  
Рисунок 1 – окно входа

С помощью окна входа вы можете перейти на окно регистрации, или же залогиниться.

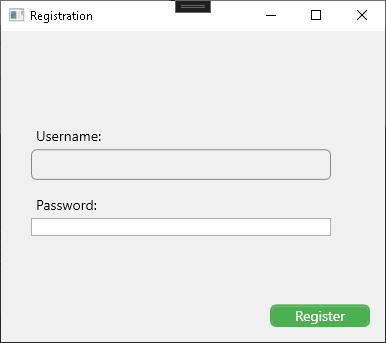


Рисунок 2 – окно регистрации

Здесь проходит регистрация нового пользователя, при регистрации нового пользователя, пользователь автоматически получает роль Users.



Рисунок 3 – окно User

Таким образом выглядит окно User, users не имеет доступа, редактировать и каким либо образом видоизменять информацию

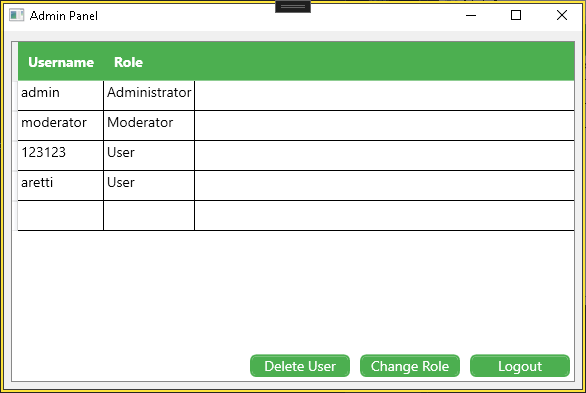


Рисунок 4 – окно Administrator

Пользователь окна Administrator имеет права доступа изменять роли роли удалять пользователей.



Рисунок 5 – окно Moderatora

Moderator имеет доступ изменять удалять и добавлять информацию в таблицу.

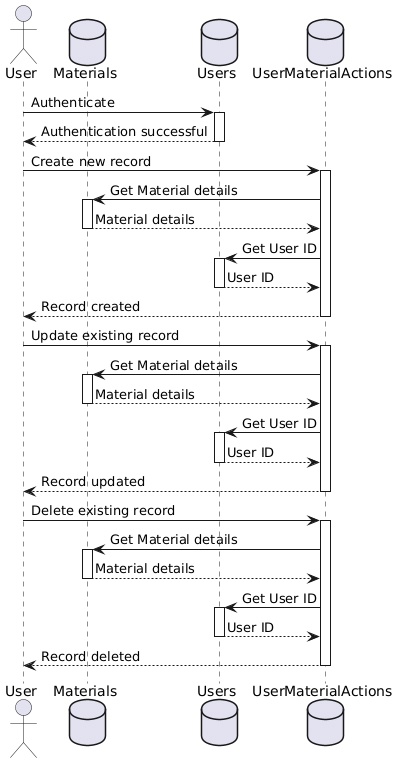


диаграмма систем управления материалами

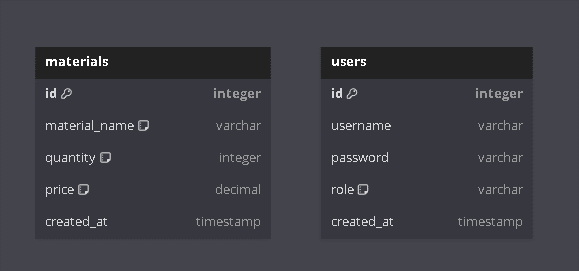
В данной блок-схеме представлена архитектура класса AdminWindow, который реализует административный интерфейс для управления пользователями в приложении. Основной класс отображает панель администратора, содержащую элементы управления и логику взаимодействия с пользовательскими данными.

В вершине схемы находится класс AdminWindow, который инициализирует свои компоненты и загружает пользователей из базы данных при создании окна. Первым этапом является метод LoadUsers, который отвечает за извлечение списка пользователей из базы данных и установку его в качестве источника данных для элемента DataGrid. В результате пользователи отображаются в таблице для дальнейших операций.

Другим важным компонентом является DataGrid, который заполняется списком пользователей и позволяет осуществлять выбор пользователя. Это поведение обрабатывается с помощью метода SelectionChanged, который отслеживает изменения в выборе пользователя.

Основные действия, связанные с управлением пользователями, реализуются через кнопку-обработчики событий: кнопка удаления пользователя (DeleteUserButton) удаляет выбранного пользователя из базы данных и обновляет таблицу, кнопка изменения роли пользователя (ChangeRoleButton) позволяет изменить роль выбранного пользователя. При этом открывается диалоговое окно для ввода новой роли, и если ввод корректен, изменения сохраняются в базе данных. Кнопка выхода (LogoutButton) закрывает текущее окно и открывает окно логина, что завершает сессию администратора.

В целом, блок-схема отображает основные функциональные узлы и их связи в классе AdminWindow, подчеркивая логику взаимодействия между визуальными элементами и обработкой данных.

  
Диаграмма базы данных

В данной блок-схеме представлена структура проекта, связанного с разработкой приложения, разделенного на три ключевые компоненты: интерфейс, логика и база данных. В начале схемы обозначена основная цель — разработать приложение, что служит отправной точкой для всех дальнейших действий.

Первая ветка — интерфейс. На первом этапе осуществляется проектирование интерфейса, где разрабатывается его визуальное оформление и пользовательский опыт. Затем создаются интерактивные прототипы, которые позволяют тестировать и корректировать интерфейс на ранних этапах разработки. Завершает эту ветку этап тестирования интерфейса, который помогает выявить потенциальные проблемы и улучшить взаимодействие пользователя с приложением.

Вторая ветка — логика. На первом этапе разрабатываются алгоритмы, которые отвечают за обработку данных и взаимодействие между компонентами приложения. Второй этап включает в себя программирование кода, реализующего разработанные алгоритмы. Последним шагом является отладка, на которой устраняются ошибки и проверяется правильность работы логики приложения.

Третья ветка — база данных. Первое действие заключается в создании модели данных, описывающей, какие данные и в каком формате будут храниться. После моделирования данных создаются необходимые таблицы, которые будут хранить информацию. Завершает эту ветку этап определения связей между таблицами, что обеспечивает целостность данных и эффективный доступ к ним.

Этот структурированный подход к разработке приложения помогает четко распределять задачи между командами и следовать логическому маршруту от идеи к конечному продукту. Блок-схема служит визуальным инструментом, упрощающим понимание задач и их взаимосвязей в рамках проекта.

## **Приложение A**

## **Текст программы**

**AdminWindow**  
  
<Window x:Class="MaterialsViewer.AdminWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Admin Panel" Height="400" Width="600" Background="#f0f0f0" WindowStartupLocation="CenterScreen">

<Grid>

<DataGrid Name="UsersDataGrid" AutoGenerateColumns="False" Margin="10"

SelectionChanged="UsersDataGrid\_SelectionChanged">

<DataGrid.Columns>

<DataGridTextColumn Header="Id" Binding="{Binding Id}" Visibility="Hidden"/>

<DataGridTextColumn Header="Username" Binding="{Binding Username}" />

<DataGridTextColumn Header="Role" Binding="{Binding Role}" />

</DataGrid.Columns>

</DataGrid>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right" VerticalAlignment="Bottom" Margin="10">

<Button Content="Delete User" Click="DeleteUserButton\_Click" Width="100" Margin="5"/>

<Button Content="Change Role" Click="ChangeRoleButton\_Click" Width="100" Margin="5"/>

<Button Content="Logout" Click="LogoutButton\_Click" Width="100" Margin="5"/>

</StackPanel>

</Grid>

</Window>

**AdminWindow.xaml.cs**using System.Linq;

using System.Windows;

using System.Windows.Controls;

using MaterialsViewer.Data;

using MaterialsViewer.Models;

namespace MaterialsViewer

{

public partial class AdminWindow : Window

{

public AdminWindow()

{

InitializeComponent();

LoadUsers();

}

private void LoadUsers()

{

using (var context = new AppDbContext())

{

var users = context.Users.ToList(); // Получаем список пользователей

UsersDataGrid.ItemsSource = users; // Устанавливаем источник данных

}

}

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

{

if (UsersDataGrid.SelectedItem is User selectedUser)

{

// Здесь можно обрабатывать выбранного пользователя

}

}

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

{

if (UsersDataGrid.SelectedItem is User selectedUser)

{

using (var context = new AppDbContext())

{

context.Users.Remove(selectedUser);

context.SaveChanges();

}

LoadUsers(); // Обновляем список пользователей

}

}

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

{

if (UsersDataGrid.SelectedItem is User selectedUser)

{

string newRole = Microsoft.VisualBasic.Interaction.InputBox("Enter new role for user:", "Change Role", selectedUser.Role);

if (!string.IsNullOrEmpty(newRole))

{

using (var context = new AppDbContext())

{

selectedUser.Role = newRole;

context.Users.Update(selectedUser);

context.SaveChanges();

}

LoadUsers(); // Обновляем список пользователей

}

}

}

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

{

LoginWindow loginWindow = new LoginWindow();

loginWindow.Show();

this.Close(); // Закрывает текущее окно

}

}

}

**App**<Application x:Class="MaterialsViewer.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

StartupUri="LoginWindow.xaml">

<Application.Resources>

<Style TargetType="Button">

<Setter Property="Background" Value="#4CAF50"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Padding" Value="10,5"/>

<Setter Property="Margin" Value="5"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="BorderBrush" Value="#4CAF50"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Cursor" Value="Hand"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}"

BorderBrush="{TemplateBinding BorderBrush}"

BorderThickness="{TemplateBinding BorderThickness}"

CornerRadius="5">

<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style TargetType="TextBox">

<Setter Property="Margin" Value="5"/>

<Setter Property="Padding" Value="10,5"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="BorderBrush" Value="#888888"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="Background" Value="White"/>

<Setter Property="Foreground" Value="Black"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="TextBox">

<Border BorderBrush="{TemplateBinding BorderBrush}"

BorderThickness="{TemplateBinding BorderThickness}"

CornerRadius="5">

<ScrollViewer x:Name="PART\_ContentHost"/>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<!-- Остальные стили и свойства -->

<Style TargetType="DataGrid">

<Setter Property="Background" Value="#ffffff"/>

<Setter Property="AutoGenerateColumns" Value="False"/>

<Setter Property="BorderBrush" Value="#888888"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="RowHeight" Value="30"/>

<Setter Property="SelectionMode" Value="Single"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="DataGridColumnHeader">

<Setter Property="Background" Value="#4CAF50"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Padding" Value="10"/>

</Style>

</Application.Resources>

</Application>  
  
**App.xaml.cs**

using System.Configuration;

using System.Data;

using System.Windows;

namespace Kon

{

/// <summary>

/// Interaction logic for App.xaml

/// </summary>

public partial class App : Application

{

}

}

**AppDbContext.cs**

using Microsoft.EntityFrameworkCore;

using MaterialsViewer.Models;

namespace MaterialsViewer.Data

{

public class AppDbContext : DbContext

{

public DbSet<Material> Materials { get; set; }

public DbSet<User> Users { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer("Server=localhost;Database=KURSACH;Integrated Security=True;TrustServerCertificate=True;");

}

}

}

**AssemblyInfo.cs**

using System.Windows;

[assembly: ThemeInfo(

ResourceDictionaryLocation.None, //where theme specific resource dictionaries are located

//(used if a resource is not found in the page,

// or application resource dictionaries)

ResourceDictionaryLocation.SourceAssembly //where the generic resource dictionary is located

//(used if a resource is not found in the page,

// app, or any theme specific resource dictionaries)

)]  
  
  
**LoginWindow**<Window x:Class="MaterialsViewer.LoginWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Login" Height="300" Width="400" Background="#f0f0f0" WindowStartupLocation="CenterScreen">

<Grid>

<StackPanel VerticalAlignment="Center">

<Label Content="Username:" HorizontalAlignment="Left" Margin="30,10,0,0" FontSize="14"/>

<TextBox Name="UsernameTextBox" HorizontalAlignment="Left" Width="300" Margin="30,0"/>

<Label Content="Password:" HorizontalAlignment="Left" Margin="30,10,0,0" FontSize="14"/>

<PasswordBox Name="PasswordBox" HorizontalAlignment="Left" Width="300" Margin="30,0"/>

<TextBlock Name="MessageTextBlock" Foreground="Red" HorizontalAlignment="Left" Margin="30,10,0,0" />

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right" VerticalAlignment="Bottom" Margin="10">

<Button Content="Login" Click="LoginButton\_Click" Width="100" Margin="5"/>

<Button Content="Register" Click="RegisterButton\_Click" Width="100" Margin="5"/>

</StackPanel>

</Grid>

</Window>

**LoginWindow.xaml.cs**using System.Linq;

using System.Windows;

using MaterialsViewer.Models;

using MaterialsViewer.Data;

namespace MaterialsViewer

{

public partial class LoginWindow : Window

{

public LoginWindow()

{

InitializeComponent();

}

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

{

string username = UsernameTextBox.Text;

string password = PasswordBox.Password;

using (var context = new AppDbContext())

{

var user = context.Users.FirstOrDefault(u => u.Username == username && u.Password == password);

if (user != null)

{

MessageBox.Show($"Welcome, {user.Username} ({user.Role})", "Login Success", MessageBoxButton.OK, MessageBoxImage.Information);

if (user.Role == "Administrator")

{

AdminWindow adminWindow = new AdminWindow();

adminWindow.Show();

}

else if (user.Role == "Moderator")

{

ModeratorWindow moderatorWindow = new ModeratorWindow();

moderatorWindow.Show();

}

else if (user.Role == "User")

{

MainWindow mainWindow = new MainWindow(); // Основное окно для обычных пользователей

mainWindow.Show();

}

this.Close(); // Закрываем окно логина

}

}

}

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

{

RegistrationWindow registrationWindow = new RegistrationWindow();

registrationWindow.ShowDialog(); // Используйте ShowDialog для открытия модального окна

}

}

}  
 **MainWindow**

<Window x:Class="MaterialsViewer.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Materials Viewer" Height="400" Width="600" Background="#f0f0f0" WindowStartupLocation="CenterScreen">

<Grid>

<DataGrid Name="MaterialsDataGrid" AutoGenerateColumns="False" Margin="10">

<DataGrid.Columns>

<DataGridTextColumn Header="Id" Binding="{Binding Id}" Visibility="Hidden"/>

<DataGridTextColumn Header="Material Name" Binding="{Binding MaterialName}" />

<DataGridTextColumn Header="Quantity" Binding="{Binding Quantity}" />

<DataGridTextColumn Header="Price" Binding="{Binding Price}" />

</DataGrid.Columns>

</DataGrid>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right" VerticalAlignment="Bottom" Margin="10">

<Button Content="Logout" Click="LogoutButton\_Click" Width="100" Margin="5"/>

</StackPanel>

</Grid>

</Window>  
  
  
**MainWindow.xaml.cs**

using System.Linq;

using System.Windows;

using MaterialsViewer.Data;

namespace MaterialsViewer

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

LoadMaterials();

}

private void LoadMaterials()

{

using (var context = new AppDbContext())

{

var materials = context.Materials.ToList();

MaterialsDataGrid.ItemsSource = materials;

}

}

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

{

LoginWindow loginWindow = new LoginWindow();

loginWindow.Show();

this.Close(); // Закрыть текущее окно

}

}

}  
  
  
**Material.cs**

namespace MaterialsViewer.Models

{

public class Material

{

public int Id { get; set; }

public string MaterialName { get; set; }

public int Quantity { get; set; }

public decimal Price { get; set; }

}

}

**MaterialsViewer**<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>WinExe</OutputType>

<TargetFramework>net8.0-windows</TargetFramework>

<Nullable>enable</Nullable>

<ImplicitUsings>enable</ImplicitUsings>

<UseWPF>true</UseWPF>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.Data.SqlClient" Version="5.2.2" />

<PackageReference Include="Microsoft.EntityFrameworkCore" Version="9.0.0" />

<PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="9.0.0" />

<PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="9.0.0">

<PrivateAssets>all</PrivateAssets>

<IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>

</PackageReference>

</ItemGroup>

</Project>

**MaterialsViewer.csproj.user**<?xml version="1.0" encoding="utf-8"?>

<Project ToolsVersion="Current" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">

<PropertyGroup />

<ItemGroup>

<ApplicationDefinition Update="App.xaml">

<SubType>Designer</SubType>

</ApplicationDefinition>

</ItemGroup>

<ItemGroup>

<Compile Update="AdminWindow.xaml.cs">

<SubType>Code</SubType>

</Compile>

<Compile Update="LoginWindow.xaml.cs">

<SubType>Code</SubType>

</Compile>

<Compile Update="ModeratorWindow.xaml.cs">

<SubType>Code</SubType>

</Compile>

<Compile Update="RegistrationWindow.xaml.cs">

<SubType>Code</SubType>

</Compile>

</ItemGroup>

<ItemGroup>

<Page Update="AdminWindow.xaml">

<SubType>Designer</SubType>

</Page>

<Page Update="LoginWindow.xaml">

<SubType>Designer</SubType>

</Page>

<Page Update="MainWindow.xaml">

<SubType>Designer</SubType>

</Page>

<Page Update="ModeratorWindow.xaml">

<SubType>Designer</SubType>

</Page>

<Page Update="RegistrationWindow.xaml">

<SubType>Designer</SubType>

</Page>

</ItemGroup>

</Project>

**ModeratorWindow**<Window x:Class="MaterialsViewer.ModeratorWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Moderator Panel" Height="400" Width="600" Background="#f0f0f0" WindowStartupLocation="CenterScreen">

<Grid>

<DataGrid Name="MaterialsDataGrid" AutoGenerateColumns="False" Margin="10"

SelectionChanged="MaterialsDataGrid\_SelectionChanged">

<DataGrid.Columns>

<DataGridTextColumn Header="Id" Binding="{Binding Id}" Visibility="Hidden"/>

<DataGridTextColumn Header="Material Name" Binding="{Binding MaterialName}" />

<DataGridTextColumn Header="Quantity" Binding="{Binding Quantity}" />

<DataGridTextColumn Header="Price" Binding="{Binding Price}" />

</DataGrid.Columns>

</DataGrid>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right" VerticalAlignment="Bottom" Margin="10">

<Button Content="Update Material" Click="UpdateMaterialButton\_Click" Width="120" Margin="5"/>

<Button Content="Delete Material" Click="DeleteMaterialButton\_Click" Width="120" Margin="5"/>

<Button Content="Logout" Click="LogoutButton\_Click" Width="100" Margin="5"/>

</StackPanel>

</Grid>

</Window>

**ModeratorWindow.xaml.cs**  
  
using System.Linq;

using System.Windows;

using System.Windows.Controls;

using MaterialsViewer.Data;

using MaterialsViewer.Models;

namespace MaterialsViewer

{

public partial class ModeratorWindow : Window

{

public ModeratorWindow()

{

InitializeComponent();

LoadMaterials();

}

private void LoadMaterials()

{

using (var context = new AppDbContext())

{

var materials = context.Materials.ToList(); // Получаем список материалов

MaterialsDataGrid.ItemsSource = materials; // Устанавливаем источник данных

}

}

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

{

if (MaterialsDataGrid.SelectedItem is Material selectedMaterial)

{

// Обновление материала (логика обновления)

string newName = Microsoft.VisualBasic.Interaction.InputBox("Enter new name for material:", "Update Material", selectedMaterial.MaterialName);

string newQuantity = Microsoft.VisualBasic.Interaction.InputBox("Enter new quantity for material:", "Update Material", selectedMaterial.Quantity.ToString());

string newPrice = Microsoft.VisualBasic.Interaction.InputBox("Enter new price for material:", "Update Material", selectedMaterial.Price.ToString("F2"));

if (!string.IsNullOrWhiteSpace(newName) && int.TryParse(newQuantity, out int quantity) && decimal.TryParse(newPrice, out decimal price))

{

selectedMaterial.MaterialName = newName;

selectedMaterial.Quantity = quantity;

selectedMaterial.Price = price;

using (var context = new AppDbContext())

{

context.Materials.Update(selectedMaterial);

context.SaveChanges();

}

LoadMaterials(); // Обновляем список материалов

}

}

else

{

MessageBox.Show("Please select a material to update.", "Error", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

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

{

if (MaterialsDataGrid.SelectedItem is Material selectedMaterial)

{

using (var context = new AppDbContext())

{

context.Materials.Remove(selectedMaterial);

context.SaveChanges();

}

LoadMaterials(); // Обновляем список материалов

}

else

{

MessageBox.Show("Please select a material to delete.", "Error", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

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

{

// Вы можете оставить этот метод пустым, если он не нужен

}

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

{

LoginWindow loginWindow = new LoginWindow();

loginWindow.Show();

this.Close(); // Закрыть текущее окно

}

}

}

**RegistrationWindow**<Window x:Class="MaterialsViewer.RegistrationWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Registration" Height="350" Width="400" Background="#f0f0f0" WindowStartupLocation="CenterScreen">

<Grid>

<StackPanel VerticalAlignment="Center">

<Label Content="Username:" HorizontalAlignment="Left" Margin="30,10,0,0" FontSize="14"/>

<TextBox Name="UsernameTextBox" HorizontalAlignment="Left" Width="300" Margin="30,0"/>

<Label Content="Password:" HorizontalAlignment="Left" Margin="30,10,0,0" FontSize="14"/>

<PasswordBox Name="PasswordBox" HorizontalAlignment="Left" Width="300" Margin="30,0"/>

<TextBlock Name="MessageTextBlock" Foreground="Red" HorizontalAlignment="Left"

Margin="30,10,0,0"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right" VerticalAlignment="Bottom" Margin="10">

<Button Content="Register" Click="RegisterButton\_Click" Width="100" Margin="5"/>

</StackPanel>

</Grid>

</Window>  
  
  
**RegistrationWindow.xaml.cs**using System.Linq;

using System.Windows;

using MaterialsViewer.Models;

using MaterialsViewer.Data;

namespace MaterialsViewer

{

public partial class RegistrationWindow : Window

{

public RegistrationWindow()

{

InitializeComponent();

}

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

{

string username = UsernameTextBox.Text;

string password = PasswordBox.Password;

string role = "User";

using (var context = new AppDbContext())

{

// Проверка, существует ли пользователь с таким именем

if (context.Users.Any(u => u.Username == username))

{

MessageTextBlock.Text = "Username already exists!";

return;

}

// Создание нового пользователя

User newUser = new User

{

Username = username,

Password = password, // Важно: лучше использовать хэширование!

Role = role

};

context.Users.Add(newUser);

context.SaveChanges(); // Сохранить изменения в базе данных

MessageBox.Show("Registration successful!", "Success", MessageBoxButton.OK, MessageBoxImage.Information);

this.Close(); // Закрыть окно регистрации

}

}

}

}

**User.cs**

using System.ComponentModel.DataAnnotations;

namespace MaterialsViewer.Models

{

public class User

{

public int Id { get; set; }

[Required]

[StringLength(100)]

public string Username { get; set; }

[Required]

[StringLength(256)]

public string Password { get; set; }

[Required]

[StringLength(50)]

public string Role { get; set; }

}

}

**База данных sql**

USE KURSACH;

SELECT \* FROM Materials;

USE KURSACH;

CREATE TABLE Materials (

Id INT PRIMARY KEY IDENTITY(1,1),

MaterialName NVARCHAR(100),

Quantity INT,

Price DECIMAL(10, 2)

);

INSERT INTO Materials (MaterialName, Quantity, Price)

VALUES

('Asphalt Shingles', 100, 12.50),

('Metal Sheets', 50, 20.00),

('Tiles', 200, 15.75),

('Underlayment', 150, 0.50);

CREATE TABLE Users (

Id INT PRIMARY KEY IDENTITY(1,1),

Username NVARCHAR(100) UNIQUE,

Password NVARCHAR(256),

Role NVARCHAR(50)

);

USE KURSACH;

SELECT \* FROM Materials;

SELECT \* FROM Users;

-- Добавление администратора

INSERT INTO Users (Username, Password, Role)

VALUES ('admin', 'admin\_password', 'Administrator');

-- Добавление модератора

INSERT INTO Users (Username, Password, Role)

VALUES ('moderator', 'moderator\_password', 'Moderator');