

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)**

Колледж информатики и программирования

ОТЧЁТ
По учебной практике

Специальность 09.02.07 «Информационные системы и программирование»
(код) (наименование)

Профессиональный модуль ПМ.01 Разработка модулей программного обеспечения для компьютерных систем
(код) (наименование)

Междисциплинарный курс МДК.01.02 Поддержка и тестирование программных модулей
(код) (наименование)

Выполнил:

Студент 4 курса 4ИСИП-421 учебной группы
(номер) (номер)

(подпись) Н.Д. Сакович
(инициалы, фамилия)

Проверил:

Руководитель практики от Колледжа
информатики и программирования
Преподаватель ВКК, к.п.н. Т.Г. Аксёнова
(квалификационная (инициалы, фамилия)
категория или звание,
должность)

Москва – 2024

Перечень работ, выполненных в ходе учебной практики

№ п/п	Виды работ	Оценка
1	Разработка программного модуля многооконного интерфейса	
2	Тестирование программного модуля доступа к БД	

СОДЕРЖАНИЕ

ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 1.....	4
ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 2.....	30

Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)
Колледж информатики и программирования

ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 1

**Тема: Разработка графического приложения для управления
пользователями информационной системы**

Студент: Сакович Н.Д.

Группа: 4ИСИП-421

Преподаватель: Аксёнова Т.Г.

Дата: 19.12.2024

РАЗРАБОТКА СТРАНИЦЫ АВТОРИЗАЦИИ ПОЛЬЗОВАТЕЛЕЙ.

СОЗДАНИЕ СЛОВАРЯ СТИЛЕЙ

Цель работы: приобрести практические навыки проектирования интерфейса приложения WPF, навигации по страницам WPF, подключения базы данных с помощью технологии Entity Framework ADO.NET, реализации страницы авторизации пользователей, создания и настройки словаря стилей.

На рисунке 1 представлена окно MainWindos и ее заполнение.

Скрипт БД:

```
-- 1. Создание базы данных
CREATE DATABASE [121_test_database];
GO

-- Подключение к созданной базе данных
USE [121_test_database];
GO

-- 2. Создание таблицы User
CREATE TABLE [dbo].[User]
(
    [ID] INT IDENTITY(1,1) PRIMARY KEY, -- Автоинкремент и первичный ключ
    [Login] NVARCHAR(50) NOT NULL,      -- Логин пользователя
    [Password] NVARCHAR(50) NOT NULL,   -- Пароль пользователя
    [Role] NVARCHAR(20) NOT NULL,        -- Роль пользователя (admin, user)
    [FIO] NVARCHAR(100) NOT NULL,       -- Полное имя пользователя
    [Photo] VARBINARY(MAX) NULL         -- Фото пользователя (опционально)
);
GO

-- 3. Вставка тестовых данных в таблицу User

-- 5 записей для роли "admin"
INSERT INTO [dbo].[User] ([Login], [Password], [Role], [FIO], [Photo])
VALUES
('admin1', 'adminpass1', 'admin', 'Иванов Иван Иванович', NULL),
('admin2', 'adminpass2', 'admin', 'Петров Петр Петрович', NULL),
('admin3', 'adminpass3', 'admin', 'Сидоров Сидор Сидорович', NULL),
('admin4', 'adminpass4', 'admin', 'Алексеев Алексей Алексеевич', NULL),
('admin5', 'adminpass5', 'admin', 'Васильев Василий Васильевич', NULL);

-- 5 записей для роли "user"
INSERT INTO [dbo].[User] ([Login], [Password], [Role], [FIO], [Photo])
VALUES
('user1', 'userpass1', 'user', 'Кузнецов Кузьма Кузьмич', NULL),
('user2', 'userpass2', 'user', 'Николаев Николай Николаевич', NULL),
('user3', 'userpass3', 'user', 'Михайлов Михаил Михайлович', NULL),
```

('user4', 'userpass4', 'user', 'Семенов Семен Семенович', NULL),
('user5', 'userpass5', 'user', 'Тимофеев Тимофей Тимофеевич', NULL);
GO

На рисунке 4 представлено окно MainWindow и ниже листинг к нему.

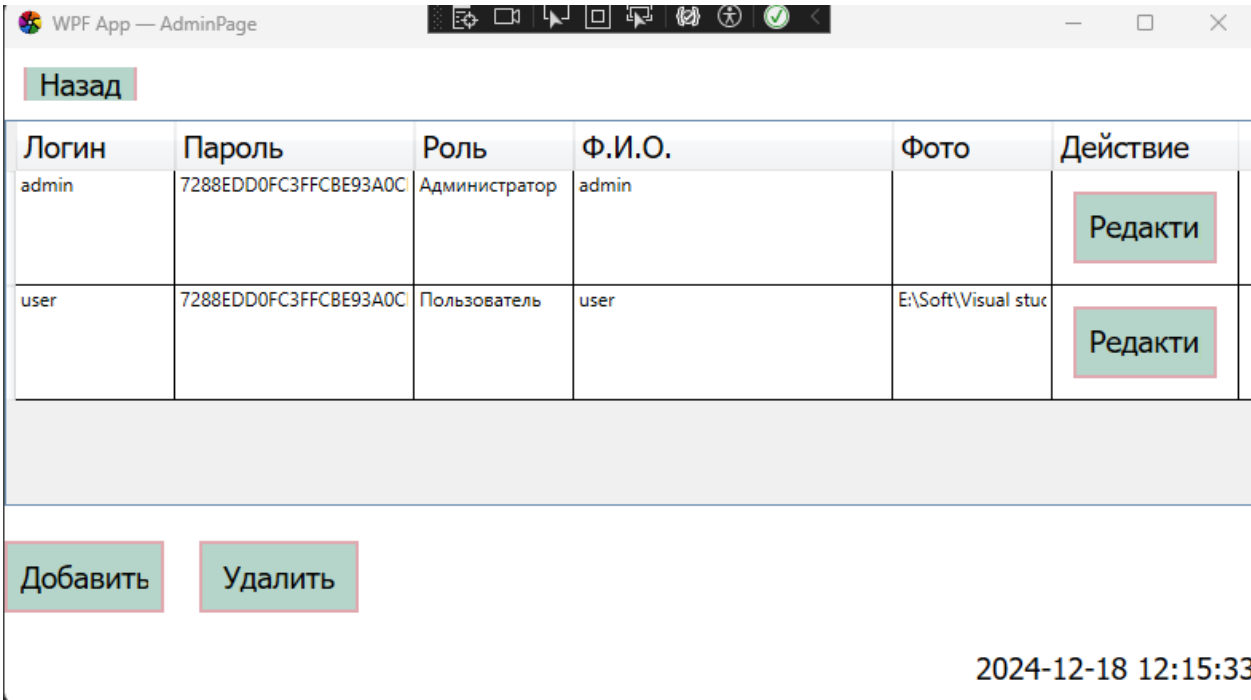


Рисунок 1. Окно MainWindow

MainWindow.xaml:

```
<Window x:Class="WpfApp1.MainWindow"
        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:WpfApp1"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800">

    <Grid>
        <!-- Определяем строки -->
        <Grid.RowDefinitions>
            <RowDefinition Height="45" />
            <RowDefinition Height="*" />
            <RowDefinition Height="45" />
        </Grid.RowDefinitions>

        <!-- Кнопка назад -->
        <Button Grid.Row="0" Content="Назад" Name="BackButton"
            HorizontalAlignment="Left" VerticalAlignment="Center"
            Click="BackButton_Click" Visibility="Hidden"/>

        <!-- Контейнер для Frame -->
        <Frame x:Name="MainFrame" Grid.Row="1" HorizontalAlignment="Stretch"
            VerticalAlignment="Stretch"
            Navigated="MainFrame_Navigated" Source="pages/AuthPage.xaml"
            NavigationUIVisibility="Hidden"/>
    </Grid>
</Window>
```

```

        <!-- Время в нижней строке -->
        <TextBlock Grid.Row="2" HorizontalAlignment="Right"
VerticalAlignment="Center" Name="TimeTextBlock"/>
    </Grid>
</Window>

```

MainWindow.xaml.cs:

```

using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Navigation;

namespace WpfApp1
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            InitializeClock();
        }

        // Обработчик события "Назад"
        private void BackButton_Click(object sender, RoutedEventArgs e)
        {
            if (MainFrame.CanGoBack)
                MainFrame.GoBack();
        }

        // Обновление времени
        private void InitializeClock()
        {
            var timer = new System.Windows.Threading.DispatcherTimer();
            timer.Interval = TimeSpan.FromSeconds(1);
            timer.Tick += (s, e) => { TimeTextBlock.Text =
DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss"); };
            timer.Start();
        }

        // Обработчик Navigated
        private void MainFrame_Navigated(object sender, NavigationEventArgs
e)
        {
            BackButton.Visibility = MainFrame.CanGoBack ? Visibility.Visible
: Visibility.Hidden;

            if (e.Content is Page page)
            {
                this.Title = $"WPF App – {page.Title}";
            }
        }
    }
}

```

На рисунке 5-6 показано подключение бд к VisualStudio, а далее демонстрация словаря стилей (рис.7) и его листинг под рисунком.

На рисунке 2-3 предоставлен код и результат подключения к БД:

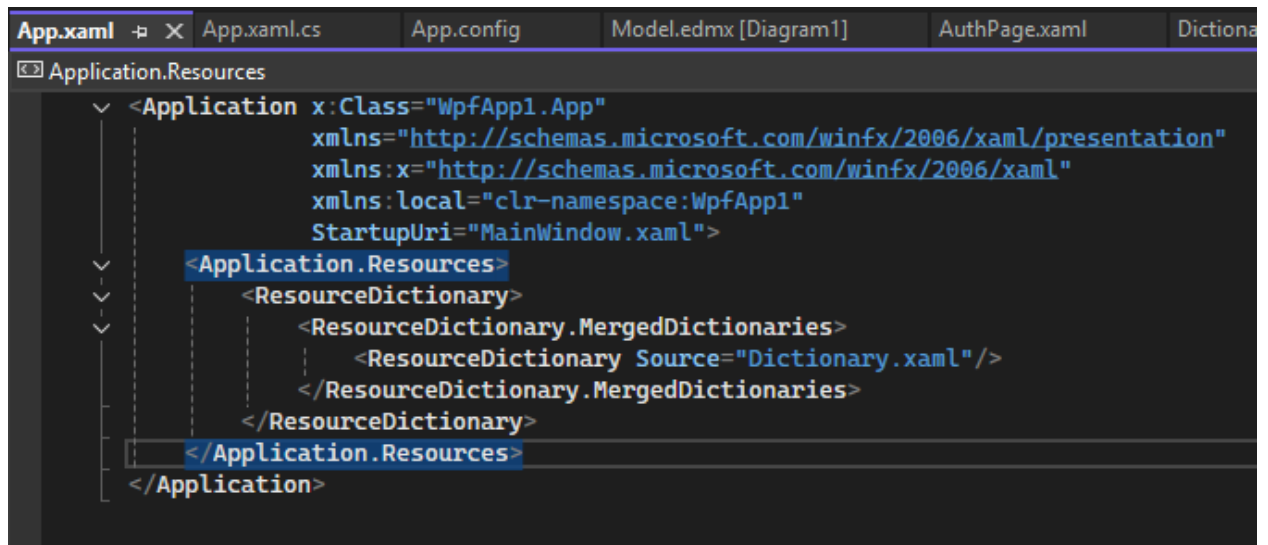


Рисунок 2. Подключение к БД

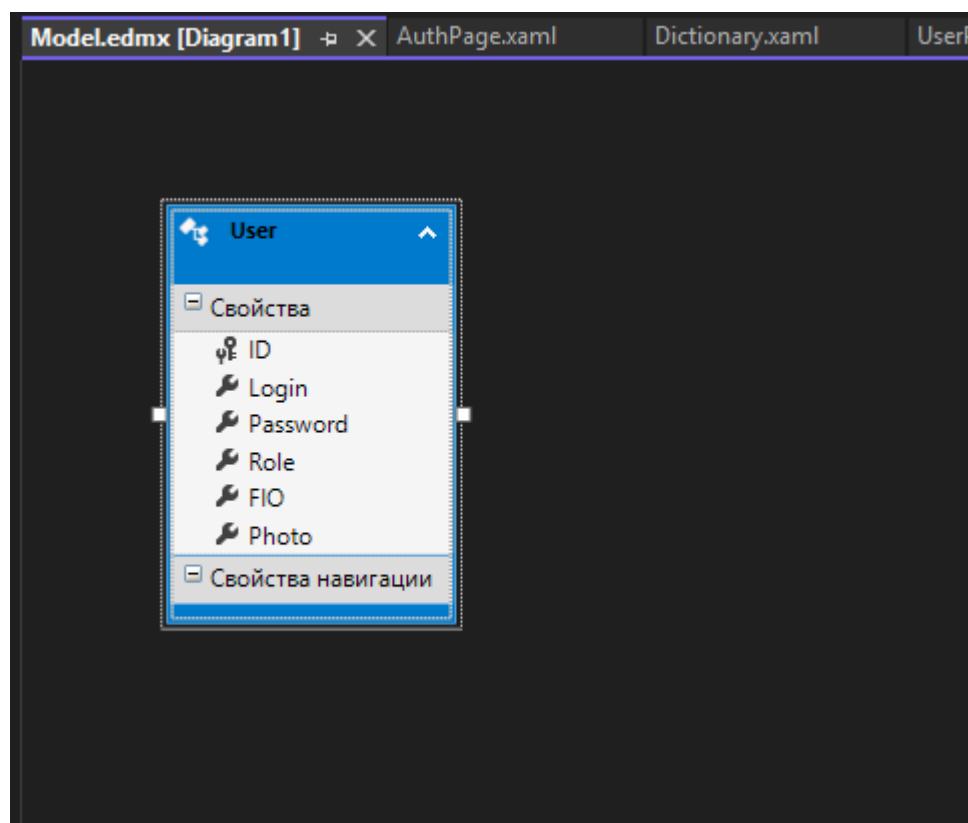


Рисунок 3. Результат подключения к БД

Словарь стилей:

```
<ResourceDictionary
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

  <!-- Основные цвета -->
  <Color x:Key="PrimaryColor">#B5D5CA</Color>
  <!-- Основной цвет 1 -->
  <Color x:Key="SecondaryColor">#E0A9AF</Color>
```



```

<!-- Основной цвет 2 -->

<!-- Дополнительные цвета -->
<Color x:Key="AccentColor1">#D1EEFC</Color>
<Color x:Key="AccentColor2">#FFFC6</Color>

<!-- Цвета для текста -->
<Color x:Key="TextWhite">#FFFFFF</Color>
<Color x:Key="TextBlack">#000000</Color>

<!-- Кисти на основе цветов -->
<SolidColorBrush x:Key="PrimaryBrush" Color="{StaticResource
PrimaryColor}"/>
<SolidColorBrush x:Key="SecondaryBrush" Color="{StaticResource
SecondaryColor}"/>
<SolidColorBrush x:Key="AccentBrush1" Color="{StaticResource
AccentColor1}"/>
<SolidColorBrush x:Key="AccentBrush2" Color="{StaticResource
AccentColor2}"/>
<SolidColorBrush x:Key="TextWhiteBrush" Color="{StaticResource
TextWhite}"/>
<SolidColorBrush x:Key="TextBlackBrush" Color="{StaticResource
TextBlack}"/>

<!-- Стил для кнопок -->
<Style TargetType="Button">
    <Setter Property="Height" Value="45"/>
    <Setter Property="Padding" Value="8"/>
    <Setter Property="Margin" Value="12"/>
    <Setter Property="FontFamily" Value="Tahoma"/>
    <Setter Property="FontSize" Value="12pt"/>
    <Setter Property="Background" Value="{StaticResource
PrimaryBrush}"/>
    <Setter Property="Foreground" Value="{StaticResource
TextBlackBrush}"/>
    <Setter Property="BorderBrush" Value="{StaticResource
SecondaryBrush}"/>
    <Setter Property="BorderThickness" Value="2"/>
</Style>

<!-- Стил для TextBlock -->
<Style TargetType="TextBlock">
    <Setter Property="Foreground" Value="{StaticResource
TextBlackBrush}"/>
    <Setter Property="FontFamily" Value="Tahoma"/>
    <Setter Property="FontSize" Value="14pt"/>
</Style>

<!-- Стил для фона окна -->
<Style TargetType="Window">
    <Setter Property="Background" Value="{StaticResource
AccentBrush2}"/>
</Style>

<Style TargetType="ComboBox">
    <Setter Property="FontFamily" Value="Tahoma"/>
    <Setter Property="FontSize" Value="12"/>
    <Setter Property="Padding" Value="5"/>
    <Setter Property="Margin" Value="10"/>
    <Setter Property="Background" Value="LightGray"/>
    <Setter Property="Foreground" Value="Black"/>
</Style>

</ResourceDictionary>

```

На рисунке 4 показана страница авторизации, далее листинг к этой странице.

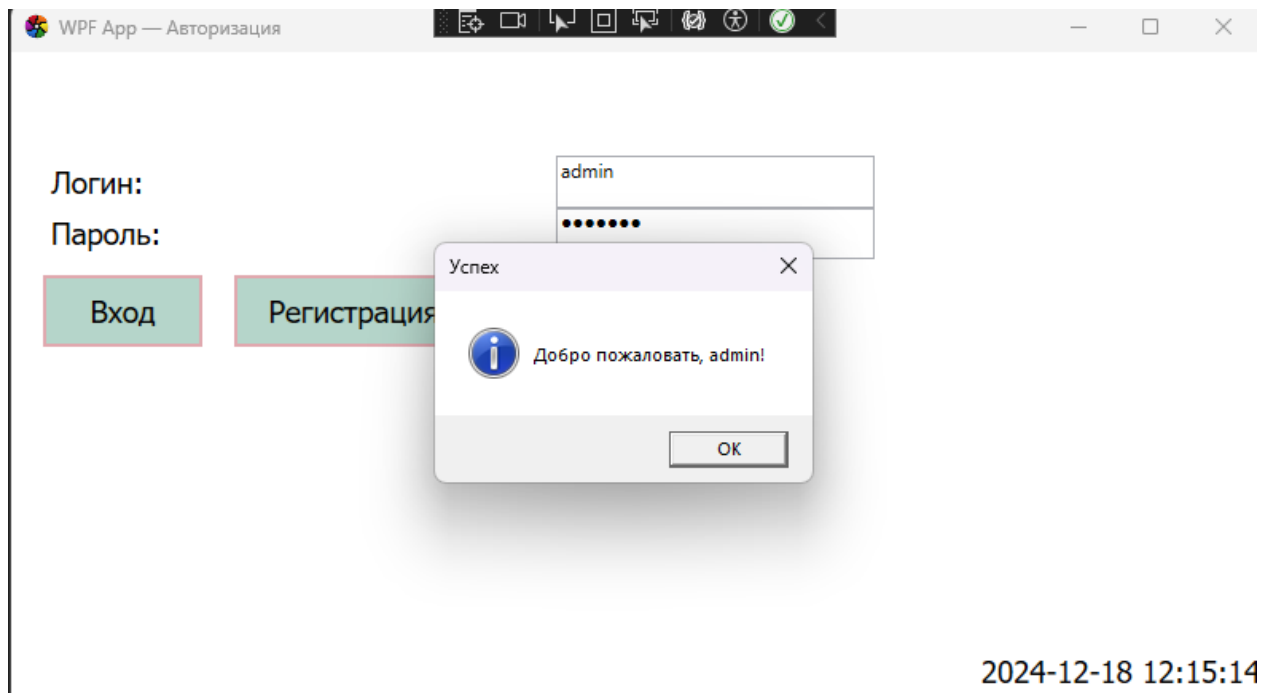


Рисунок 4. Страница авторизации

AuthPage.xaml:

```
<Page x:Class="WpfApp1.Pages.AuthPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Авторизация">

    <Grid Margin="20">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
        </Grid.RowDefinitions>

        <!-- Логин -->
        <Label Content="Логин:" Grid.Row="0" VerticalAlignment="Center"/>
        <TextBox x:Name="TextBoxLogin" Grid.Row="0" Margin="100,0,0,0"
Width="200"
                Background="Transparent" ToolTip="Введите ваш логин"/>

        <!-- Пароль -->
        <Label Content="Пароль:" Grid.Row="1" VerticalAlignment="Center"/>
        <PasswordBox x:Name="PasswordBoxPassword" Grid.Row="1"
Margin="100,0,0,0" Width="200"
                ToolTip="Введите ваш пароль"/>

        <!-- Кнопки -->
        <Button Content="Вход" Grid.Row="2" Margin="0,10,0,0" Width="100"
                HorizontalAlignment="Left" Click="LoginButton_Click"/>
    </Grid>
</Page>
```

```

        <Button Content="Регистрация" Grid.Row="2" Margin="120,10,0,0"
Width="150"
        HorizontalAlignment="Left" Click="RegisterButton_Click"/>
    </Grid>
</Page>

```

AuthPage.xaml.cs:

```

using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using WpfAppl.pages;

namespace WpfAppl.Pages
{
    public partial class AuthPage : Page
    {
        // Контекст данных для ADO.NET Entity Model
        private readonly Entities db = new Entities();

        public AuthPage()
        {
            InitializeComponent();
        }

        // Метод для хэширования пароля (SHA-1)
        public static string GetHash(string password)
        {
            using (var hash = SHA1.Create())
            {
                return
string.Concat(hash.ComputeHash(Encoding.UTF8.GetBytes(password))
                .Select(x => x.ToString("X2")));
            }
        }

        private void LoginButton_Click(object sender, RoutedEventArgs e)
        {
            string login = TextBoxLogin.Text.Trim();
            string password = PasswordBoxPassword.Password;

            // Проверка на пустые поля
            if (string.IsNullOrEmpty(login) ||
string.IsNullOrEmpty(password))
            {
                MessageBox.Show("Заполните все поля!", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                return;
            }

            // Хэширование пароля перед сравнением
            string hashedPassword = GetHash(password);

            // Поиск пользователя в базе данных
            var user = db.User.FirstOrDefault(u => u.Login == login &&
u.Password == hashedPassword);

            if (user != null)
            {
                MessageBox.Show($"Добро пожаловать, {user.FIO}!", "Успех",
MessageBoxButton.OK, MessageBoxImage.Information);

                // Переход на страницу в зависимости от роли

```

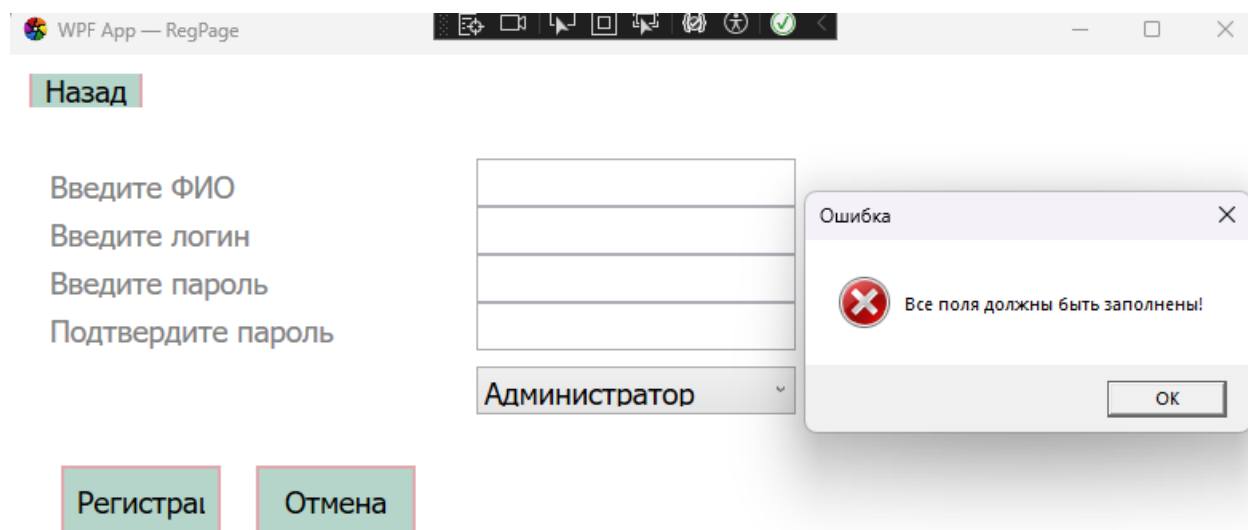
```

        if (user.Role == "Администратор")
            NavigationService.Navigate(new AdminPage());
        else if (user.Role == "Пользователь")
            NavigationService.Navigate(new UserPage());
        }
        else
        {
            MessageBox.Show("Неверные логин или пароль!", "Ошибка",
                MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }

    private void RegisterButton_Click(object sender, RoutedEventArgs e)
    {
        NavigationService.Navigate(new RegPage());
    }
}

```

На рисунках 5-7 происходит проверка заполненности полей и проверка на существующего пользователя страницы авторизации.



2024-12-18 12:18:32

Рисунок 5. Проверка

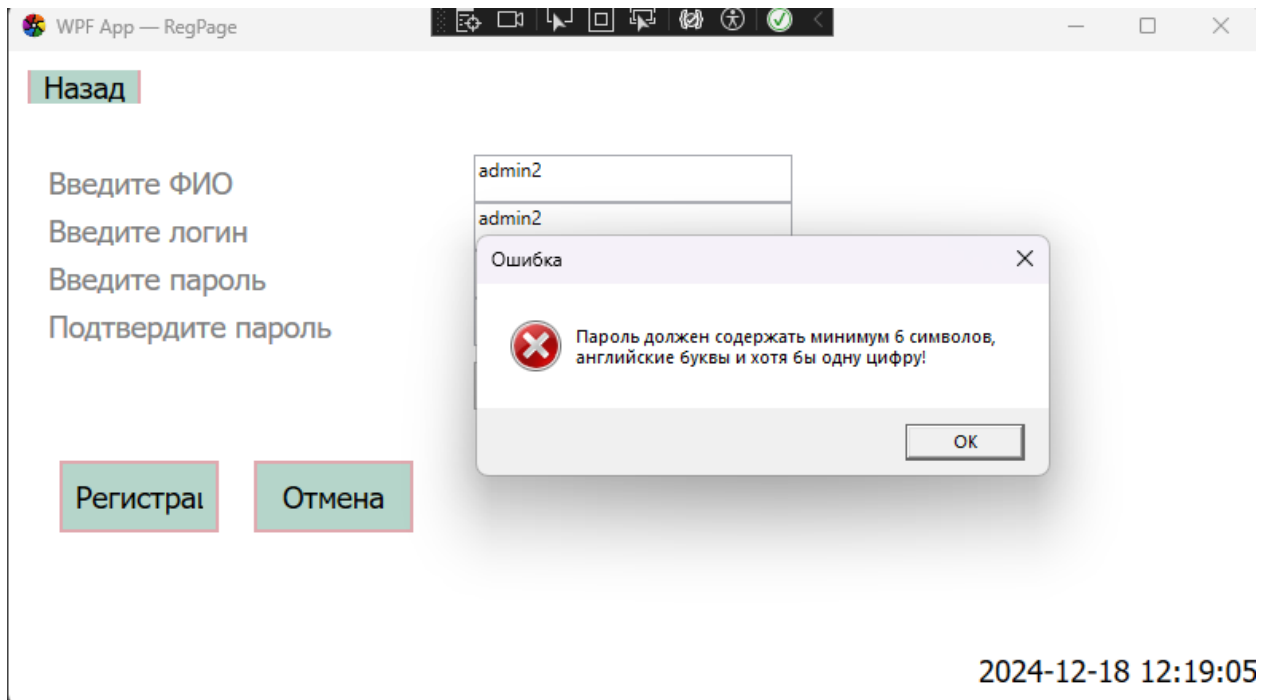


Рисунок 6. Проверка

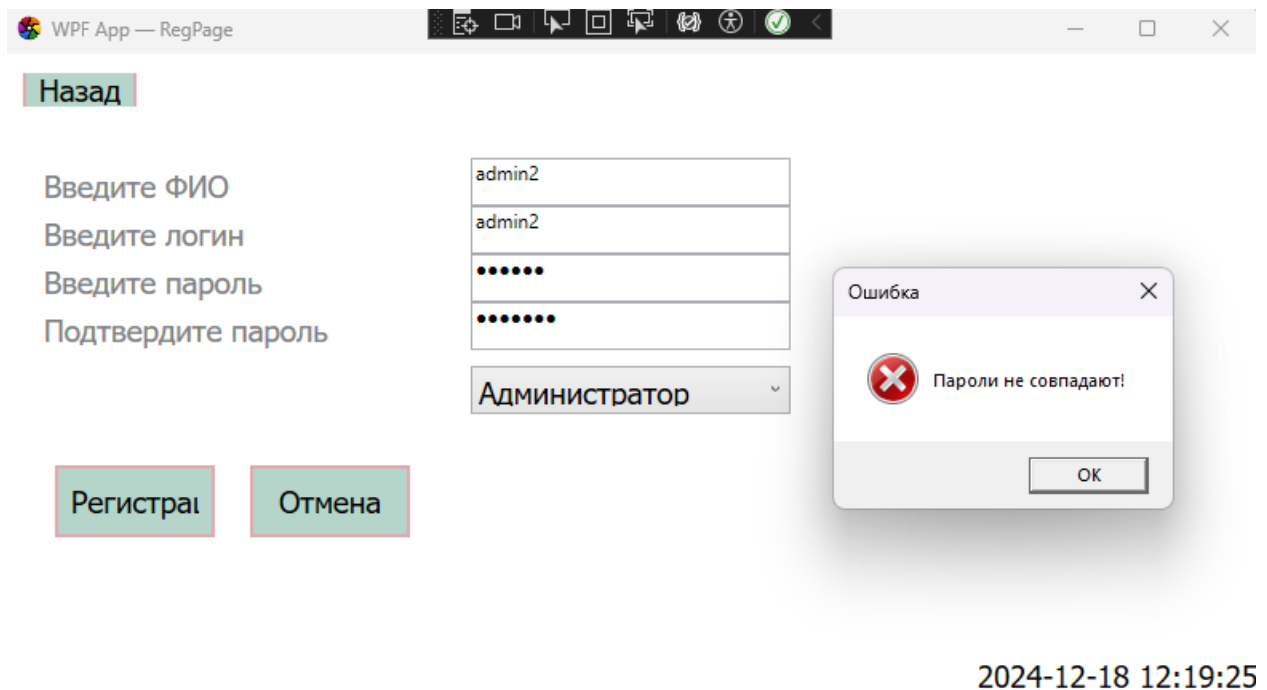


Рисунок 7. Проверка

На листинге предоставлена страница регистрации.
RegPage.xaml:

```
<Page x:Class="WpfApp1.pages.RegPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:WpfApp1.pages"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800">
```

```

        Title="RegPage">

        <Grid Margin="20">
            <Grid.RowDefinitions>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
            </Grid.RowDefinitions>

            <!-- ФИО -->
            <TextBox x:Name="TextBoxFIO" Grid.Row="0" Width="200" Height="30"
                Tooltip="Введите ваше ФИО" />
            <TextBlock Text="Введите ФИО" Grid.Row="0" Margin="5,5,0,0"
                VerticalAlignment="Center" Foreground="Gray"
                IsHitTestVisible="False"/>

            <!-- Логин -->
            <TextBox x:Name="TextBoxLogin" Grid.Row="1" Width="200" Height="30"
                Tooltip="Введите логин" />
            <TextBlock Text="Введите логин" Grid.Row="1" Margin="5,5,0,0"
                VerticalAlignment="Center" Foreground="Gray"
                IsHitTestVisible="False"/>

            <!-- Пароль -->
            <PasswordBox x:Name="PasswordBoxPassword" Grid.Row="2" Width="200"
                Height="30"
                Tooltip="Введите пароль"/>
            <TextBlock Text="Введите пароль" Grid.Row="2" Margin="5,5,0,0"
                VerticalAlignment="Center" Foreground="Gray"
                IsHitTestVisible="False"/>

            <!-- Подтверждение пароля -->
            <PasswordBox x:Name="PasswordBoxConfirm" Grid.Row="3" Width="200"
                Height="30"
                Tooltip="Подтвердите пароль"/>
            <TextBlock Text="Подтвердите пароль" Grid.Row="3" Margin="5,5,0,0"
                VerticalAlignment="Center" Foreground="Gray"
                IsHitTestVisible="False"/>

            <!-- Роль пользователя -->
            <ComboBox x:Name="CmbRole" Grid.Row="4" Width="200" Height="30"
                Tooltip="Выберите роль пользователя" SelectedIndex="0">
                <ComboBoxItem Content="Администратор" />
                <ComboBoxItem Content="Пользователь" />
            </ComboBox>

            <!-- Кнопки -->
            <StackPanel Grid.Row="5" Orientation="Horizontal" Margin="0,10,0,0">
                <Button Content="Регистрация" Width="100"
                    Click="RegisterButton_Click"/>
                <Button Content="Отмена" Width="100" Margin="10,0,0,0"
                    Click="CancelButton_Click"/>
            </StackPanel>
        </Grid>
    </Page>

```

RegPage.xaml.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;

```

```

using System.Security.Cryptography;
using System.Text;
using System.Text.RegularExpressions;
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;
using WpfAppl.Pages;

namespace WpfAppl.pages
{
    /// <summary>
    /// Логика взаимодействия для RegPage.xaml
    /// </summary>
    public partial class RegPage : Page
    {
        private readonly Entities db = new Entities();

        public RegPage()
        {
            InitializeComponent();

            // Метод для хэширования пароля (SHA-1)
            public static string GetHash(string password)
            {
                using (var hash = SHA1.Create())
                {
                    return
string.Concat(hash.ComputeHash(Encoding.UTF8.GetBytes(password))
                    .Select(x => x.ToString("X2")));
                }
            }

            private void RegisterButton_Click(object sender, RoutedEventArgs e)
            {
                string fio = TextBoxFIO.Text.Trim();
                string login = TextBoxLogin.Text.Trim();
                string password = PasswordBoxPassword.Password;
                string confirmPassword = PasswordBoxConfirm.Password;
                string role = CmbRole.Text;

                // Проверка на пустые поля
                if (string.IsNullOrEmpty(fio) || string.IsNullOrEmpty(login) ||
                    string.IsNullOrEmpty(password) ||
string.IsNullOrEmpty(confirmPassword))
                {
                    MessageBox.Show("Все поля должны быть заполнены!", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                    return;
                }

                // Проверка на существующий логин
                if (db.User.Any(u => u.Login == login))
                {
                    MessageBox.Show("Логин уже существует!", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                    return;
                }
            }
        }
    }
}

```

```

    }

    // Проверка формата пароля
    if (password.Length < 6 ||
!System.Text.RegularExpressions.Regex.IsMatch(password, @"^[a-zA-Z0-9]+$")
||
        !password.Any(char.IsDigit))
    {
        MessageBox.Show("Пароль должен содержать минимум 6 символов,
английские буквы и хотя бы одну цифру!",
            "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
        return;
    }

    // Проверка на совпадение паролей
    if (password != confirmPassword)
    {
        MessageBox.Show("Пароли не совпадают!", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
        return;
    }

    // Добавление пользователя с хэшированным паролем
    User newUser = new User
    {
        FIO = fio,
        Login = login,
        Password = GetHash(password), // Хэшируем пароль перед
сохранением
        Role = role
    };

    db.User.Add(newUser);
    db.SaveChanges();

    MessageBox.Show("Регистрация успешна!", "Успех",
MessageBoxButton.OK, MessageBoxImage.Information);

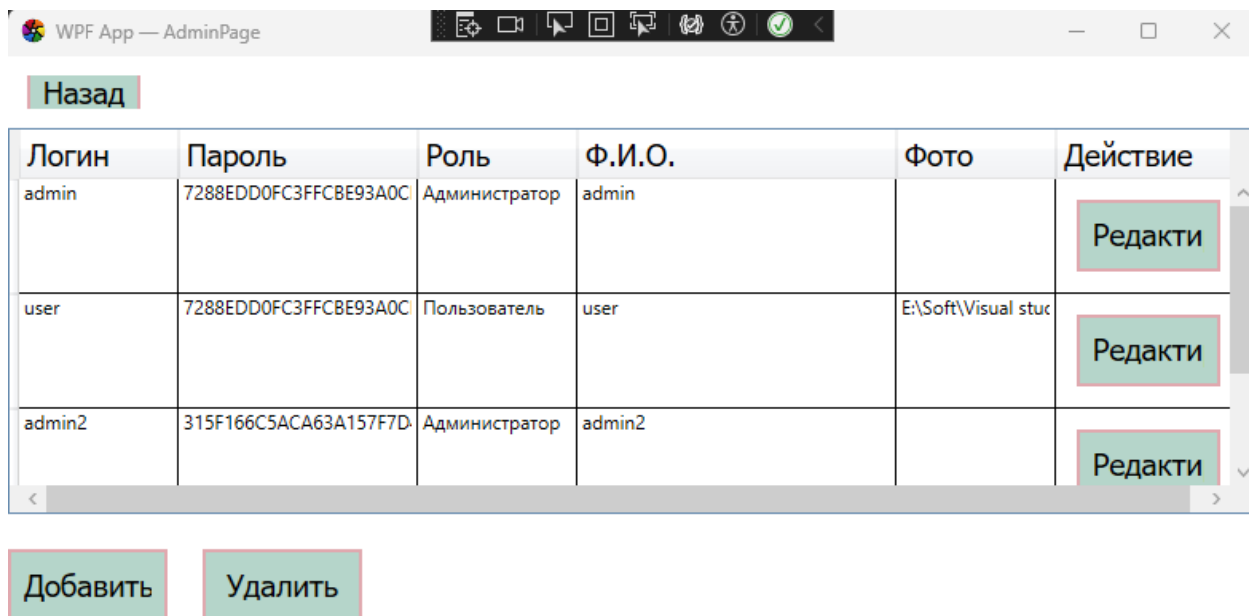
    // Переход на страницу авторизации
    NavigationService.Navigate(new AuthPage());
}

private void CancelButton_Click(object sender, RoutedEventArgs e)
{
    // Очистка полей
    TextBoxFIO.Clear();
    TextBoxLogin.Clear();
    PasswordBoxPassword.Clear();
    PasswordBoxConfirm.Clear();
    CmbRole.SelectedIndex = 0;

    // Возврат на страницу авторизации
    NavigationService.Navigate(new AuthPage());
}
}

```

Далее представлена страница регистрации и ее проверки.



2024-12-18 12:20:38

Рисунок 8. Результат

После проверок пароли захешировались.

DESKTOP-S4HTB4O\...aykova - dbo.User		DESKTOP-S4HTB4O\...ment - Diagram_0*					SQLQuery
	Id	Login	Password	Role	FIO	Photo	
►	1	admin1	password1	администр...	Иван Иван...	C:\учеба\01...	
	2	admin2	password2	администр...	Петр Петров	C:\учеба\01...	
	3	admin3	password3	администр...	Сергей Сер...	C:\учеба\01...	
	4	admin4	password4	администр...	Алексей Ал...	C:\учеба\01...	
	5	admin5	password5	администр...	Дмитрий Д...	C:\учеба\01...	
	6	user1	password6	пользовате...	Анна Анто...	C:\учеба\01...	
	7	user2	password7	пользовате...	Елена Елен...	C:\учеба\01...	
	8	user3	password8	пользовате...	Олег Олегов	C:\учеба\01...	
	9	user4	password9	пользовате...	Мария Мар...	C:\учеба\01...	
	10	user5	password10	пользовате...	Николай Н...	C:\учеба\01...	
	12	admin6	password6	администр...	Иванов Ив...	C:\учеба\01...	
	17	admin7	330BA60E243186E9FA25...	администр...	Кучин Егор...	C:\учеба\01...	
	20	admin8	A8DBBFA41CEC833F8DD...	администр...	Ковальчук ...	C:\учеба\01...	
	21	user10	D089DA97B9E447158A0...	пользовате...	Ковалев Ал...	C:\учеба\01...	
	22	admin9	024B01916E3EAEC66A2C...	администр...	Крепчук Иг...	NULL	
	23	user11	DDF6C9A1DF4D57AEF04...	пользовате...	Шишкова А...	NULL	
*	NULL	NULL	NULL	NULL	NULL	NULL	

Рисунок 9. Пароли хешируются

AddUserPage.xaml:

```
<Page x:Class="WpfApp1.pages.AddUserPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
```

```

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:local="clr-namespace:WpfAppl.pages"
mc:Ignorable="d"
d:DesignHeight="450" d:DesignWidth="800"
Title="AddUserPage">

<Grid Margin="20">
    <Grid.RowDefinitions>
        <RowDefinition Height="*"/>
        <RowDefinition Height="*"/>
        <RowDefinition Height="*"/>
        <RowDefinition Height="*"/>
        <RowDefinition Height="*"/>
        <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="200"/>
        <ColumnDefinition Width="*"/>
    </Grid.ColumnDefinitions>

    <!-- Метки и поля ввода -->
    <TextBlock Text="Логин" Grid.Row="0" Grid.Column="0"
Margin="10,0,0,0"/>
    <TextBox x:Name="TextBoxLogin" Grid.Row="0" Grid.Column="1"
ToolTip="Введите логин"/>

    <TextBlock Text="Пароль" Grid.Row="1" Grid.Column="0"
Margin="10,0,0,0"/>
    <TextBox x:Name="TextBoxPassword" Grid.Row="1" Grid.Column="1"
ToolTip="Введите пароль"/>

    <TextBlock Text="Роль" Grid.Row="2" Grid.Column="0"
Margin="10,0,0,0"/>
    <ComboBox x:Name="CmbRole" Grid.Row="2" Grid.Column="1"
ToolTip="Выберите роль">
        <ComboBoxItem Content="Администратор"/>
        <ComboBoxItem Content="Пользователь"/>
    </ComboBox>

    <TextBlock Text="ФИО" Grid.Row="3" Grid.Column="0"
Margin="10,0,0,0"/>
    <TextBox x:Name="TextBoxFIO" Grid.Row="3" Grid.Column="1"
ToolTip="Введите ФИО"/>

    <TextBlock Text="Фото (необязательно)" Grid.Row="4" Grid.Column="0"
Margin="10,0,0,0"/>
    <TextBox x:Name="TextBoxPhoto" Grid.Row="4" Grid.Column="1"
ToolTip="Введите путь к фото"/>

    <!-- Кнопки -->
    <StackPanel Grid.Row="5" Grid.ColumnSpan="2"
Orientation="Horizontal" HorizontalAlignment="Right">
        <Button x:Name="ButtonSave" Content="Сохранить" Width="100"
Margin="0,0,10,0" Click="ButtonSave_Click"/>
        <Button x:Name="ButtonCancel" Content="Отмена" Width="100"
Click="ButtonCancel_Click"/>
    </StackPanel>
</Grid>
</Page>

```

AddPage.xaml.cs:

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Security.Cryptography;
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 WpfApp1.pages
{
    /// <summary>
    /// Логика взаимодействия для AddUserPage.xaml
    /// </summary>
    public partial class AddUserPage : Page
    {
        private User _currentUser;

        public static string GetHash(string input)
        {
            using (var sha1 = SHA1.Create())
            {
                byte[] data =
                sha1.ComputeHash(Encoding.UTF8.GetBytes(input));
                return string.Concat(data.Select(b => b.ToString("X2"))); //
                Возвращает хеш в виде строки в верхнем регистре
            }
        }

        public AddUserPage(User selectedUser = null)
        {
            InitializeComponent();

            // Если передан пользователь для редактирования, используем его,
            // иначе создаем нового
            _currentUser = selectedUser ?? new User();
            DataContext = _currentUser;

            // Заполняем поля при редактировании
            TextBoxLogin.Text = _currentUser.Login; // Заполняем логин
            TextBoxFIO.Text = _currentUser.FIO; // Заполняем ФИО
            TextBoxPhoto.Text = _currentUser.Photo; // Заполняем путь к фото

            // Установка значения роли в ComboBox
            if (!string.IsNullOrEmpty(_currentUser.Role))
            {
                // Сопоставляем текущую роль с элементами ComboBox
                CmbRole.SelectedItem = CmbRole.Items
                    .Cast<ComboBoxItem>()
                    .FirstOrDefault(item => item.Content.ToString() ==
                    _currentUser.Role);
            }
            else
            {
                CmbRole.SelectedIndex = 0; // Значение по умолчанию
            }

            // Поле пароля оставляем пустым для редактирования
            TextBoxPassword.Text = string.Empty;
        }
    }
}

```

```

    }

    // Обработчик кнопки "Сохранить"
    private void ButtonSave_Click(object sender, RoutedEventArgs e)
    {
        StringBuilder errors = new StringBuilder();

        // Обновляем данные модели вручную перед сохранением
        _currentUser.Login = TextBoxLogin.Text.Trim();
        _currentUser.Password = GetHash(TextBoxPassword.Text.Trim());
        _currentUser.Role = (CmbRole.SelectedItem as
ComboBoxItem)?.Content.ToString();
        _currentUser.FIO = TextBoxFIO.Text.Trim();
        _currentUser.Photo = TextBoxPhoto.Text.Trim() ?? string.Empty;

        // Проверка на обязательные поля
        if (string.IsNullOrEmpty(_currentUser.Login))
            errors.AppendLine("Укажите логин!");
        if (string.IsNullOrEmpty(_currentUser.Password))
            errors.AppendLine("Укажите пароль!");
        if (string.IsNullOrEmpty(_currentUser.Role))
            errors.AppendLine("Выберите роль!");
        if (string.IsNullOrEmpty(_currentUser.FIO))
            errors.AppendLine("Укажите ФИО!");

        // Если есть ошибки, выводим их и выходим из метода
        if (errors.Length > 0)
        {
            MessageBox.Show(errors.ToString(), "Ошибка ввода",
MessageBoxButton.OK, MessageBoxImage.Error);
            return;
        }

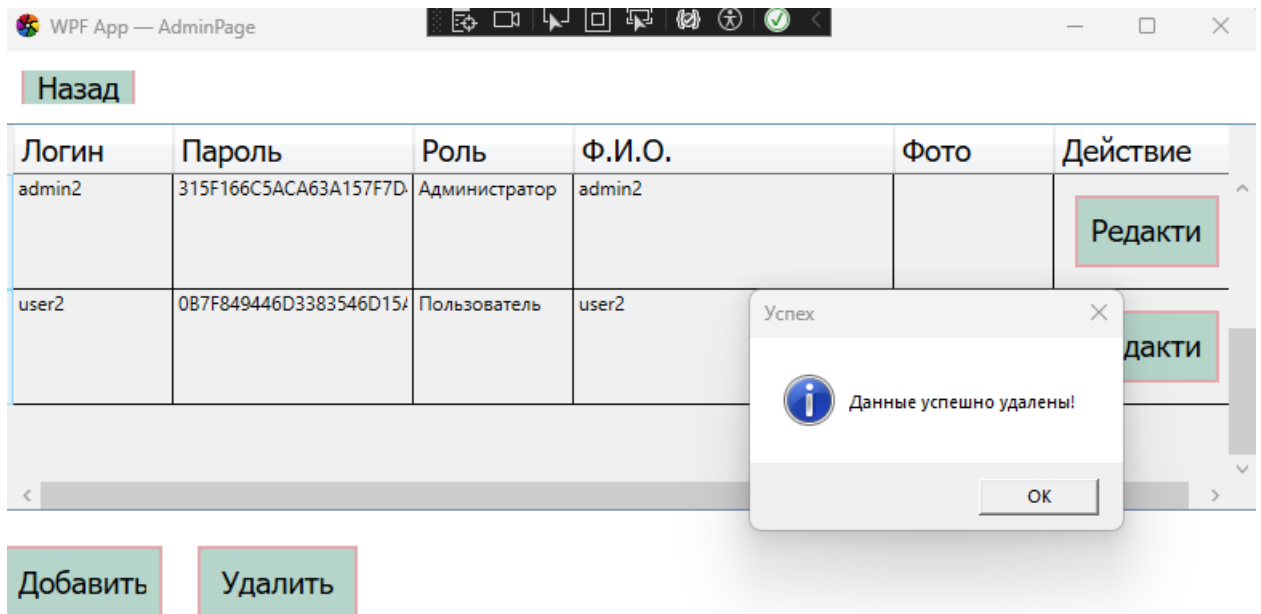
        // Добавление нового пользователя
        if (_currentUser.ID == 0) // ID == 0 означает новый пользователь
            Entities.GetContext().User.Add(_currentUser);

        // Попытка сохранения данных
        try
        {
            Entities.GetContext().SaveChanges();
            MessageBox.Show("Данные успешно сохранены!", "Успех",
MessageBoxButton.OK, MessageBoxImage.Information);
            NavigationService.GoBack();
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Ошибка при сохранении данных:
{ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }

    // Обработчик кнопки "Отмена"
    private void ButtonCancel_Click(object sender, RoutedEventArgs e)
    {
        NavigationService.GoBack();
    }
}

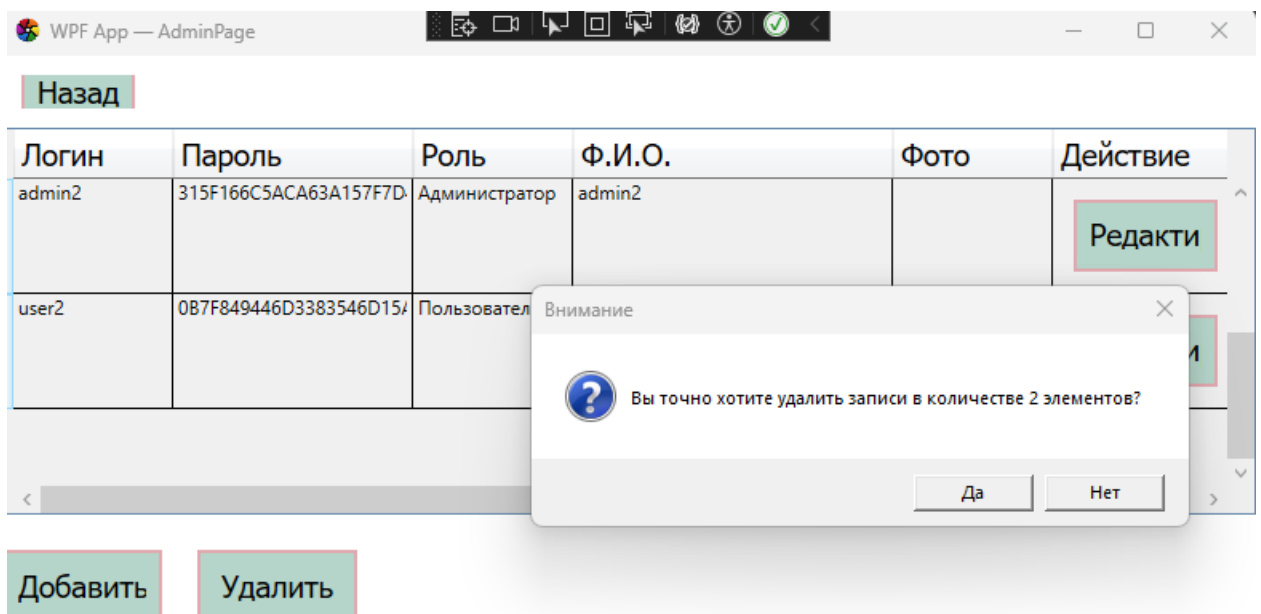
```

На рисунках 10-11 представлены проверки страницы.



2024-12-18 12:24:20

Рисунок 10. Изменения



2024-12-18 12:24:07

Рисунок 11. Проверка

На рисунке 28 показана страница для вывода информации о пользователях от имени пользователя с названием ManagerMenu, а ниже листинг к ней.

UserPage.xaml:

```
<Page x:Class="WpfApp1.pages.UserPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```

```

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expressions/blend/2008"
xmlns:local="clr-namespace:WpfAppl.pages"
mc:Ignorable="d"
d:DesignHeight="450" d:DesignWidth="800"
Title="UserPage">

<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto" />
        <RowDefinition Height="*" />
    </Grid.RowDefinitions>

    <!-- Панель для поиска, фильтрации и сортировки -->
    <WrapPanel Grid.Row="0" HorizontalAlignment="Center" Margin="10">
        <!-- Поиск по Ф.И.О. -->
        <TextBlock Text="Поиск по Ф.И.О.:" Margin="5"
VerticalAlignment="Center" />
        <TextBox x:Name="TextBoxSearch" Width="200" Margin="5"
            ToolTip="Введите Ф.И.О. пользователя для поиска"
            TextChanged="TextBoxSearch_TextChanged" />

        <!-- Сортировка -->
        <TextBlock Text="Сортировка:" Margin="5"
VerticalAlignment="Center" />
        <ComboBox x:Name="CmbSorting" Width="150" Margin="5"
            SelectionChanged="CmbSorting_SelectionChanged"
            ToolTip="Выберите порядок сортировки">
            <ComboBoxItem Content="По возрастанию" />
            <ComboBoxItem Content="По убыванию" />
        </ComboBox>

        <!-- Фильтр по ролям -->
        <CheckBox x:Name="CheckDriver" Content="Только пользователи"
Margin="5"
            Checked="CheckDriver_Checked"
Unchecked="CheckDriver_Unchecked"
            ToolTip="Показать только пользователей" />

        <!-- Очистить фильтр -->
        <Button Content="Очистить фильтр" Margin="5" Width="120"
            Click="ButtonClear_Click" ToolTip="Сбросить поиск и
фильтры" />
    </WrapPanel>

    <!-- ListView для отображения пользователей -->
    <ListView Name="ListUser" Grid.Row="1"
        ScrollViewer.HorizontalScrollBarVisibility="Disabled"
        HorizontalContentAlignment="Center">
        <ListView.ItemsPanel>
            <ItemsPanelTemplate>
                <WrapPanel Orientation="Horizontal"
HorizontalAlignment="Center" />
            </ItemsPanelTemplate>
        </ListView.ItemsPanel>

        <ListView.ItemTemplate>
            <DataTemplate>
                <Grid Margin="10" Width="450" Height="600"
Background="LightGray">
                    <Grid.RowDefinitions>
                        <RowDefinition Height="Auto" />
                        <RowDefinition Height="450" />
                    </Grid.RowDefinitions>
                </Grid>
            </DataTemplate>
        </ListView.ItemTemplate>
    </ListView>
</Grid>

```

```

        <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>

    <!-- ФИО пользователя -->
    <TextBlock VerticalAlignment="Center"
TextAlignment="Center"
Width="450" Grid.Row="0"
TextWrapping="Wrap"
Text="{Binding FIO}"
HorizontalAlignment="Center"
FontSize="18" FontWeight="Bold" />

    <!-- Изображение пользователя -->
    <Image Width="450" Grid.Row="1"
Stretch="UniformToFill" HorizontalAlignment="Center">
        <Image.Source>
            <Binding Path="Photo" >
                <Binding.TargetNullValue>

<ImageSource>/Resources/DefaultPhoto.png</ImageSource>
                </Binding.TargetNullValue>
            </Binding>
        </Image.Source>
    </Image>

    <!-- Роль пользователя -->
    <TextBlock VerticalAlignment="Center"
TextAlignment="Left"
Width="450" Grid.Row="2"
HorizontalAlignment="Center"
FontSize="18" Text="{Binding Role}" />

    </Grid>
</DataTemplate>
</ListView.ItemTemplate>
</ListView>
</Grid>
</Page>

```

UserPage.xaml.cs:

```

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 WpfApp1.pages
{
    /// <summary>
    /// Логика взаимодействия для UserPage.xaml
    /// </summary>
    public partial class UserPage : Page
    {
        public UserPage()
        {
            InitializeComponent();

```

```

        CmbSorting.SelectedIndex = 0; // Установка сортировки по
умолчанию
        CheckDriver.IsChecked = false; // Сбрасываем фильтр
        UpdateUsers(); // Начальная загрузка данных
    }

    // Метод для обновления данных
    private void UpdateUsers()
    {
        // Загружаем всех пользователей
        var currentUsers = Entities.GetContext().User.ToList();

        // Поиск по Ф.И.О. (без учета регистра)
        if (!string.IsNullOrEmpty(TextBoxSearch.Text))
        {
            currentUsers = currentUsers.Where(x => x.FIO.ToLower()
                .Contains(TextBoxSearch.Text.ToLower())).ToList();
        }

        // Фильтрация: показываем только пользователей
        if (CheckDriver.IsChecked == true)
        {
            currentUsers = currentUsers.Where(x =>
x.Role.Contains("Пользователь")).ToList();
        }

        // Сортировка
        if (CmbSorting.SelectedIndex == 0) // По возрастанию
        {
            currentUsers = currentUsers.OrderBy(x => x.FIO).ToList();
        }
        else // По убыванию
        {
            currentUsers = currentUsers.OrderByDescending(x =>
x.FIO).ToList();
        }

        // Устанавливаем данные в ListView
        ListUser.ItemsSource = currentUsers;
    }

    // Обработчик поиска
    private void TextBoxSearch_TextChanged(object sender,
TextChangedEventArgs e)
    {
        UpdateUsers();
    }

    // Обработчик изменения сортировки
    private void CmbSorting_SelectionChanged(object sender,
SelectionChangedEventArgs e)
    {
        UpdateUsers();
    }

    // Обработчик фильтрации
    private void CheckDriver_Checked(object sender, RoutedEventArgs e)
    {
        UpdateUsers();
    }

    private void CheckDriver_Unchecked(object sender, RoutedEventArgs e)
    {
        UpdateUsers();
    }

```



```

    }

    // Обработчик кнопки "Очистить фильтр"
    private void ButtonClear_Click(object sender, RoutedEventArgs e)
    {
        TextBoxSearch.Clear();
        CmbSorting.SelectedIndex = 0;
        CheckDriver.IsChecked = false;
        UpdateUsers();
    }
}

```

На рисунках 12-15 предоставлен фильтр пользователей и пользователи с иконками.

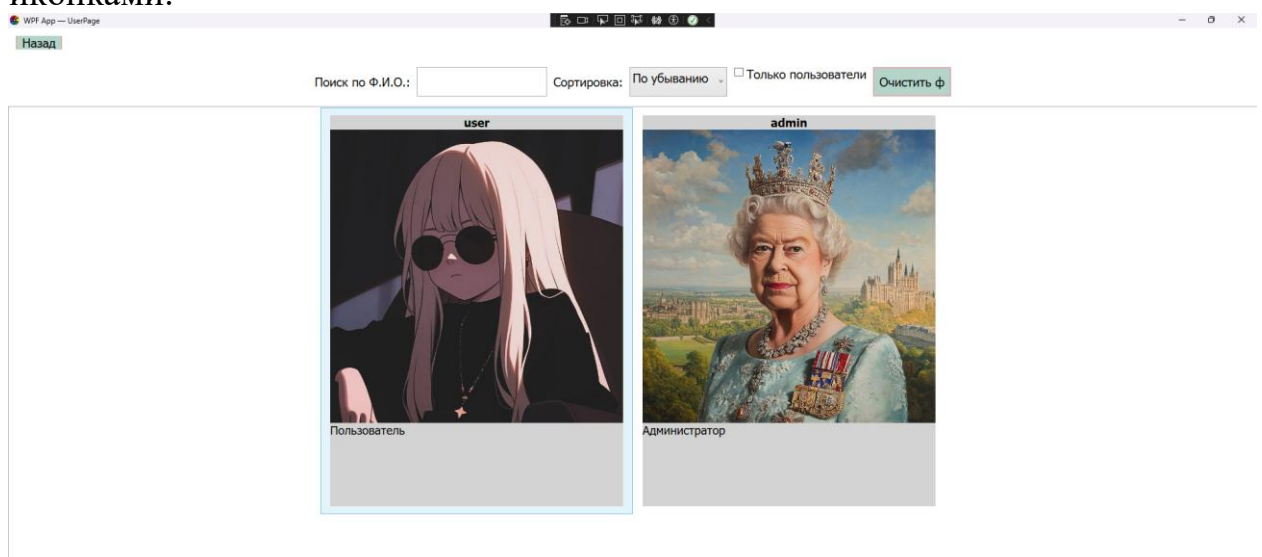


Рисунок 12. Пользователи



Рисунок 13. Фильтр

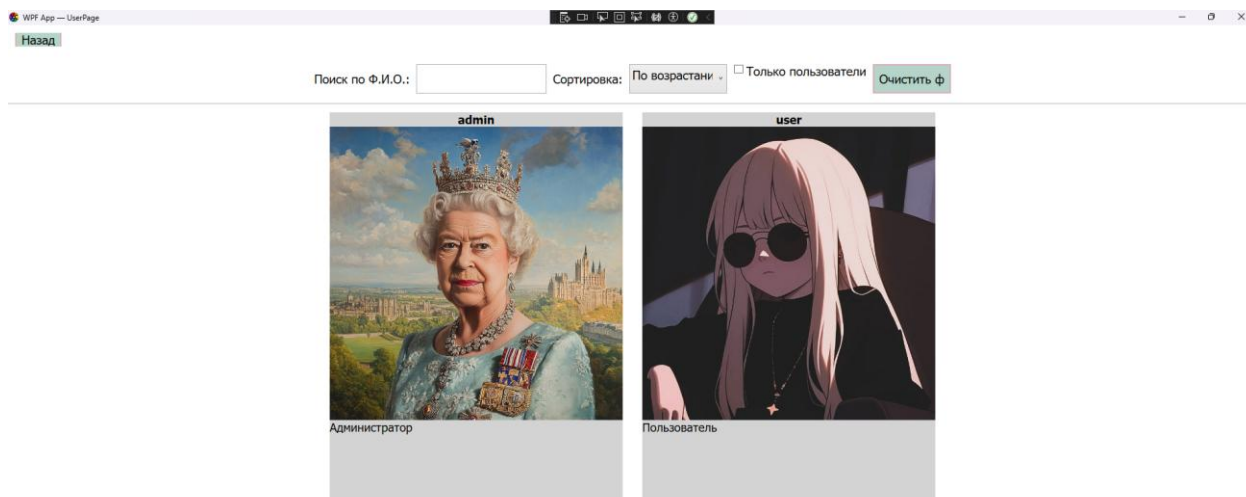


Рисунок 14. Очистка фильтра

Листинг UserPage.xaml

```
<Page x:Class="WpfApp1.pages.UserPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:WpfApp1.pages"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="UserPage">

    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <!-- Панель для поиска, фильтрации и сортировки -->
        <WrapPanel Grid.Row="0" HorizontalAlignment="Center" Margin="10">
            <!-- Поиск по Ф.И.О. -->
            <TextBlock Text="Поиск по Ф.И.О.:" Margin="5"
                VerticalAlignment="Center" />
            <TextBox x:Name="TextBoxSearch" Width="200" Margin="5"
                ToolTip="Введите Ф.И.О. пользователя для поиска"
                TextChanged="TextBoxSearch_TextChanged" />

            <!-- Сортировка -->
            <TextBlock Text="Сортировка:" Margin="5"
                VerticalAlignment="Center" />
            <ComboBox x:Name="CmbSorting" Width="150" Margin="5"
                SelectionChanged="CmbSorting_SelectionChanged"
                ToolTip="Выберите порядок сортировки">
                <ComboBoxItem Content="По возрастанию" />
                <ComboBoxItem Content="По убыванию" />
            </ComboBox>

            <!-- Фильтр по ролям -->
            <CheckBox x:Name="CheckDriver" Content="Только пользователи"
                Margin="5"
                Checked="CheckDriver_Checked"
                Unchecked="CheckDriver_Unchecked"
                ToolTip="Показать только пользователей" />
        </WrapPanel>
    </Grid>
</Page>
```

```

        <!-- Очистить фильтр -->
        <Button Content="Очистить фильтр" Margin="5" Width="120"
                Click="ButtonClear_Click" ToolTip="Сбросить поиск и
фильтры" />
    </WrapPanel>

    <!-- ListView для отображения пользователей -->
    <ListView Name="ListUser" Grid.Row="1"
        ScrollViewer.HorizontalScrollBarVisibility="Disabled"
        HorizontalContentAlignment="Center">
        <ListView.ItemsPanel>
            <ItemsPanelTemplate>
                <WrapPanel Orientation="Horizontal"
HorizontalAlignment="Center" />
            </ItemsPanelTemplate>
        </ListView.ItemsPanel>

        <ListView.ItemTemplate>
            <DataTemplate>
                <Grid Margin="10" Width="450" Height="600"
Background="LightGray">
                    <Grid.RowDefinitions>
                        <RowDefinition Height="Auto" />
                        <RowDefinition Height="450" />
                        <RowDefinition Height="Auto" />
                    </Grid.RowDefinitions>

                    <!-- ФИО пользователя -->
                    <TextBlock VerticalAlignment="Center"
TextAlignment="Center"
                                Width="450" Grid.Row="0"
TextWrapping="Wrap"
                                Text="{Binding FIO}"
HorizontalAlignment="Center"
                                FontSize="18" FontWeight="Bold" />

                    <!-- Изображение пользователя -->
                    <Image Width="450" Grid.Row="1"
Stretch="UniformToFill" HorizontalAlignment="Center">
                        <Image.Source>
                            <Binding Path="Photo" >
                                <Binding.TargetNullValue>

<ImageSource>/Resources/DefaultPhoto.png</ImageSource>
                                </Binding.TargetNullValue>
                            </Binding>
                        </Image.Source>
                    </Image>

                    <!-- Роль пользователя -->
                    <TextBlock VerticalAlignment="Center"
TextAlignment="Left"
                                Width="450" Grid.Row="2"
HorizontalAlignment="Center"
                                FontSize="18" Text="{Binding Role}" />
                </Grid>
            </DataTemplate>
        </ListView.ItemTemplate>
    </ListView>
</Grid>
</Page>

```

Листинг UserPage.xaml.cs

```
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 WpfApp1.pages
{
    /// <summary>
    /// Логика взаимодействия для UserPage.xaml
    /// </summary>
    public partial class UserPage : Page
    {
        public UserPage()
        {
            InitializeComponent();
            CmbSorting.SelectedIndex = 0; // Установка сортировки по
            умолчанию
            CheckDriver.IsChecked = false; // Сбрасываем фильтр
            UpdateUsers(); // Начальная загрузка данных
        }

        // Метод для обновления данных
        private void UpdateUsers()
        {
            // Загружаем всех пользователей
            var currentUsers = Entities.GetContext().User.ToList();

            // Поиск по Ф.И.О. (без учета регистра)
            if (!string.IsNullOrEmpty(TextBoxSearch.Text))
            {
                currentUsers = currentUsers.Where(x => x.FIO.ToLower().Contains(TextBoxSearch.Text.ToLower())).ToList();
            }

            // Фильтрация: показываем только пользователей
            if (CheckDriver.IsChecked == true)
            {
                currentUsers = currentUsers.Where(x =>
                x.Role.Contains("Пользователь")).ToList();
            }

            // Сортировка
            if (CmbSorting.SelectedIndex == 0) // По возрастанию
            {
                currentUsers = currentUsers.OrderBy(x => x.FIO).ToList();
            }
            else // По убыванию
            {
                currentUsers = currentUsers.OrderByDescending(x =>
                x.FIO).ToList();
            }
        }
    }
}
```

```

        // Устанавливаем данные в ListView
        ListUser.ItemsSource = currentUsers;
    }

    // Обработчик поиска
    private void TextBoxSearch_TextChanged(object sender,
TextChangedEventArgs e)
    {
        UpdateUsers();
    }

    // Обработчик изменения сортировки
    private void CmbSorting_SelectionChanged(object sender,
SelectionChangedEventArgs e)
    {
        UpdateUsers();
    }

    // Обработчик фильтрации
    private void CheckDriver_Checked(object sender, RoutedEventArgs e)
    {
        UpdateUsers();
    }

    private void CheckDriver_Unchecked(object sender, RoutedEventArgs e)
    {
        UpdateUsers();
    }

    // Обработчик кнопки "Очистить фильтр"
    private void ButtonClear_Click(object sender, RoutedEventArgs e)
    {
        TextBoxSearch.Clear();
        CmbSorting.SelectedIndex = 0;
        CheckDriver.IsChecked = false;
        UpdateUsers();
    }
}

```

Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)
Колледж информатики и программирования

ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 2

**Тема: Разработка графического приложения для формирования
графиков и отчетов**

Студент: Сакович Н.Д.

Группа: 4ИСИП-421

Преподаватель: Аксёнова Т.Г.

Дата: 19.12.2024

Цель работы: приобрести практические навыки реализации диаграмм, формирования текстовых документов Microsoft Word и электронных таблиц Microsoft Excel в приложениях WPF.

Скрипт создания БД:

```
-- Создание базы данных
CREATE DATABASE RepairRequestsDB;
GO

USE RepairRequestsDB;
GO

-- Таблица Users (Пользователи)
CREATE TABLE Users (
    ID INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор пользователя
    FullName NVARCHAR(150) NOT NULL, -- ФИО пользователя
    Role NVARCHAR(50) NOT NULL -- Роль: Администратор, Менеджер, Исполнитель
);

-- Таблица Equipment (Оборудование)
CREATE TABLE Equipment (
    ID INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор оборудования
    Name NVARCHAR(100) NOT NULL, -- Название оборудования
    SerialNumber NVARCHAR(100) NOT NULL -- Серийный номер оборудования
);

-- Таблица Requests (Заявки)
CREATE TABLE Requests (
    ID INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор заявки
    RequestNumber NVARCHAR(50) NOT NULL, -- Номер заявки
    DateAdded DATETIME NOT NULL, -- Дата добавления заявки
    EquipmentID INT NOT NULL FOREIGN KEY REFERENCES Equipment(ID), -- ID
оборудования
    FaultType NVARCHAR(100) NOT NULL, -- Тип неисправности
    Description NVARCHAR(MAX), -- Описание проблемы
    ClientID INT NOT NULL FOREIGN KEY REFERENCES Users(ID), -- Клиент,
подающий заявку
    Status NVARCHAR(50) NOT NULL -- Статус заявки: "В ожидании", "В работе",
"Выполнено"
);

-- Таблица Repairs (Исполнения заявок)
CREATE TABLE Repairs (
    ID INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор ремонта
    RequestID INT NOT NULL FOREIGN KEY REFERENCES Requests(ID), -- Заявка на
ремонт
    ExecutorID INT NOT NULL FOREIGN KEY REFERENCES Users(ID), -- Исполнитель
    StartDate DATETIME NOT NULL, -- Дата начала ремонта
    EndDate DATETIME, -- Дата окончания ремонта
    Comments NVARCHAR(MAX) -- Комментарии исполнителя
);

-- Таблица Feedback (Отзывы)
CREATE TABLE Feedback (
    ID INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор отзыва
    RequestID INT NOT NULL FOREIGN KEY REFERENCES Requests(ID), -- Заявка
Rating INT NOT NULL CHECK (Rating >= 1 AND Rating <= 5), -- Оценка
работы
    FeedbackText NVARCHAR(MAX) -- Текст отзыва
);
```

```

ALTER TABLE Users ADD Login NVARCHAR(100) NOT NULL UNIQUE;
ALTER TABLE Users ADD PasswordHash NVARCHAR(256) NOT NULL;

-- Заполнение таблицы Users
INSERT INTO Users (FullName, Role, Login, PasswordHash)
VALUES
('Администратор Иванов', 'Администратор', 'admin',
'40BD001563085FC35165329EA1FF5C5ECBDBBEEF'), -- пароль: 123
('Менеджер Петров', 'Менеджер', 'manager',
'40BD001563085FC35165329EA1FF5C5ECBDBBEEF'), -- пароль: admin
('Исполнитель Сидоров', 'Исполнитель', 'executor',
'A94A8FE5CCB19BA61C4C0873D391E987982FBBD3'), -- пароль: test
('Клиент Алексей Алексеев', 'Клиент', 'alexei',
'E99A18C428CB38D5F260853678922E03AB869F48'), -- пароль: password
('Клиент Мария Марьянова', 'Клиент', 'maria',
'B1B3773A05C0ED0176787A4F1574FF0075F7521E'); -- пароль: welcome

-- Заполнение таблицы Equipment
INSERT INTO Equipment (Name, SerialNumber)
VALUES
('Ноутбук ASUS', 'ASUS12345'),
('Принтер Canon', 'CANON56789'),
('Смартфон Samsung', 'SAMSUNG11223'),
('Сервер HP', 'HP445566'),
('Кофемашина Bosch', 'BOSCH99887');

-- Заполнение таблицы Requests
INSERT INTO Requests (RequestNumber, DateAdded, EquipmentID, FaultType,
Description, ClientID, Status)
VALUES
('REQ-001', GETDATE(), 1, 'Не включается', 'Ноутбук не реагирует на кнопку
питания', 4, 'В ожидании'),
('REQ-002', GETDATE(), 2, 'Не печатает', 'Принтер не выводит документы на
печать', 5, 'В работе'),
('REQ-003', GETDATE(), 3, 'Не заряжается', 'Смартфон не заряжается от
зарядного устройства', 4, 'В ожидании'),
('REQ-004', GETDATE(), 4, 'Перезагрузки', 'Сервер постоянно
перезагружается', 5, 'В работе'),
('REQ-005', GETDATE(), 5, 'Шумит', 'Кофемашина издает громкий шум при
работе', 4, 'Выполнено');

```

На рисунках 12-15 представлено создание БД, ее сущности и их наполнение.

DESKTOP-S4HTB40\...nt - dbo.Category			
	Имя столбца	Тип данных	Разрешить знач...
PK	ID	int	<input type="checkbox"/>
	Name	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Рисунок 12. Атрибуты сущности Category

DESKTOP-S4HTB4O\...yment - dbo.User			
	Имя столбца	Тип данных	Разрешить знач...
▶	ID	int	<input type="checkbox"/>
	FIO	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Рисунок 13. Атрибуты сущности User

DESKTOP-S4HTB4O\...nt - dbo.Payment			
	Имя столбца	Тип данных	Разрешить знач...
▶	ID	int	<input type="checkbox"/>
	CategoryID	int	<input type="checkbox"/>
	UserID	int	<input type="checkbox"/>
	Date	date	<input type="checkbox"/>
	Name	nvarchar(50)	<input type="checkbox"/>
	Num	int	<input type="checkbox"/>
	Price	decimal(10, 2)	<input type="checkbox"/>
			<input type="checkbox"/>

Рисунок 14. Атрибуты сущности Payment

DESKTOP-S4HTB4O\...nt - dbo.Category		
	ID	Name
▶	1	Транспортные Расходы
	2	Коммунальные услуги
	3	Развлечения
	4	Продукты
	5	Одежда
	6	Здоровье
	7	Образование
	8	Связь
	9	Подарки
	10	Прочие расходы
*	NULL	NULL

Рисунок 15. Наполнение сущности Category

Вид страницы PaymentProject на рисунке 16. На рисунках 17 показан результат экспорта информации в Excel, Word.

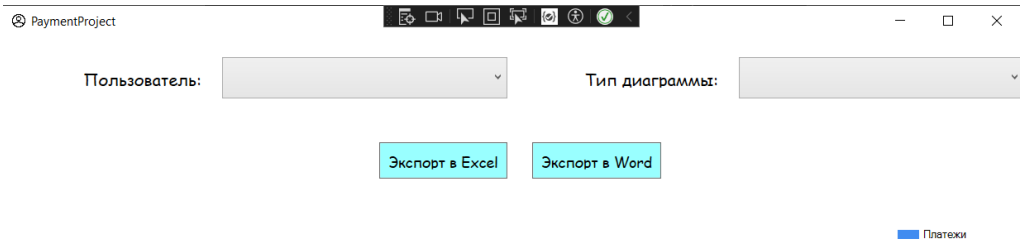


Рисунок 16. Вид страницы PaymentProject

Дата платежа	Название	Стоимость	Количество	Сумма
Здоровье				
30.10.2023	Врач	300,00	2	600,00
ИТОГО:				600,00
Транспортные Расходы				
26.10.2023	Поезд	50,00	1	50,00
ИТОГО:				50,00
Общий итог:				650,00

Рисунок 17. Экспорт в Excel

Листинг к странице представлен ниже, также словарь стилей.
PaymentProject.xaml.cs:

```
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;
using System.Windows.Forms.DataVisualization.Charting;
using Word = Microsoft.Office.Interop.Word;
using Excel = Microsoft.Office.Interop.Excel;

namespace Yp13_15TRPO_DavMav
{
    public partial class MainWindow : Window
    {
        private Entities _context = new Entities();
        public MainWindow()
        {
            InitializeComponent();
            ChartPayments.ChartAreas.Add(new ChartArea("Main"));
            var currentSeries = new Series("Платежи")
            {
                IsValueShownAsLabel = true
            };
            ChartPayments.Series.Add(currentSeries);
            Usercmd.ItemsSource = _context.User.ToList(); //ФИО
пользователей
            Typecmd.ItemsSource = Enum.GetValues(typeof(SeriesChartType));
        }

        private void Window_Closing(object sender,
System.ComponentModel.CancelEventArgs e)
        {
            MessageBoxResult result = MessageBox.Show("Вы уверены что хотите
закрыть приложение?", "Подтверждение закрытия", MessageBoxButton.YesNo,
MessageBoxImage.Question);
            if (result == MessageBoxResult.No)
                e.Cancel = true;
        }

        private void Excel_Click(object sender, RoutedEventArgs e)
        {
            var allUsers = _context.User.ToList().OrderBy(u =>
u.FIO).ToList();
            var application = new Excel.Application();
            application.SheetsInNewWorkbook = allUsers.Count();
            Excel.Workbook workbook =
application.Workbooks.Add(Type.Missing);

            for (int i = 0; i < allUsers.Count(); i++)
            {
                int startRowIndex = 1;
                Excel.Worksheet worksheet = application.Worksheets.Item[i +
1];

                worksheet.Name = allUsers[i].FIO;
                worksheet.Cells[1][startRowIndex] = "Дата платежа";
                worksheet.Cells[2][startRowIndex] = "Название";
                worksheet.Cells[3][startRowIndex] = "Стоимость";
                worksheet.Cells[4][startRowIndex] = "Количество";
                worksheet.Cells[5][startRowIndex] = "Сумма";
                Excel.Range columlHeaderRange =
worksheet.Range[worksheet.Cells[1][1], worksheet.Cells[5][1]];

```

```

        columnlHeaderRange.HorizontalAlignment =
Excel.XlHAlign.xlHAlignCenter;
        columnlHeaderRange.Font.Bold = true;
        startRowIndex++;
        var userCategories = allUsers[i].Payment.OrderBy(u =>
u.Date).GroupBy(u => u.Category).OrderBy(u => u.Key.Name);

        decimal totalSum = 0; // Переменная для хранения общей суммы
платежей

        foreach (var groupCategory in userCategories)
        {
            Excel.Range headerRange =
worksheet.Range[worksheet.Cells[1][startRowIndex],
worksheet.Cells[5][startRowIndex]];
            headerRange.Merge();
            headerRange.Value = groupCategory.Key.Name;
            headerRange.HorizontalAlignment =
Excel.XlHAlign.xlHAlignCenter;
            headerRange.Font.Italic = true;
            startRowIndex++;
            foreach (var payment in groupCategory)
            {
                worksheet.Cells[1][startRowIndex] =
payment.Date.ToString("dd.MM.yyyy");
                worksheet.Cells[2][startRowIndex] = payment.Name;
                worksheet.Cells[3][startRowIndex] = payment.Price;
                (worksheet.Cells[3][startRowIndex] as
Excel.Range).NumberFormat = "0.00";
                worksheet.Cells[4][startRowIndex] = payment.Num;
                worksheet.Cells[5][startRowIndex].Formula =
$"=C{startRowIndex}*D{startRowIndex}";
                (worksheet.Cells[5][startRowIndex] as
Excel.Range).NumberFormat = "0.00";
                totalSum += payment.Price * payment.Num; //
Обновляем общую сумму

                startRowIndex++;
            }
            Excel.Range sumRange =
worksheet.Range[worksheet.Cells[1][startRowIndex],
worksheet.Cells[4][startRowIndex]];
            sumRange.Merge();
            sumRange.Value = "ИТОГО:";
            sumRange.HorizontalAlignment =
Excel.XlHAlign.xlHAlignRight;
            worksheet.Cells[5][startRowIndex].Formula =
$"=SUM(E{startRowIndex - groupCategory.Count()}:" + $"E{startRowIndex -
1}))";
            sumRange.Font.Bold =
worksheet.Cells[5][startRowIndex].Font.Bold = true;
            startRowIndex++;
            Excel.Range rangeBorders =
worksheet.Range[worksheet.Cells[1][1], worksheet.Cells[5][startRowIndex -
1]];

            rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle =

            rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle =

            rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle =

            rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle =

            rangeBorders.Borders[Excel.XlBordersIndex.xlInsideHorizontal].LineStyle =

```

```

rangeBorders.Borders[Excel.XlBordersIndex.xlInsideVertical].LineStyle =
Excel.XlLineStyle.xlContinuous;
        worksheet.Columns.AutoFit();
    }

    // Добавление строки "Общий итог"
    Excel.Range totalRowRange =
worksheet.Range[worksheet.Cells[1][startRowIndex],
worksheet.Cells[4][startRowIndex]];
        totalRowRange.Merge();
        totalRowRange.Value = "Общий итог:";
        totalRowRange.HorizontalAlignment =
Excel.XlHAlign.xlHAlignRight;
        totalRowRange.Font.Bold = true;
        totalRowRange.Font.Color =
System.Drawing.ColorTranslator.ToOle(System.Drawing.Color.Red); // Красный
цвет
        worksheet.Cells[5][startRowIndex].Font.Color =
System.Drawing.ColorTranslator.ToOle(System.Drawing.Color.Red); // Красный
цвет
        worksheet.Cells[5][startRowIndex].Value = totalSum; //
Заполнение общей суммы
        (worksheet.Cells[5][startRowIndex] as
Excel.Range).NumberFormat = "0.00";

        // Оформление границ для общей строки
        Excel.Range totalRowBorders =
worksheet.Range[worksheet.Cells[1][startRowIndex],
worksheet.Cells[5][startRowIndex]];

totalRowBorders.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle =
Excel.XlLineStyle.xlContinuous;

        application.Visible = true;
    }
}
private void Word_Click(object sender, RoutedEventArgs e)
{
    var allUsers = _context.User.ToList();
    var allCategories = _context.Category.ToList();
    var application = new Word.Application();
    Word.Document document = application.Documents.Add();

    // Добавление верхнего колонтитула с текущей датой
    foreach (Word.Section section in document.Sections)
    {
        Word.HeaderFooter header =
section.Headers[Word.WdHeaderFooterIndex.wdHeaderFooterPrimary];
        header.Range.Text = $"Дата:
{DateTime.Now.ToString("dd.MM.yyyy")}";
        header.Range.ParagraphFormat.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphRight; // Выравнивание по правому
краю
    }

    foreach (var user in allUsers)
    {
        Word.Paragraph userParagraph = document.Paragraphs.Add();
        Word.Range userRange = userParagraph.Range;
        userRange.Text = user.FIO;
        userParagraph.set_Style("Заголовок");
        userRange.ParagraphFormat.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphCenter;
    }
}

```

```

        userRange.InsertParagraphAfter();
        document.Paragraphs.Add(); // Пустая строка
        Word.Paragraph tableParagraph = document.Paragraphs.Add();
        Word.Range tableRange = tableParagraph.Range;
        Word.Table paymentsTable = document.Tables.Add(tableRange,
allCategories.Count() + 1, 2);
        paymentsTable.Borders.InsideLineStyle =
paymentsTable.Borders.OutsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;
        paymentsTable.Range.Cells.VerticalAlignment =
Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

        Word.Range cellRange;

        cellRange = paymentsTable.Cell(1, 1).Range;
        cellRange.Text = "Категория";
        cellRange = paymentsTable.Cell(1, 2).Range;
        cellRange.Text = "Сумма расходов";

        paymentsTable.Rows[1].Range.Font.Name = "Times New Roman";
        paymentsTable.Rows[1].Range.Font.Size = 14;
        paymentsTable.Rows[1].Range.Bold = 1;
        paymentsTable.Rows[1].Range.ParagraphFormat.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphCenter;

        for (int i = 0; i < allCategories.Count(); i++)
        {
            var currentCategory = allCategories[i];
            cellRange = paymentsTable.Cell(i + 2, 1).Range;
            cellRange.Text = currentCategory.Name;
            cellRange.Font.Name = "Times New Roman";
            cellRange.Font.Size = 12;

            cellRange = paymentsTable.Cell(i + 2, 2).Range;
            cellRange.Text = user.Payment.ToList().
                Where(u => u.Category == currentCategory).Sum(u =>
u.Num * u.Price).ToString("N2") + " руб.";
            cellRange.Font.Name = "Times New Roman";
            cellRange.Font.Size = 12;
        } // завершение цикла по строкам таблицы
        document.Paragraphs.Add(); // пустая строка
        Payment maxPayment = user.Payment.OrderByDescending(u =>
u.Price * u.Num).FirstOrDefault();
        if (maxPayment != null)
        {
            Word.Paragraph maxPaymentParagraph =
document.Paragraphs.Add();
            Word.Range maxPaymentRange = maxPaymentParagraph.Range;
            maxPaymentRange.Text = $"Самый дорогостоящий платеж -
{maxPayment.Name} за {(maxPayment.Price * maxPayment.Num).ToString("N2")} "
+ $"руб. от {maxPayment.Date.ToString("dd.MM.yyyy")}";
            maxPaymentParagraph.setStyle("Подзаголовок");
            maxPaymentRange.Font.Color =
Word.WdColor.wdColorDarkRed;
            maxPaymentRange.InsertParagraphAfter();
        }
        document.Paragraphs.Add(); // пустая строка

        Payment minPayment = user.Payment.OrderBy(u => u.Price *
u.Num).FirstOrDefault();
        if (minPayment != null)
        {
            Word.Paragraph minPaymentParagraph =
document.Paragraphs.Add();
            Word.Range minPaymentRange = minPaymentParagraph.Range;

```

```

        minPaymentRange.Text = $"Самый дешевый платеж -
{minPayment.Name} за {(minPayment.Price * minPayment.Num).ToString("N2")} "
+ $"руб. от {minPayment.Date.ToString("dd.MM.yyyy")}";
        minPaymentParagraph.setStyle("Подзаголовок");
        minPaymentRange.Font.Color =
Word.WdColor.wdColorDarkGreen;
        minPaymentRange.InsertParagraphAfter();
    }
    if (user != allUsers.LastOrDefault())
document.Words.Last.InsertBreak(Word.WdBreakType.wdPageBreak);
    } // завершение цикла по пользователям

    // Добавление нижнего колонтитула с номером страницы
    foreach (Word.Section section in document.Sections)
    {
        Word.HeaderFooter footer =
section.Footers[Word.WdHeaderFooterIndex.wdHeaderFooterPrimary];
        footer.Range.Text = "Страница ";
        footer.Range.Fields.Add(footer.Range,
Word.WdFieldType.wdFieldPage); // Добавление поля для номера страницы
        footer.Range.ParagraphFormat.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphCenter; // Выравнивание по центру
    }

    application.Visible = true;
    document.SaveAs2(@"C:\Users\vlad5\Downloads\Payments.docx");
    document.SaveAs2(@"C:\Users\vlad5\Downloads\Payments.pdf",
Word.WdExportFormat.wdExportFormatPDF);
}
private void UpdateChart(object sender, SelectionChangedEventArgs e)
{
    if (Usercmd.SelectedItem is User currentUser &&
Typecmd.SelectedItem is SeriesChartType currentType)
    {
        Series currentSeries =
ChartPayments.Series.FirstOrDefault();
        currentSeries.ChartType = currentType;
        currentSeries.Points.Clear();

        var categoriesList = _context.Category.ToList();
        foreach (var category in categoriesList)
        {
            currentSeries.Points.AddXY(category.Name,
currentUser
                _context.Payment.ToList().Where(u => u.User ==
u.Num));
                && u.Category == category).Sum(u => u.Price *
        }
    }
}
}
}
}

```

PaymentProject.xaml:

```

<Window x:Class="Yp13_15TRPO_DavMav.MainWindow"
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:Yp13_15TRPO_DavMav"

```

```

        xmlns:charts="clr-
namespace:System.Windows.Forms.DataVisualization.Charting;assembly=System.Wi
ndows.Forms.DataVisualization"
        mc:Ignorable="d"
        Title="PaymentProject" Height="700" Width="1000"
Closing="Window_Closing">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="160"/>
            <RowDefinition/>
        </Grid.RowDefinitions>
        <StackPanel Grid.Row="0">
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="80"/>
                    <RowDefinition Height="80"/>
                </Grid.RowDefinitions>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="200"/>
                    <ColumnDefinition Width="300"/>
                    <ColumnDefinition Width="200"/>
                    <ColumnDefinition Width="300"/>
                </Grid.ColumnDefinitions>
                <Label Content="Пользователь:" Grid.Row="0"
HorizontalAlignment="Right" VerticalAlignment="Center" Grid.Column="0"/>
                <ComboBox x:Name="Usercmd" DisplayMemberPath="FIO"
SelectionChanged="UpdateChart" Grid.Row="0" Grid.Column="1"
HorizontalAlignment="Stretch"/>
                <Label Content="Тип диаграммы:" Grid.Row="0"
HorizontalAlignment="Right" VerticalAlignment="Center" Grid.Column="2"/>
                <ComboBox x:Name="Typecmd" Grid.Row="0"
SelectionChanged="UpdateChart" Grid.Column="3"
HorizontalAlignment="Stretch"/>
                <Button x:Name="Excelbt" Click="Excel_Click" Grid.Column="1"
Grid.Row="1" HorizontalAlignment="Right" VerticalAlignment="Center"
FontSize="14" Content="Экспорт в Excel"/>
                <Button x:Name="Wordbt" Click="Word_Click" Grid.Column="2"
Grid.Row="1" HorizontalAlignment="Left" VerticalAlignment="Center"
FontSize="14" Content="Экспорт в Word"/>
            </Grid>
        </StackPanel>
        <WindowsFormsHost Grid.Row="1" Margin="5">
            <charts:Chart x:Name="ChartPayments">
                <charts:Chart.Legends>
                    <charts:Legend>

                </charts:Legend>
            </charts:Chart.Legends>
        </charts:Chart>
    </WindowsFormsHost>
</Grid>
</Window>

```

Dictionary.xaml:

```

<ResourceDictionary
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

    <!-- Основные цвета -->
    <Color x:Key="PrimaryColor">#B5D5CA</Color>
    <!-- Основной цвет 1 -->
    <Color x:Key="SecondaryColor">#E0A9AF</Color>
    <!-- Основной цвет 2 -->

```



```

<!-- Дополнительные цвета -->
<Color x:Key="AccentColor1">#D1EEFC</Color>
<Color x:Key="AccentColor2">#FFFC6D</Color>

<!-- Цвета для текста -->
<Color x:Key="TextWhite">#FFFFFF</Color>
<Color x:Key="TextBlack">#000000</Color>

<!-- Кисти на основе цветов -->
<SolidColorBrush x:Key="PrimaryBrush" Color="{StaticResource
PrimaryColor}"/>
<SolidColorBrush x:Key="SecondaryBrush" Color="{StaticResource
SecondaryColor}"/>
<SolidColorBrush x:Key="AccentBrush1" Color="{StaticResource
AccentColor1}"/>
<SolidColorBrush x:Key="AccentBrush2" Color="{StaticResource
AccentColor2}"/>
<SolidColorBrush x:Key="TextWhiteBrush" Color="{StaticResource
TextWhite}"/>
<SolidColorBrush x:Key="TextBlackBrush" Color="{StaticResource
TextBlack}"/>

<!-- Стил ь для кнопок -->
<Style TargetType="Button">
    <Setter Property="Height" Value="45"/>
    <Setter Property="Padding" Value="8"/>
    <Setter Property="Margin" Value="12"/>
    <Setter Property="FontFamily" Value="Tahoma"/>
    <Setter Property="FontSize" Value="12pt"/>
    <Setter Property="Background" Value="{StaticResource
PrimaryBrush}"/>
    <Setter Property="Foreground" Value="{StaticResource
TextBlackBrush}"/>
    <Setter Property="BorderBrush" Value="{StaticResource
SecondaryBrush}"/>
    <Setter Property="BorderThickness" Value="2"/>
</Style>

<!-- Стил ь для TextBlock -->
<Style TargetType="TextBlock">
    <Setter Property="Foreground" Value="{StaticResource
TextBlackBrush}"/>
    <Setter Property="FontFamily" Value="Tahoma"/>
    <Setter Property="FontSize" Value="14pt"/>
</Style>

<!-- Стил ь для фона окна -->
<Style TargetType="Window">
    <Setter Property="Background" Value="{StaticResource
AccentBrush2}"/>
</Style>

<Style TargetType="ComboBox">
    <Setter Property="FontFamily" Value="Tahoma"/>
    <Setter Property="FontSize" Value="12"/>
    <Setter Property="Padding" Value="5"/>
    <Setter Property="Margin" Value="10"/>
    <Setter Property="Background" Value="LightGray"/>
    <Setter Property="Foreground" Value="Black"/>
</Style>

</ResourceDictionary>

```