

Настольное приложение «Салон красоты»

Создание приложения

Настроить новый проект

Приложение WPF (.NET Framework)

C#

XAML

Windows

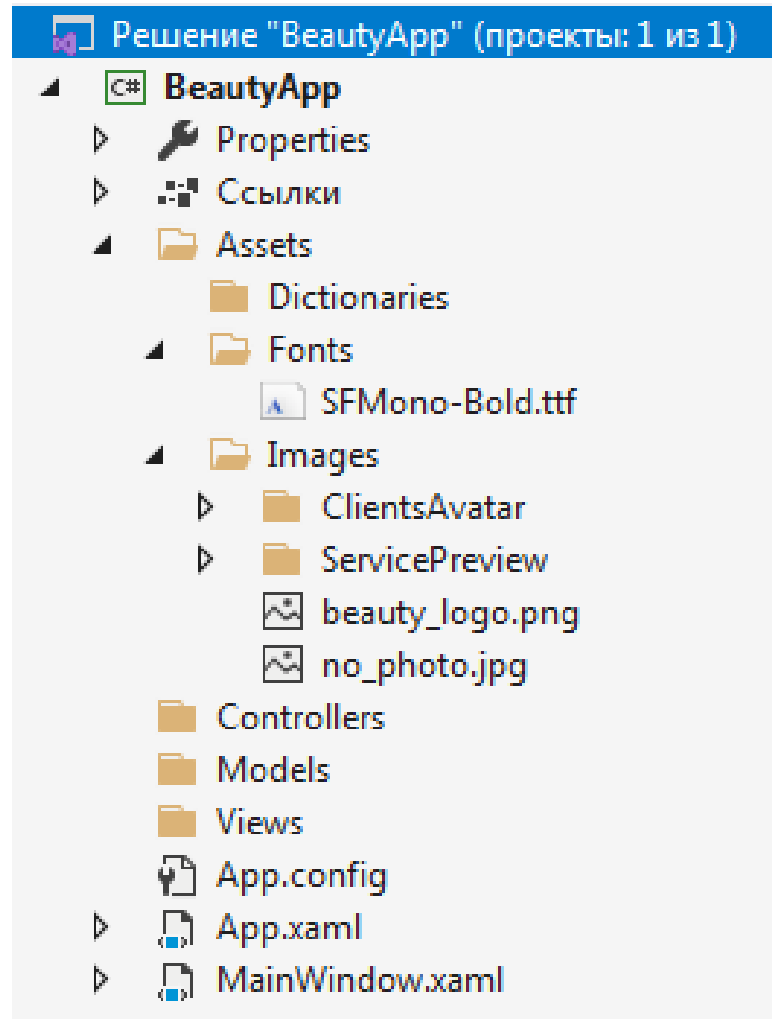
Рабочий стол

Имя проекта

BeautyApp

Расположение

Структура проекта



Оформление главного окна

```
<Window
```

```
...
```

```
    Icon="Assets/Images/beauty_logo.png"
```

```
    Title="Салон красоты"
```

```
    Height="450"
```

```
    Width="800"
```

```
    MaxHeight="900"
```

```
    MaxWidth="1920"
```

```
    MinHeight="450"
```

```
    MinWidth="800"
```

```
    WindowStartupLocation="CenterScreen"
```

```
    WindowStyle="SingleBorderWindow">
```

```
</Window>
```

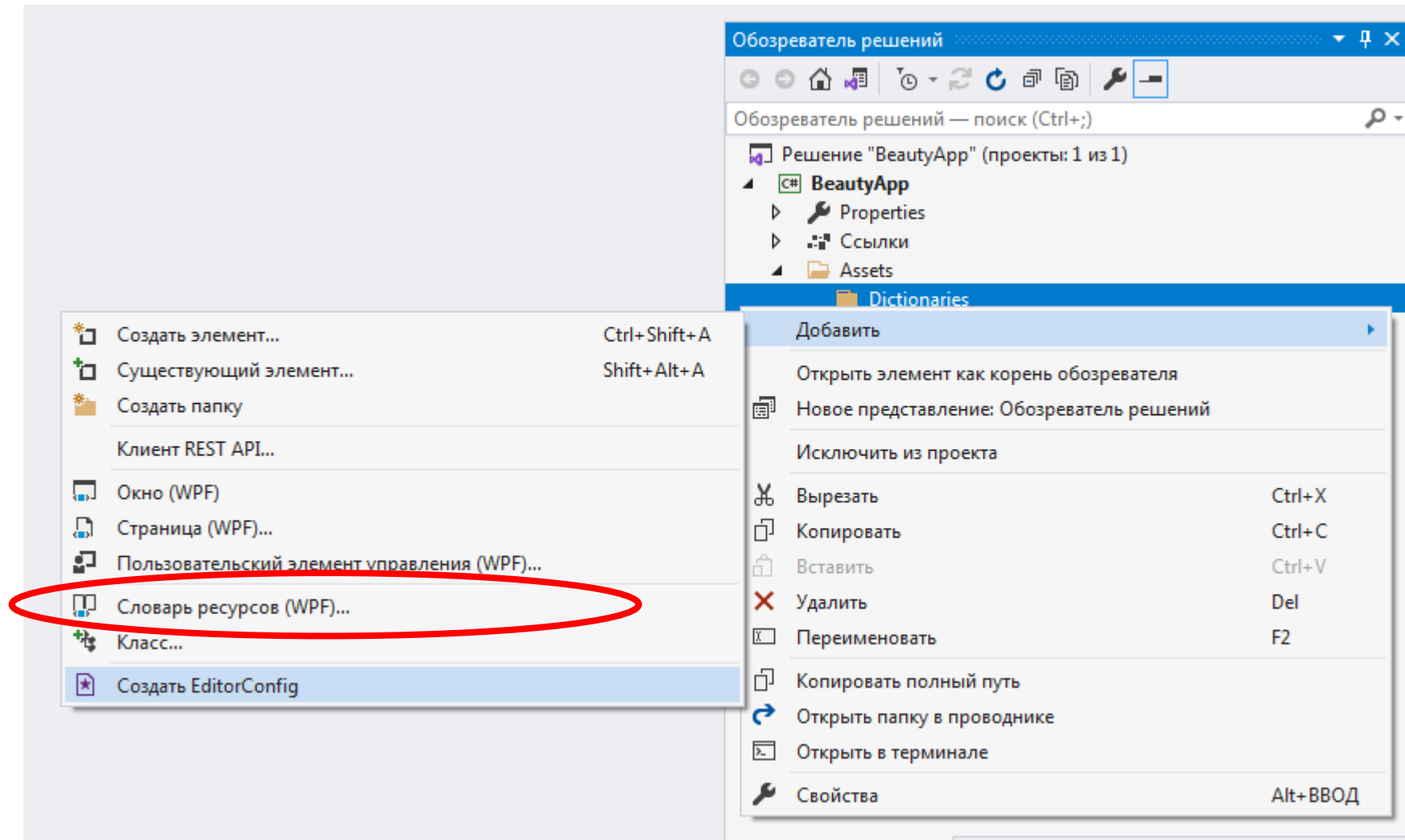
Верстка каркаса главного окна

```
<Window >
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition Height="Auto"/>
      <RowDefinition Height="Auto"/>
      <RowDefinition/>
      <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>
    <WebBrowser x:Name="MainWebFrame" Grid.RowSpan="2" Visibility="Collapsed"/>
    <StackPanel Grid.Row="2" Orientation="Horizontal">
      <Ellipse Fill="Red" />
      <TextBlock x:Name="StatusTextBlock"/>
    </StackPanel>
  </Grid>
```

Событие вызова справки

```
KeyDown += (s, e) => {  
    if (e.Key == Key.F1)  
    {  
        string filename = $"{AppDomain.CurrentDomain.BaseDirectory}help.pdf";  
        MainWebFrame.Visibility=Visibility.Visible;  
        MainWebFrame.Navigate(filename);  
        this.Title = "Руководство пользователя";  
        StatusTextBlock.Text = "Для выхода из справки нажмите ESC";  
    }  
    if (e.Key == Key.Escape)  
    {  
        MainWebFrame.Visibility = Visibility.Collapsed;  
        this.Title = "Салон красоты";  
        StatusTextBlock.Text = String.Empty;  
    }  
};
```

Создание стилевого файла



Подключение стилевого файла в App.xaml

```
<Application ...>  
  <Application.Resources>  
    <ResourceDictionary>  
      <ResourceDictionary.MergedDictionaries>  
        <ResourceDictionary Source="/Assets/DictionaryMain.xaml"/>  
      </ResourceDictionary.MergedDictionaries>  
    </ResourceDictionary>  
  </Application.Resources>  
</Application>
```


Стилизация (файл DictionaryMain.xaml)

```
<FontFamily x:Key="CaptionFont">  
    pack://application:,,,/Assets/Fonts/SFMono-Bold.ttf#SF Mono  
</FontFamily>
```

Задаем шрифт для заголовков!

Стилизация (файл DictionaryMain.xaml)

```
<SolidColorBrush Color="#F29B99" x:Key="MainColor"/>  
<SolidColorBrush Color="#D9D2D0" x:Key="SecondColor"/>  
<SolidColorBrush Color="Azure" x:Key="ActiveColor"/>  
<SolidColorBrush Color="DarkSlateGray" x:Key="TextColor"/>
```

Задаем основные цвета проекта!

Стилизация (файл DictionaryMain.xaml)

```
<Style TargetType="Grid" >
    <Setter Property="Background" Value="{StaticResource MainColor}"/>
</Style>
<Style TargetType="TextBlock" >
    <Setter Property="HorizontalAlignment" Value="Center"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="Margin" Value="0 0 5 0"/>
</Style>
<Style TargetType="TextBlock" x:Key="TitleTextBlock">
    <Setter Property="HorizontalAlignment" Value="Center"/>
    <Setter Property="VerticalAlignment" Value="Center"/>
    <Setter Property="Foreground" Value="{StaticResource TextColor}"/>
    <Setter Property="FontSize" Value="20"/>
    <Setter Property="FontFamily" Value="{StaticResource CaptionFont}"/>
</Style>
```



Стилизуем текст!

Создание шапки главного окна

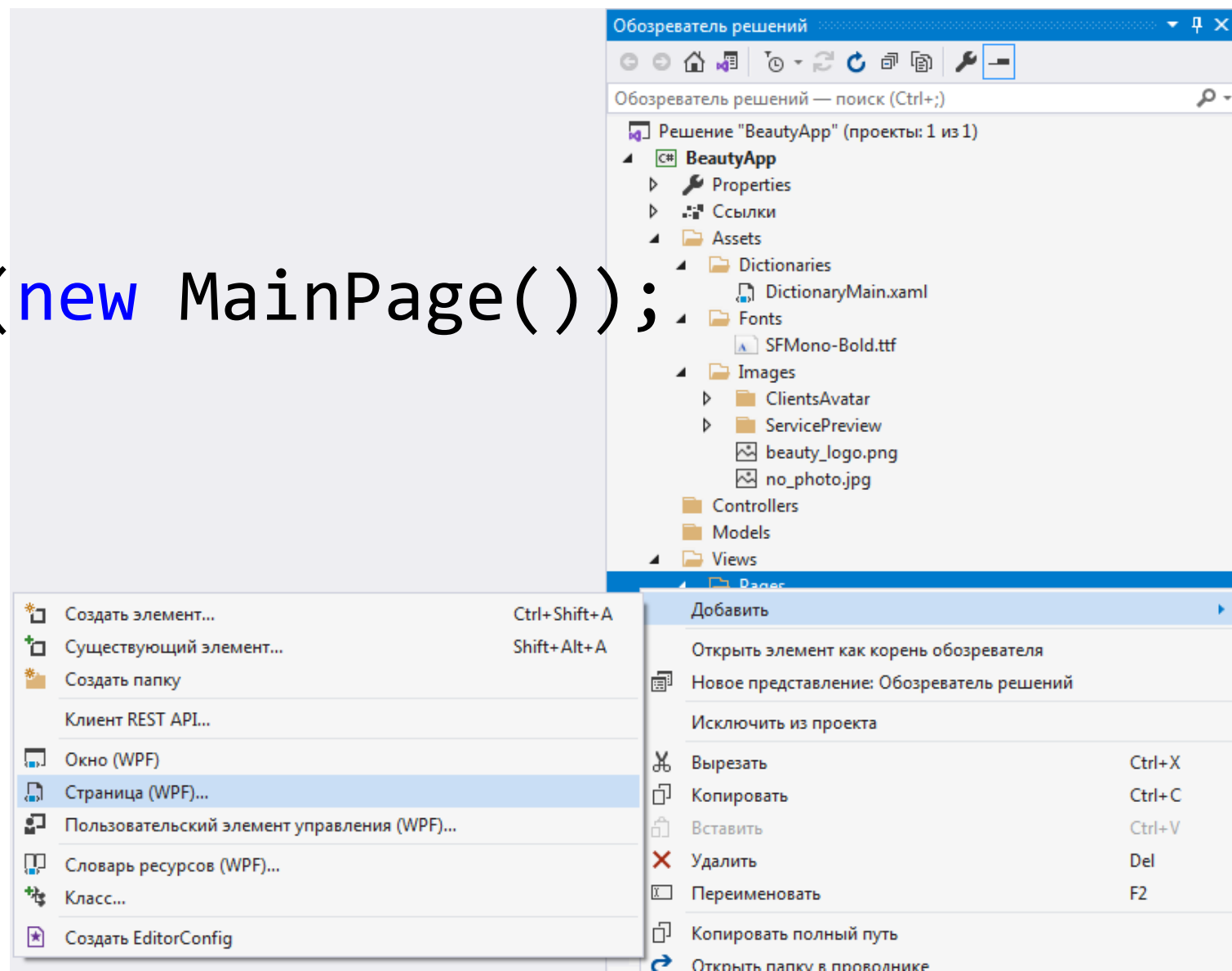
```
<StackPanel HorizontalAlignment="Right" Orientation="Horizontal">
    <TextBlock x:Name="PersonTextBlock" Text="Вы не вошли в систему"/>
    <TextBlock Text="Вход" x:Name="LogInTextBlock"
        MouseLeftButtonDown="LogInTextBlockMouseLeftButtonDown"/>
</StackPanel>

<TextBlock
    Grid.Row="1" Text="{Binding ElementName=MainFrame,Path=Content.Title}"
    Style="{StaticResource TitleTextBlock}" />

<Frame
    x:Name="MainFrame"
    Grid.Row="2"
    Navigated="MainFrameNavigated"
    NavigationUIVisibility="Hidden"/>
```

Загружаем во фрейм страницу MainPage при запуске приложения

```
MainFrame.Navigate(new MainPage());
```



Программируем «Вход»

```
private void LogInTextBlockMouseDown(object sender,
                                     MouseButtonEventArgs e)
{
    MainFrame.Navigate(new AuthPage());
    LogInTextBlock.Visibility = Visibility.Collapsed;
}
```

При нажатии на надпись «ВХОД»
Открывается страница авторизации



Создаем авторизацию (страница AuthPage.xaml)

```
<Grid>
  <StackPanel Width="300" HorizontalAlignment="Center" VerticalAlignment="Center">
    <UniformGrid Columns="2">
      <TextBlock Text="Логин" VerticalAlignment="Center"/>
      <TextBox x:Name="LoginTextBox" Padding="5" VerticalContentAlignment="Center"/>
    </UniformGrid>
    <UniformGrid Columns="2">
      <TextBlock Text="Пароль"/>
      <PasswordBox FontSize="20" x:Name="PasswordPasswordBox" PasswordChar="." />
    </UniformGrid>
    <Button Content="Sign In" x:Name="SignInButton" Click="SignInButtonClick"/>
    <TextBlock Text="Зарегистрироваться" x:Name="RegTextBlock"
      MouseLeftButtonDown="RegTextBlockMouseLeftButtonDown"
      HorizontalAlignment="Right"/>
  </StackPanel>
</Grid>
```

Стилизуем авторизацию

```
<Style TargetType="UniformGrid">
    <Setter Property="Margin" Value="0 5"/>
</Style>
<Style TargetType="TextBox" >
    <Setter Property="Padding" Value="5"/>
</Style>
<Style TargetType="Button" >
    <Setter Property="Background" Value="{StaticResource SecondColor}"/>
    <Setter Property="Padding" Value="5"/>
    <Setter Property="Margin" Value="0 5"/>
</Style>
```


Реализуем фокусировку курсора

```
/// <summary>
/// Логика взаимодействия для AuthPage.xaml
/// </summary>
```

Ссылки: 3

```
public partial class AuthPage : Page
```

```
{
```

ссылка: 1

```
public AuthPage()
```

```
{
```

```
    InitializeComponent();
```

```
    LoginTextBox.Focus();
```

```
}
```

```
<Grid>
```

```
    <StackPanel Width="300" HorizontalAlignment="Center" VerticalAlignment="Center">
```

```
        <UniformGrid Columns="2">
```

```
            <TextBlock Text="Логин" VerticalAlignment="Center"/>
```

```
            <TextBox
```

```
                x:Name="LoginTextBox"
```

```
                TabIndex="0"
```

```
                Padding="5"
```

```
                VerticalContentAlignment="Center" />
```

```
        </UniformGrid>
```

```
        <UniformGrid Columns="2">
```

```
            <TextBlock Text="Пароль"/>
```

```
            <PasswordBox FontSize="20" x:Name="PasswordPasswordBox" PasswordChar="." TabIndex="1" />
```

```
        </UniformGrid>
```

```
        <Button Content="Sign In" x:Name="SignInButton" Click="SignInButtonClick" TabIndex="2"/>
```

```
        <TextBlock
```

```
            Text="Зарегистрироваться"
```

```
            x:Name="RegTextBlock"
```

```
            Focusable="True"
```

```
            HorizontalAlignment="Right"
```

```
            MouseLeftButtonDown="RegTextBlockMouseLeftButtonDown"
```

```
            KeyDown="RegTextBlockKeyDown"
```

```
        />
```

```
    </StackPanel>
```

```
</Grid>
```

Выполнение функций с помощью клавиш

```
/// <summary>
/// Нажатие левой клавиши мыши на "Зарегистрироваться"
/// </summary>
Ссылка: 2
private void RegTextBlockMouseLeftButtonDown(object sender, MouseButtonEventArgs e)
{
    ...
}

/// <summary>
/// Нажатие клавиши Enter на "Зарегистрироваться"
/// </summary>
ссылка: 1
private void RegTextBlockKeyDown(object sender, KeyEventArgs e)
{
    ...
    if (e.Key == Key.Enter)
    {
        ...
        RegTextBlockMouseLeftButtonDown(RegTextBlock, null);
        ...
    }
    ...
}
```

Создаем класс Users (в папке Models)

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using Newtonsoft.Json;  
  
namespace BeautyApp.Models  
{  
    Ссылка: 0  
    public class Users  
    {  
        Ссылка: 0  
        public int IdRole { get; set; }  
        Ссылка: 0  
        public string UserLogin { get; set; }  
        Ссылка: 0  
        public string UserPassword { get; set; }  
    }  
}
```

Создаем класс UsersController

Ссылка: 0

```
public class UsersController
```

```
{  
    /// <summary>  
    /// Авторизация  
    /// </summary>  
    /// <param name="login">Логин</param>  
    /// <param name="password">Пароль</param>  
    /// <returns>  
    /// Объект класса Users  
    /// </returns>  
    Ссылка: 0  
    public Users GetUser(string login, string password)  
    {  
        using (HttpClient client = new HttpClient())  
        {  
            string url = Manager.RootUrl + "Users/" + login + "/" + password;  
            HttpResponseMessage response = client.GetAsync($"{url}").Result;  
            var content = response.Content.ReadAsStringAsync();  
            var answer = JsonConvert.DeserializeObject<Users>(content.Result);  
            return answer;  
        }  
    }  
}
```

```
namespace BeautyApp.Controllers
```

```
{
```

Ссылка: 2

```
public class Manager
```

```
{  
    ...  
}
```

```
    public static string RootUrl = "http://127.0.0.1:50710/api/";
```

Сохранение информации о пользователе

```
/// <summary>  
/// Логика взаимодействия для App.xaml  
/// </summary>
```

Ссылок: 4

```
public partial class App : Application
```

```
{
```

ссылка: 1

```
public static Users currentUser { get; set; } = null;
```

```
}
```

Авторизация + сохранение информации о пользователе

```
/// <summary>
/// Авторизация
/// </summary>
ссылка: 1
private void SignInButton_Click(object sender, RoutedEventArgs e)
{
    if (LoginTextBox.Text == String.Empty || PasswordPasswordBox.Password == String.Empty)
    {
        MessageBox.Show("Вы не ввели логин или пароль!");
    }
    else
    {
        if (obj.GetUser(LoginTextBox.Text, PasswordPasswordBox.Password) != null)
        {
            Users activeUser = obj.GetUser(LoginTextBox.Text, PasswordPasswordBox.Password);
            App.CurrentUser = activeUser;
            this.NavigationService.Navigate(new MainPage());
        }
        else
        {
            MessageBox.Show("Такого пользователя нет!");
        }
    }
}
```

Информация о пользователе выводится в «шапке» главного окна

ссылка: 1

```
private void MainFrameNavigated(object sender, NavigationEventArgs e)
{
    if (App.CurrentUser!=null)
    {
        PersonTextBlock.Text = App.CurrentUser.UserLogin;
        LogInTextBlock.Text = "Выйти";
    }
    else
    {
        LogInTextBlock.Visibility = Visibility.Visible;
        LogInTextBlock.Text = "Войти";
    }
}
```