Федеральное государственное образовательное бюджетное учреждение высшего образования

«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ

РОССИЙСКОЙ ФЕДЕРАЦИИ»

**Колледж информатики и программирования**

ПРАКТИЧЕСКАЯ РАБОТА №30

по МДК.03.01 Технология разработки программного области

по теме: РЕАЛИЗАЦИЯ МНОГОСТРАНИЧНОГО WPF-ПРИЛОЖЕНИЯ

Выполнили:

студентки группы 4ПКС-220

Батхиев А.Б.,

Долгунов Е.А.,

Мамедов Р.Э.,

Абдуллаев Т.Д.

Проверила:

преподаватель Аксёнова Т.Г.

Москва

2023

**Цель работы:** закрепить практические навыки разработки многостраничного WPF-приложения по описанию предметной области в соответствии с руководством по стилю.

**Предметная область:** тестирование студентов.



Рисунок 1. Главное меню

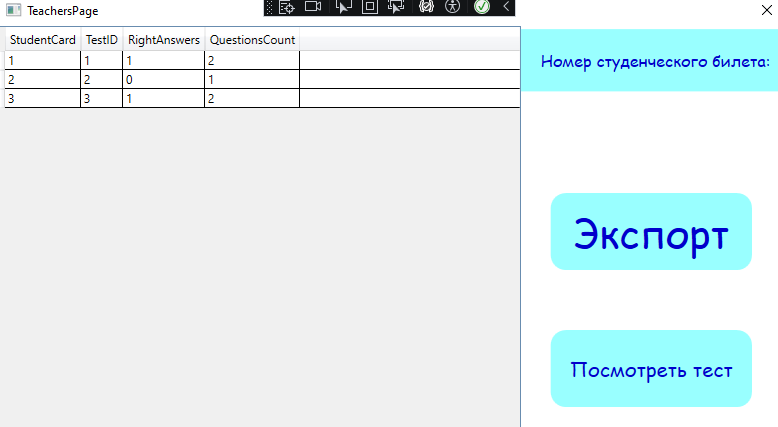


Рисунок 2. Меню учителя

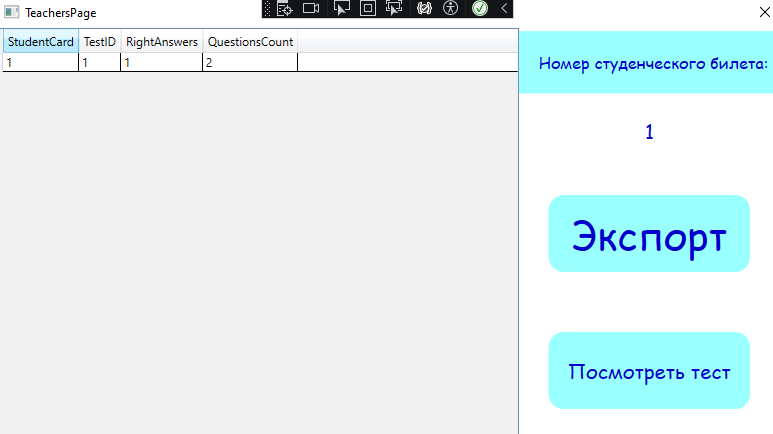


Рисунок 3. Работа поиска по номеру студенческого билета

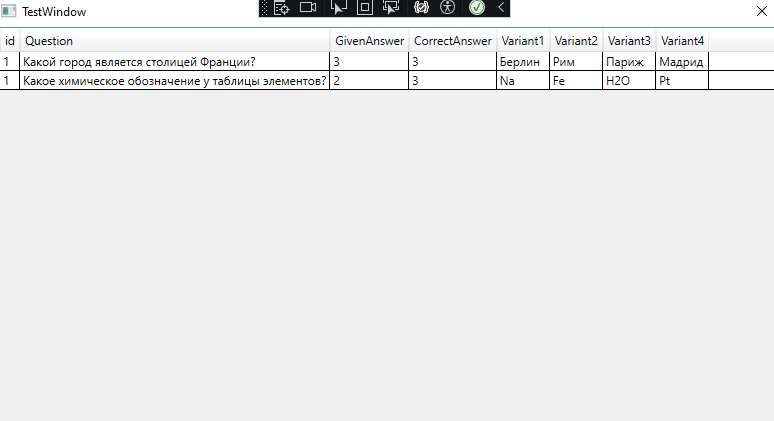


Рисунок 4. Окно с информацией о тесте

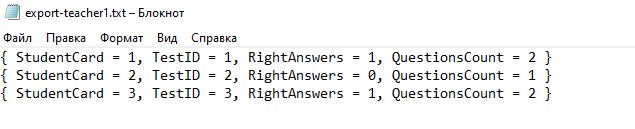


Рисунок 5. Экспортированный файл учителя

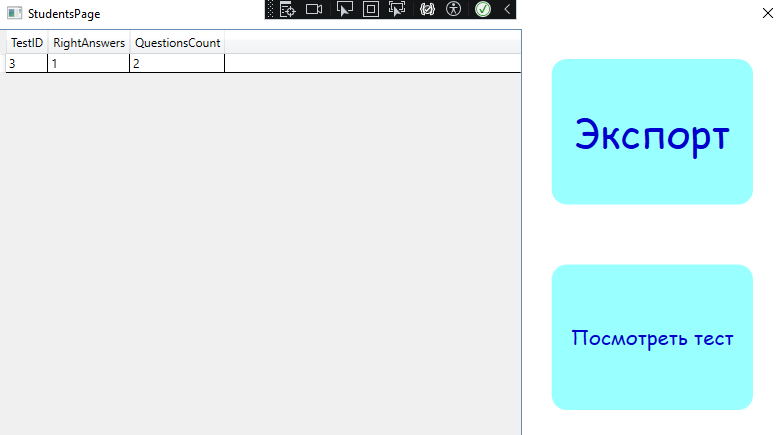


Рисунок 6. Страница студента

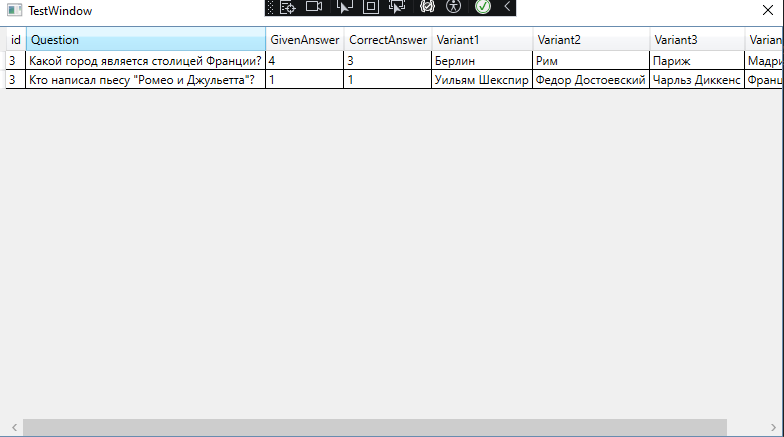


Рисунок 7. Окно с информацией о тесте

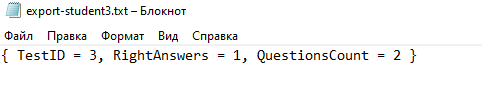


Рисунок 8. Экспортированный файл студента

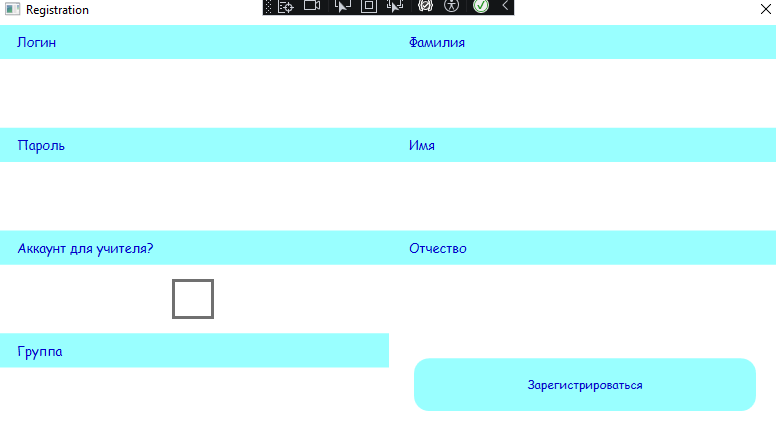


Рисунок 9. Меню регистрации

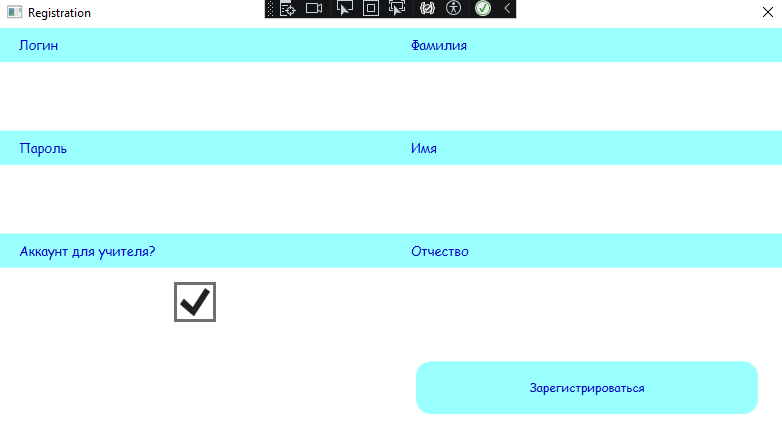


Рисунок 10. Меню регистрации

Листинг:

Файл MainWindox.xaml.cs

using PR30.Database;

using PR30.Pages;

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 PR30

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

SetVisibility(LoginInput.Text, LoginPlaceholder);

}

private void PasswordInput\_PasswordChanged(object sender, RoutedEventArgs e)

{

SetVisibility(PasswordInput.Password, PasswordPlaceholder);

}

private void SetVisibility(string text, Label placeholder)

{

placeholder.Visibility = text.Length > 0 ? Visibility.Hidden : Visibility.Visible;

}

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

{

new Registration().Show();

this.Close();

}

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

{

if (LoginInput.Text.Length == 0)

{

MessageBox.Show("Логин не может быть пустым!");

return;

}

var context = SingletonModel.GetContext();

if (context.Users.Any(x => x.Login == LoginInput.Text))

{

var user = context.Users.Where(x => x.Login == LoginInput.Text).First();

if (PasswordInput.Password == user.Password)

{

if (user.IsTeacher)

{

new TeachersPage(user).Show();

}

else

{

new StudentsPage(user).Show();

}

this.Close();

}

else

{

MessageBox.Show("Неправильный пароль");

}

}

else

{

MessageBox.Show("Такого аккаунта не существует");

}

}

}

}

Файл Registration.xaml.cs

using PR30.Database;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace PR30.Pages

{

/// <summary>

/// Логика взаимодействия для Registration.xaml

/// </summary>

public partial class Registration : Window

{

public Registration()

{

InitializeComponent();

}

private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)

{

new MainWindow().Show();

}

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

{

var context = SingletonModel.GetContext();

if (!CheckInputs(context))

{

return;

}

if (IsTeacherBox.IsChecked == true)

{

var teacher = new Teacher();

teacher.LastName = SurnameInput.Text;

teacher.Name = NameInput.Text;

teacher.Patronymic = PatronymicInput.Text;

context.Teacher.Add(teacher);

var user = new Users();

user.Login = LoginInput.Text;

user.Password = PasswordInput.Password;

user.IsTeacher = true;

user.IDTeacher = context.Teacher.Max(x => x.id);

context.Users.Add(user);

context.SaveChanges();

}

else

{

var student = new Student();

student.LastName = SurnameInput.Text;

student.Name = NameInput.Text;

student.Patronymic = PatronymicInput.Text;

student.StudentGroup = GroupInput.Text;

context.Student.Add(student);

var user = new Users();

user.Login = LoginInput.Text;

user.Password = PasswordInput.Password;

user.IsTeacher = false;

user.IDStudent = context.Student.Max(x => x.NumberStudentCard);

context.Users.Add(user);

context.SaveChanges();

}

MessageBox.Show("Регистрация успешна");

this.Close();

}

private bool CheckInputs(TestEntities context)

{

if (string.IsNullOrWhiteSpace(LoginInput.Text))

{

MessageBox.Show("Логин не может быть пустым!");

return false;

}

if (context.Users.Any(x => x.Login == LoginInput.Text))

{

MessageBox.Show("Такой аккаунт уже существует");

return false;

}

if (string.IsNullOrWhiteSpace(PasswordInput.Password))

{

MessageBox.Show("Пароль не может быть пустым!");

return false;

}

if (IsTeacherBox.IsChecked == false && string.IsNullOrWhiteSpace(GroupInput.Text))

{

MessageBox.Show("Заполните группу!");

return false;

}

if (string.IsNullOrWhiteSpace(SurnameInput.Text))

{

MessageBox.Show("Заполните фамилию!");

return false;

}

if (string.IsNullOrWhiteSpace(NameInput.Text))

{

MessageBox.Show("Заполните имя!");

return false;

}

return true;

}

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

{

if (IsTeacherBox.IsChecked == true)

{

GroupGrid.Visibility = Visibility.Hidden;

}

else

{

GroupGrid.Visibility = Visibility.Visible;

}

}

}

}

Файл StudentsPage.xaml.cs

using PR30.Database;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.IO;

using System.Linq;

using System.Runtime.Remoting.Contexts;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace PR30.Pages

{

/// <summary>

/// Логика взаимодействия для StudentsPage.xaml

/// </summary>

public partial class StudentsPage : Window

{

private Users \_user;

public StudentsPage(Users user)

{

InitializeComponent();

\_user = user;

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

var context = SingletonModel.GetContext();

var table = from rep in context.TestingReport

where rep.IDStudent == \_user.IDStudent

join qr in context.QuestionsReports on rep.id equals qr.IDTestingReport

group rep by rep.id into grep

select new

{

TestID = grep.Key,

RightAnswers = context.QuestionsReports.Where(x=> x.IDTestingReport == grep.Key)

.Where(x => x.Answer == context.TestQuestions.Where(q => q.id == x.IDTestQuestions).FirstOrDefault().CorrectAnswer).Count(),

QuestionsCount = grep.Count()

};

TestTable.ItemsSource = table.ToList();

}

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

{

using (var file = new StreamWriter($"export-student{\_user.IDStudent}.txt", false))

{

StringBuilder sb = new StringBuilder();

foreach (var item in TestTable.Items)

{

sb.AppendLine(item.ToString());

}

file.Write(sb.ToString());

}

MessageBox.Show("Экспорт завершён");

}

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

{

if (TestTable.SelectedValue != null)

{

var id = (int)TestTable.SelectedValue.GetType().GetProperty("TestID").GetValue(TestTable.SelectedValue, null);

new TestWindow(id).Show();

}

}

}

}

Файл TeachersPage.xaml.cs

using PR30.Database;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.Remoting.Contexts;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace PR30.Pages

{

/// <summary>

/// Логика взаимодействия для TeachersPage.xaml

/// </summary>

public partial class TeachersPage : Window

{

private Users \_user;

public TeachersPage(Users user)

{

InitializeComponent();

\_user = user;

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

var context = SingletonModel.GetContext();

var table = from rep in context.TestingReport

join qr in context.QuestionsReports on rep.id equals qr.IDTestingReport

group rep by new { rep.id, rep.IDStudent } into grep

select new

{

StudentCard = grep.Key.IDStudent,

TestID = grep.Key.id,

RightAnswers = context.QuestionsReports.Where(x => x.IDTestingReport == grep.Key.id)

.Where(x => x.Answer == context.TestQuestions.Where(q => q.id == x.IDTestQuestions).FirstOrDefault().CorrectAnswer).Count(),

QuestionsCount = grep.Count()

};

TestTable.ItemsSource = table.ToList();

}

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

{

using (var file = new StreamWriter($"export-teacher{\_user.IDTeacher}.txt", false))

{

StringBuilder sb = new StringBuilder();

foreach (var item in TestTable.Items)

{

sb.AppendLine(item.ToString());

}

file.Write(sb.ToString());

}

MessageBox.Show("Экспорт завершён");

}

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

{

var context = SingletonModel.GetContext();

var table = from rep in context.TestingReport

join qr in context.QuestionsReports on rep.id equals qr.IDTestingReport

group rep by new { rep.id, rep.IDStudent } into grep

select new

{

StudentCard = grep.Key.IDStudent,

TestID = grep.Key.id,

RightAnswers = context.QuestionsReports.Where(x => x.IDTestingReport == grep.Key.id)

.Where(x => x.Answer == context.TestQuestions.Where(q => q.id == x.IDTestQuestions).FirstOrDefault().CorrectAnswer).Count(),

QuestionsCount = grep.Count()

};

if (!string.IsNullOrWhiteSpace(SearchBox.Text))

{

table = table.Where(x => x.StudentCard.ToString().StartsWith(SearchBox.Text));

}

TestTable.ItemsSource = table.ToList();

}

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

{

if(TestTable.SelectedValue != null)

{

var id = (int)TestTable.SelectedValue.GetType().GetProperty("TestID").GetValue(TestTable.SelectedValue, null);

new TestWindow(id).Show();

}

}

}

}

Файл TestWindow.xaml

using PR30.Database;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace PR30.Pages

{

/// <summary>

///

/// Логика взаимодействия для TestWindow.xaml

/// </summary>

public partial class TestWindow : Window

{

private int \_id;

public TestWindow(int id)

{

InitializeComponent();

\_id = id;

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

var context = SingletonModel.GetContext();

var table = from rep in context.TestingReport

where rep.id == \_id

join qr in context.QuestionsReports on rep.id equals qr.IDTestingReport

join q in context.TestQuestions on qr.IDTestQuestions equals q.id

select new

{

rep.id,

q.Question,

GivenAnswer = qr.Answer,

CorrectAnswer = q.CorrectAnswer,

Variant1 = q.AnswerVariant1,

Variant2 = q.AnswerVariant2,

Variant3 = q.AnswerVariant3,

Variant4 = q.AnswerVariant4

};

QuestionsGrid.DataContext = table.ToList();

}

}

}

Файл SingletonModel.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Linq;

namespace PR30.Database

{

static class SingletonModel

{

private static TestEntities \_context;

public static TestEntities GetContext()

{

return \_context = \_context ?? new TestEntities();

}

}

}

<https://github.com/RuslanM1102/TRPO-PR30>

Вывод: Закрепили практические навыки разработки многостраничного WPF-приложения по описанию предметной области в соответствии с руководством по стилю.