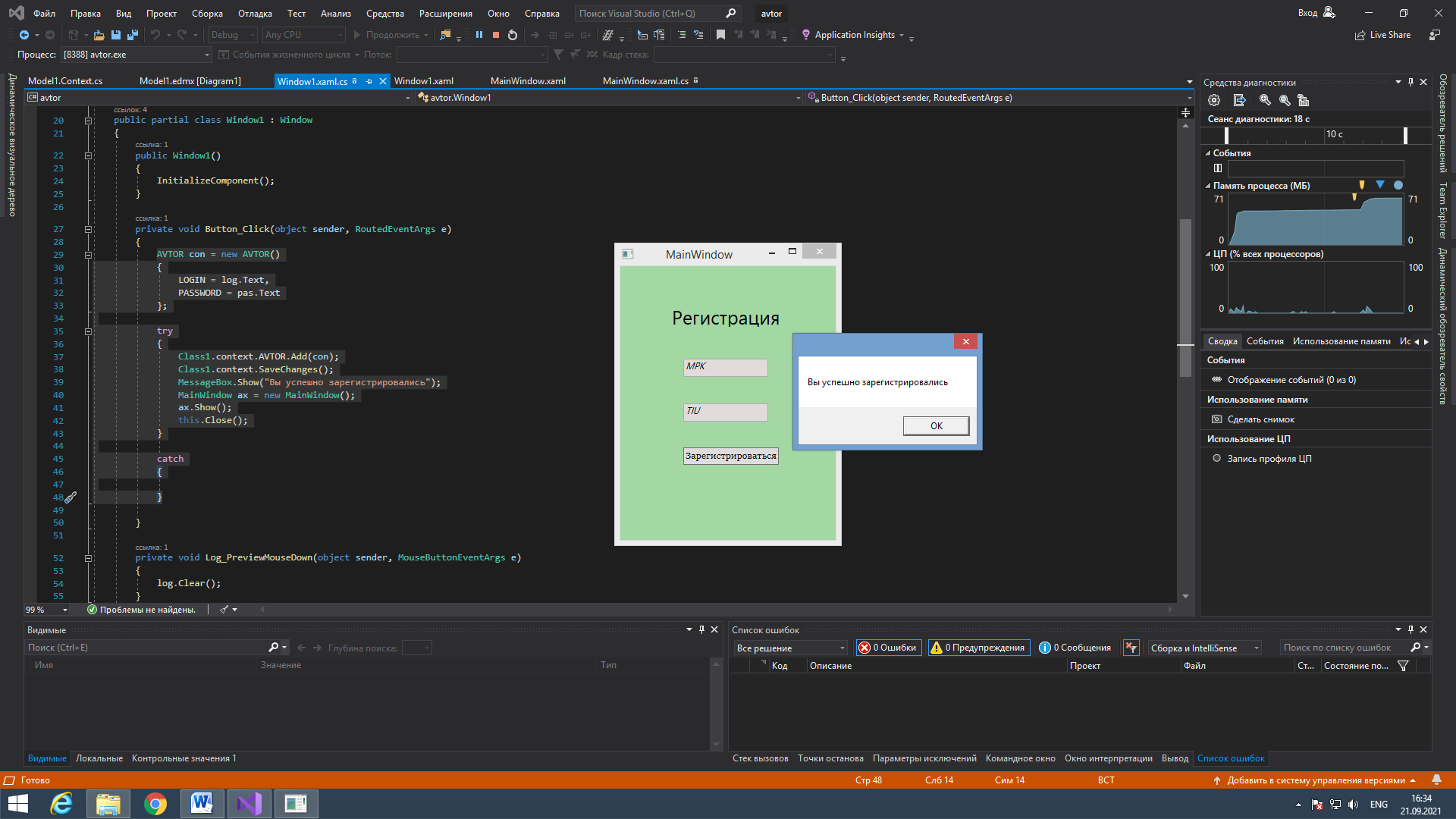


Рисунок 10 – Успешная регистрация

Данные, которые ввели при регистрации занеслись в базу данных, как представлено на рисунке 11.

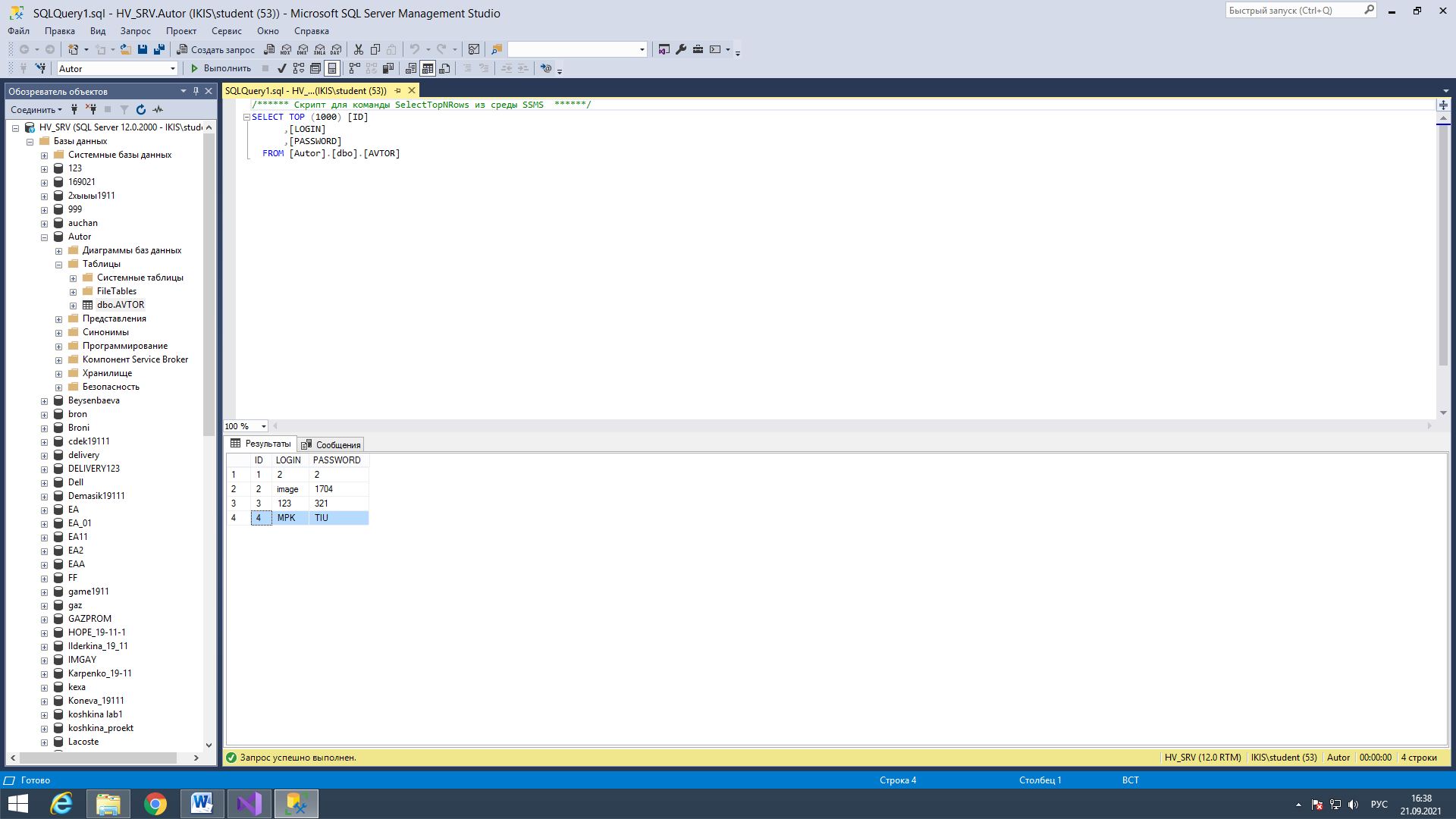


Рисунок 11 – Отображение данных в базе данных

Успешная авторизация изображена на рисунке 12.

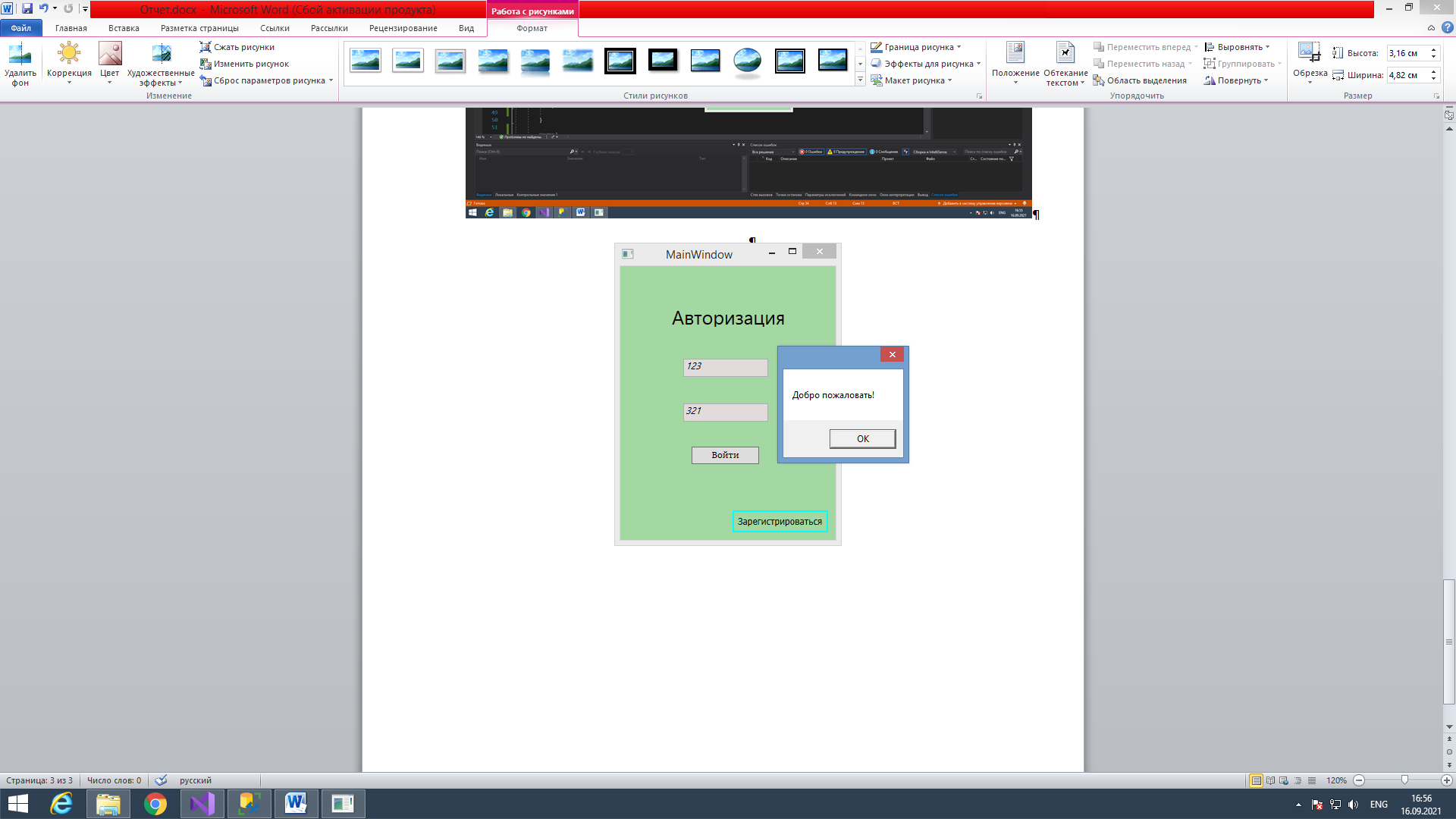


Рисунок 12 – Успешная авторизация

Ввод неправильных данных изображен на рисунке 13.

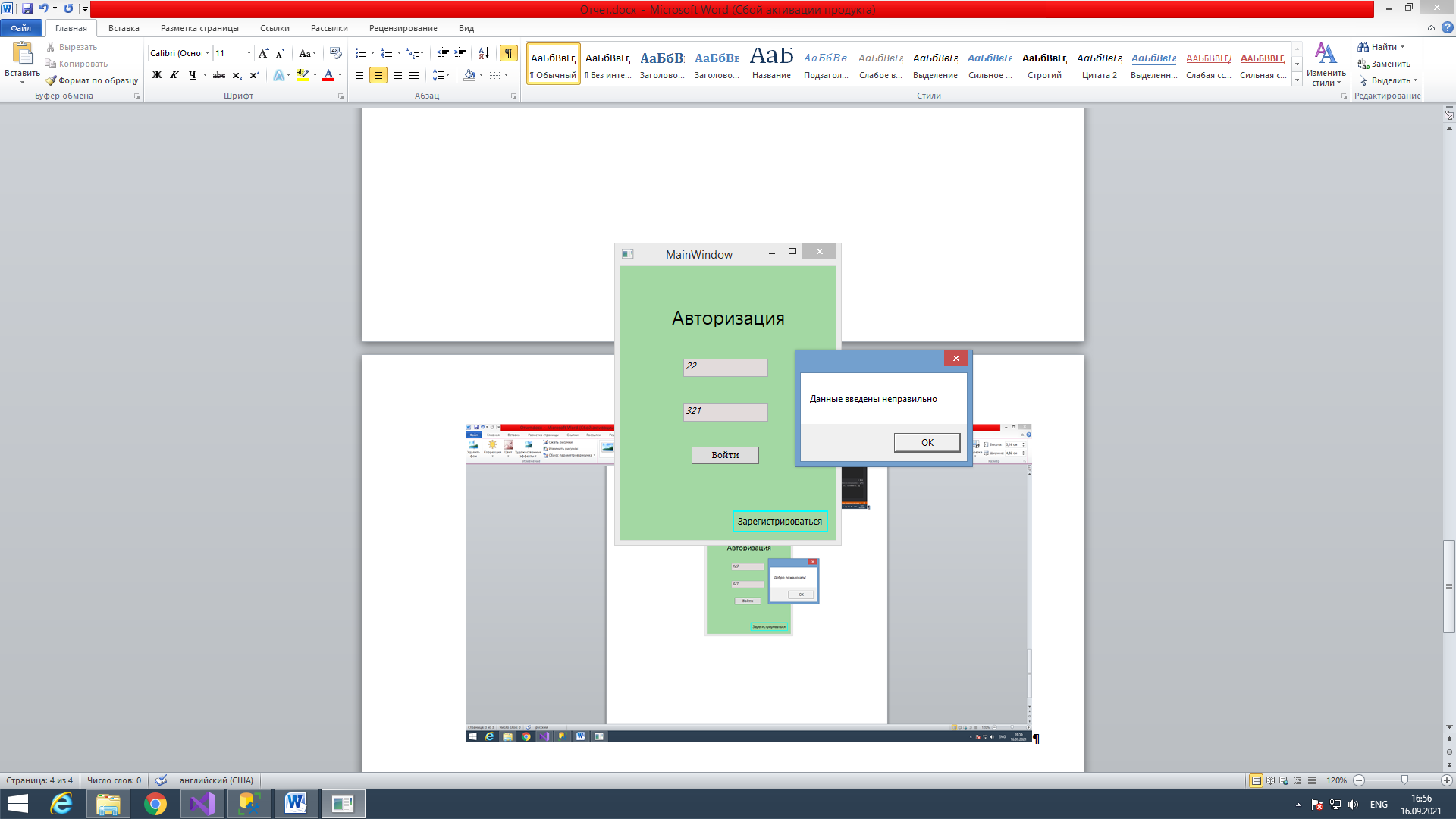
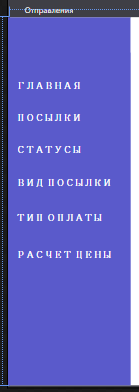


Рисунок 13 – Неправильные данные

**Реализация добавления, удаления и редактирования**

Для создания меню необходимо в XAML создать StackPanel с ListView, где будут храниться элементы ListViewItem с названием страницы на которую нужно переключится:



<StackPanel HorizontalAlignment="Left" Height="552" VerticalAlignment="Top" Width="186" Background="#FF5A5ACB" Margin="-5,0,0,0">

<Grid Height="80"></Grid>

<ListView x:Name="menu" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFC97E7E" Height="310">

<ListViewItem x:Name="Glavnaya" Content=" ГЛАВНАЯ" Height="45" BorderBrush="{x:Null}" Background="#FF5A5ACB" Foreground="White" FontFamily="Adobe Ming Std L" FontSize="14"/>

<ListViewItem x:Name="Posilki" Content=" ПОСЫЛКИ" Height="49" BorderBrush="{x:Null}" FontFamily="Adobe Ming Std L" Foreground="White" Background="#FF5A5ACB" FontSize="14"/>

<ListViewItem x:Name="Posilki\_Copy" Content=" СТАТУСЫ" Height="49" BorderBrush="{x:Null}" FontFamily="Adobe Ming Std L" Foreground="White" Background="#FF5A5ACB" FontSize="14" Selected="Posilki\_Copy\_Selected\_1"/>

<ListViewItem x:Name="Posilki\_Copy1" Content=" ВИД ПОСЫЛКИ" Height="49" BorderBrush="{x:Null}" FontFamily="Adobe Ming Std L" Foreground="White" Background="#FF5A5ACB" FontSize="14"/>

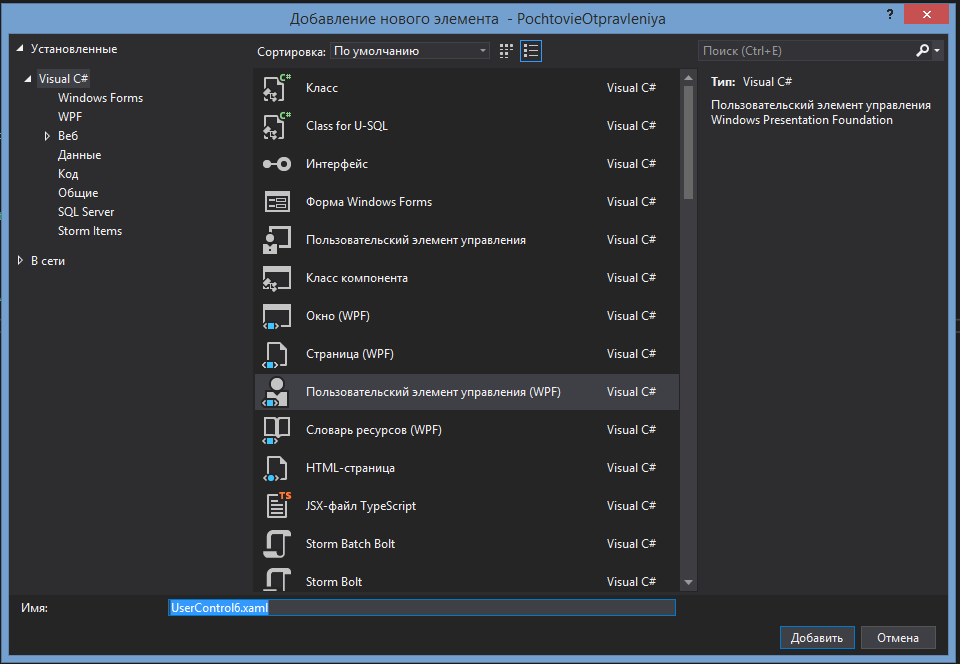
<ListViewItem x:Name="Price" Content=" ТИП ОПЛАТЫ" Height="55" BorderBrush="{x:Null}" FontFamily="Adobe Ming Std L" FontSize="14" Background="#FF5A5ACB" Foreground="White"/>

<ListViewItem x:Name="Price\_Copy" Content=" РАСЧЕТ ЦЕНЫ" Height="55" BorderBrush="{x:Null}" FontFamily="Adobe Ming Std L" FontSize="14" Background="#FF5A5ACB" Foreground="White"/>

</ListView>

</StackPanel>

После чего добавляем пользовательский элемент управления, который будет служить страницей, на которую нужно переключиться



В ранее созданный ListView добавляем событие SelectionChanged:

<ListView x:Name="menu" Background="{x:Null}" BorderBrush="{x:Null}" SelectionChanged="ListView\_SelectionChanged" Foreground="#FFC97E7E" Height="310">

И пишем подобный код

private void ListView\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

int index = menu.SelectedIndex;

switch (index)

{

case 0:

GridMain.Children.Clear();

GridMain.Children.Add(new UserControl3());

break;

case 1:

GridMain.Children.Clear();

GridMain.Children.Add(new UserControl1());

break;

case 2:

GridMain.Children.Clear();

GridMain.Children.Add(new UserControl5());

break;

case 3:

GridMain.Children.Clear();

GridMain.Children.Add(new UserControl8());

break;

case 4:

GridMain.Children.Clear();

GridMain.Children.Add(new UserControl4());

break;

case 5:

GridMain.Children.Clear();

GridMain.Children.Add(new UserControl2());

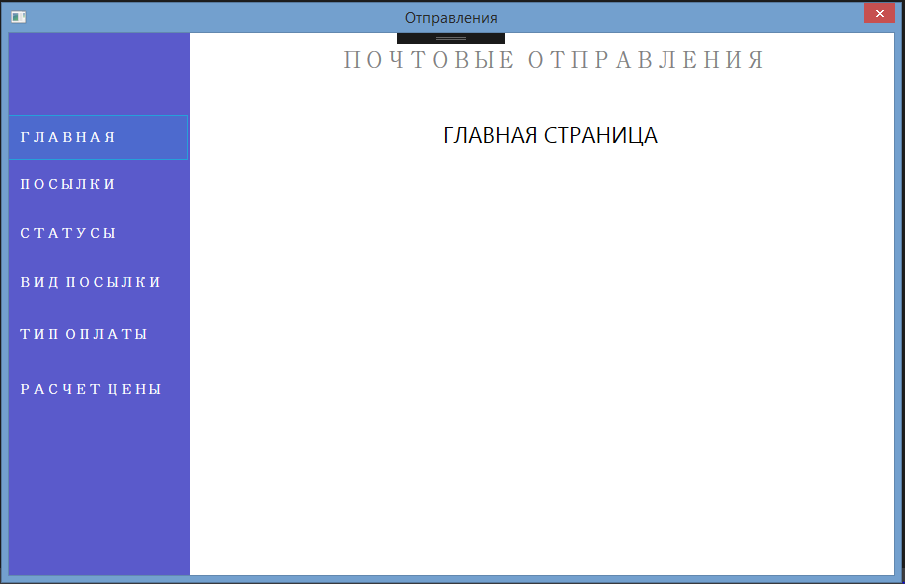
break;

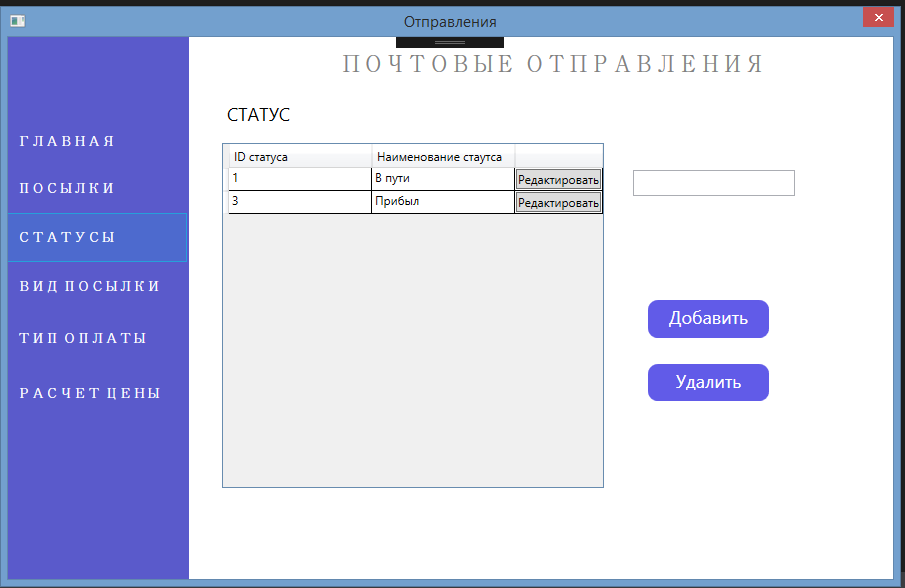
}

}

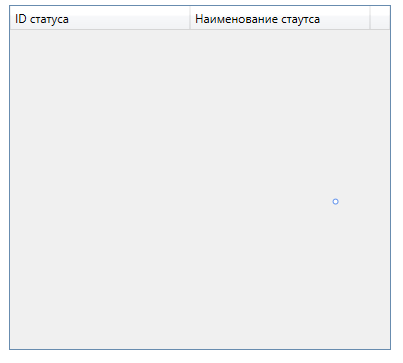
Где в GridMain.Children.Add(new UserControl4()) указываем имя нужного нам UserControl.

Теперь пользователь может переключаться между вкладками.





Для отображения данных из базы данных вставляем DataGrid



И пишем следующий код:

<DataGrid x:Name="Status" HorizontalAlignment="Left" Height="345" Margin="28,53,0,0" VerticalAlignment="Top" Width="382" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="ID статуса" Width="\*" Binding="{Binding S\_ID}" ></DataGridTextColumn>

<DataGridTextColumn Header="Наименование стаутса" Width="\*" Binding="{Binding S\_NAIMENOVANIE}"></DataGridTextColumn>

<DataGridTemplateColumn>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать"></Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

Где указываем имя таблицы (x:Name="Status"), название столбцов (Header="ID статуса") и привязываем их к БД (Binding="{Binding S\_ID}"), написав название оттуда.

И название кнопки «Редактировать» (Content="Редактировать"). Она понадобиться позже.

Для отображения данных в xaml.cs нужно написать:

public UserControl5()

{

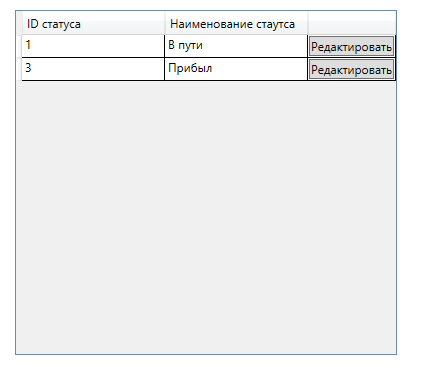
InitializeComponent();

Status.ItemsSource = pocht\_otpravEntities.GetContext().STATUS.ToList();

}

Где Status-имя DataGrid, pocht\_otpravEntities –модель которая была создана, STATUS – имя таблицы в БД.

Данные буду отображены.



Для добавления данных создаем кнопку "Добавить".

<Button Content=" Добавить" Margin="9,3,9,-1" VerticalAlignment="Top" Height="36" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="White" FontSize="18" Click="Button\_Click">

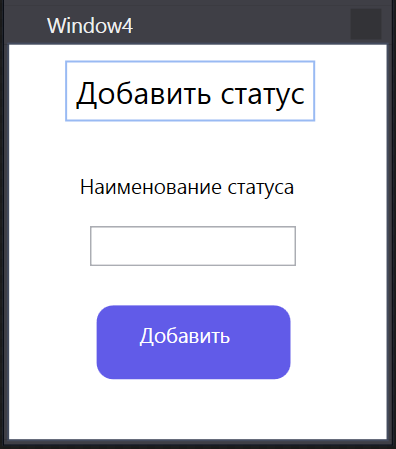


В Button\_Click пишется следующий код:

Window4 WIN4 = new Window4(null);

WIN4.Show();

После необходимо создать новое окно для добавления записи.



Со следующим кодом:

<Grid>

<Label Content="Добавить статус" HorizontalAlignment="Left" Margin="34,10,0,0" VerticalAlignment="Top" FontSize="18" RenderTransformOrigin="0.642,0.886"/>

<TextBox x:Name="Stat" HorizontalAlignment="Left" Height="23" Margin="47,106,0,0" TextWrapping="Wrap" Text="{Binding S\_NAIMENOVANIE}" VerticalAlignment="Top" Width="120"/>

<Border CornerRadius="10" BorderThickness="1" Margin="50,151,55,34" Background="#FF615BE8" RenderTransformOrigin="0.5,0.5">

<Button Background="{x:Null}" Foreground="White" Content="Добавить" HorizontalAlignment="Left" Margin="9,-1,0,0" VerticalAlignment="Top" Width="86" Click="Button\_Click" Height="37" BorderBrush="{x:Null}"/>

</Border>

<Label Content="Наименование статуса" HorizontalAlignment="Left" Margin="36,69,0,0" VerticalAlignment="Top"/>

</Grid>

Где в Text="{Binding S\_NAIMENOVANIE}" прописывается название столбца, в который нужно добавить данные.

В Click прописывается следующий код

private STATUS \_status = new STATUS();

public Window4(STATUS selectedStatus)

{

InitializeComponent();

if (selectedStatus != null)

\_status = selectedStatus;

DataContext = \_status;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (Stat.Text == null)

errors.AppendLine("Укажите статус");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;

}

if (\_status.S\_ID == 0)

pocht\_otpravEntities.GetContext().STATUS.Add(\_status);

try

{

pocht\_otpravEntities.GetContext().SaveChanges();

MessageBox.Show("Сохранено");

}

catch (Exception ex)

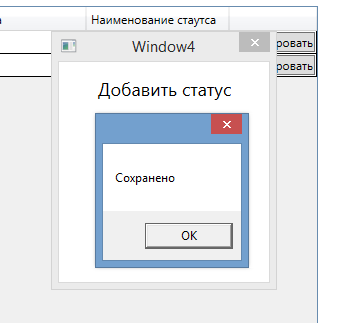
{

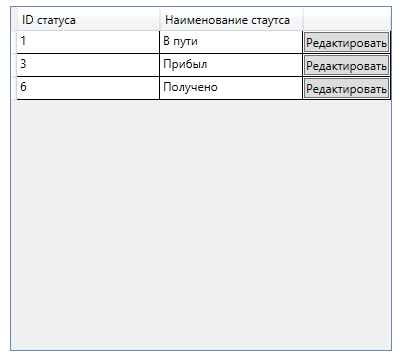
MessageBox.Show(ex.Message.ToString());

}

}

Запись будет добавлена.





Для кнопки редактирования нужно написать следующий код:

private void ButtonEdite\_Click(object sender, RoutedEventArgs e)

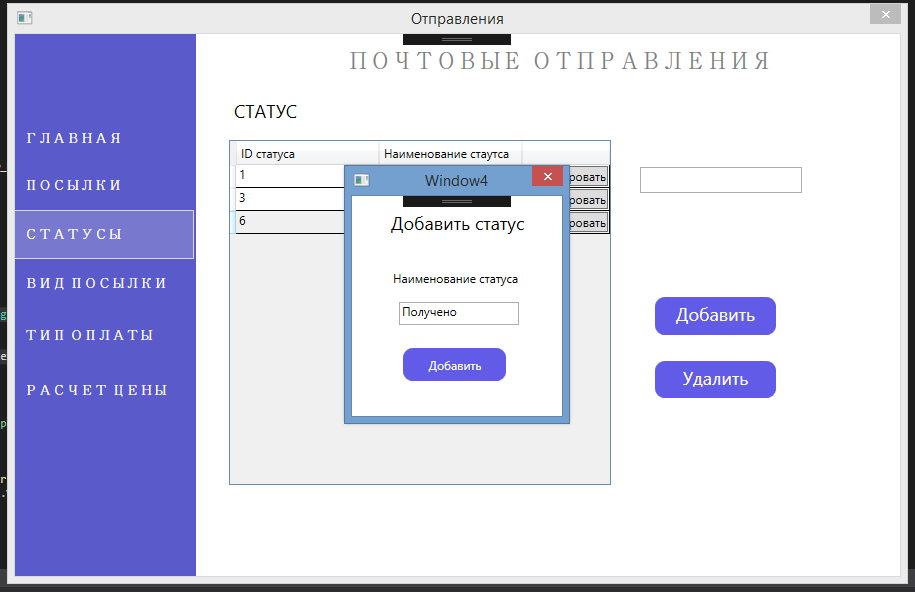
{

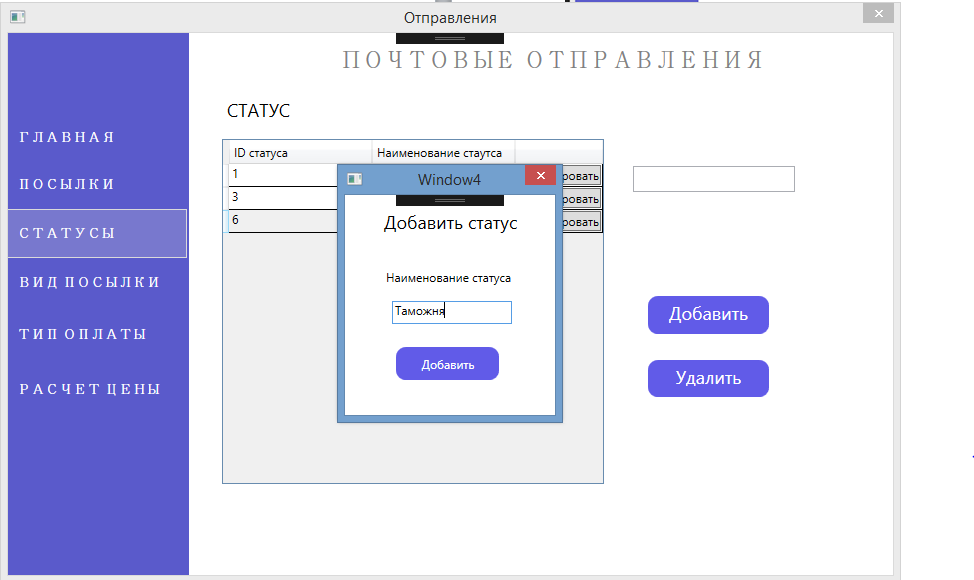
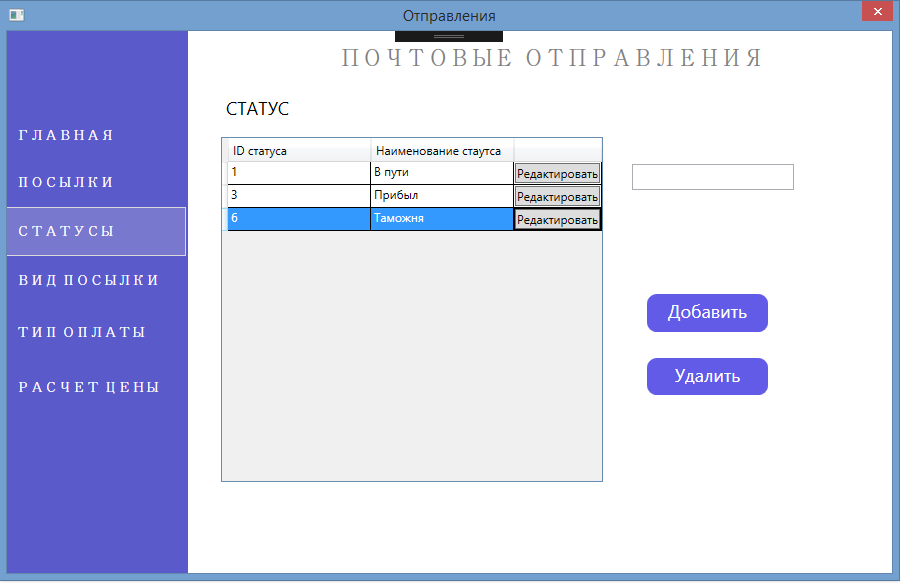
Window2 win2 = (new Window2((sender as Button).DataContext as VID\_P));

win2.Show();

}

После редактирование будет работать.



Для удаления в кнопке Удалить в событии Click пишет следующий код

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

var foremove = Status.SelectedItems.Cast<STATUS>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {foremove.Count()} элементов?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

pocht\_otpravEntities.GetContext().STATUS.RemoveRange(foremove);

pocht\_otpravEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

Status.ItemsSource = pocht\_otpravEntities.GetContext().STATUS.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

Где STATUS – название таблицы в БД.

Поиск реализован с помощью тега “TextBox” и кодом в С#:

private void TextBox\_TextChanged(object sender, TextChangedEventArgs e)

{

var pop = pocht\_otpravEntities.GetContext().STATUS.ToList();

pop = pop.Where(d => d.S\_NAIMENOVANIE.Contains(Poisk.Text)).ToList();

Status.ItemsSource = pop;

}

