МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет

им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации

Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине

Технология разработки программного обеспечения

Тема «Разработка документации на программное обеспечение»

Работу выполнил

Студент гр.4338

Бусов В.Р.

Принял

Преподаватель Исламова Л.М

Казань 2025

**Описание программы**

Обозначение и наименование программы: Kurators App

Программное обеспечение, необходимое для функционирования программы: Microsoft Visual Studio, Excel, и Microsoft SQL Server

Языки программирования, на которых написана программа: C#, XML, SQL

**Функциональное назначение**

Программа предназначена для автоматизации процесса обработки данных, включая их сбор, анализ, хранение и визуализацию. Она позволяет пользователям эффективно управлять большими объемами информации и формировать отчеты в удобном формате.

Классы решаемых задач:

* Управление списком студентов и информацией о них (ФИО, номер телефона, группа, фио куратора).
* Организация работы со студентами для куратора (авторизация и информация о группе куратора).
* Добавление студентов, заполнение информации о них.
* Изменение информации о студентах, родителях, кураторах.
* Поиск студентов.

Назначение программы:

Программа предназначена для оптимизации работы куратора, сокращения времени на административные задачи и повышения качества кураторской деятельности. Система позволяет:

* Вести оперативный учет студентов, контролировать их успеваемость и своевременно обновлять информацию.
* Автоматизировать процесс сопоставления информации о студентах и их родителях.
* Хранить и анализировать данные об учебной успеваемости.

**Описание логической структуры**

Алгоритм программы:

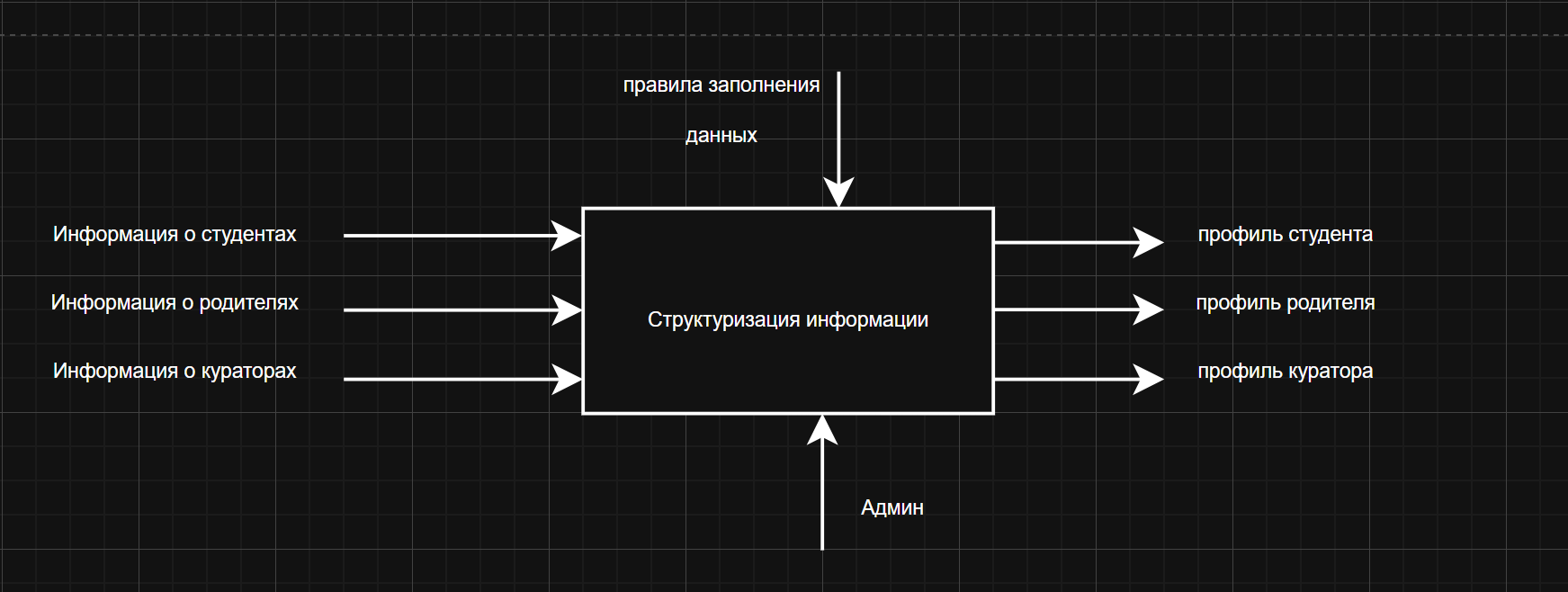


Рисунок 1 – Первый уровень SADT диаграмм

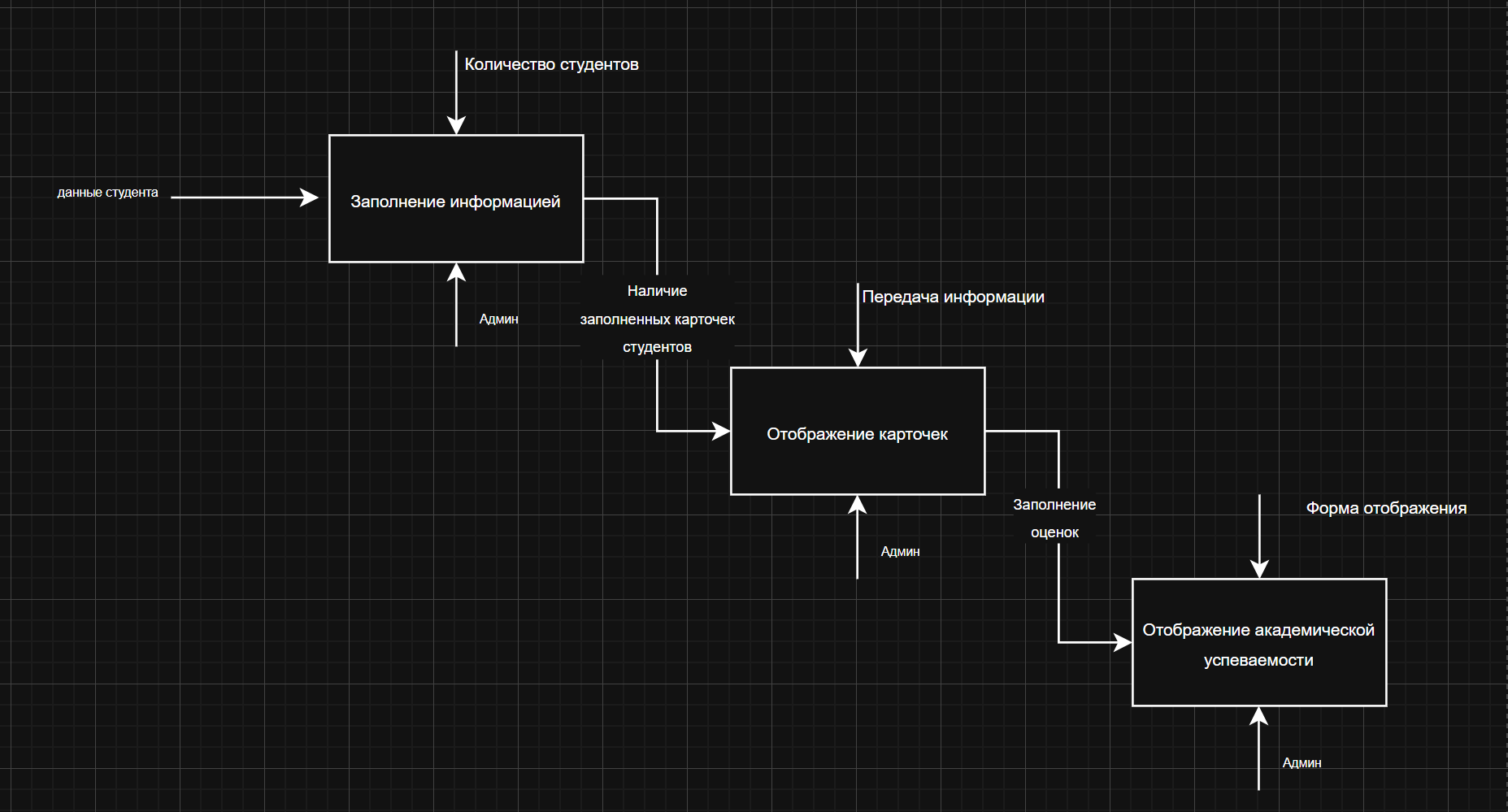


Рисунок 2 – Второй уровень SADT диаграмм

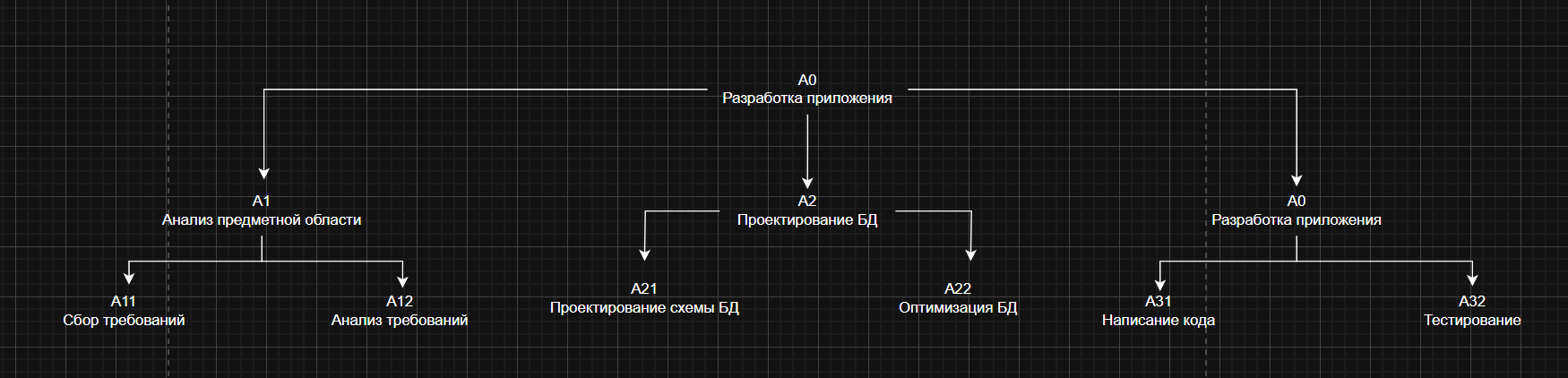


Рисунок 3 – Третий уровень SADT диаграмм

Используемые методы: Методы программирования.

Структура программы с описанием функций составных частей и связи между ними: Таблицы из бд: Student - это таблица показывает сведения студенте, его ФИО, группу, дату рождения, номер телефона и группы, а также id его куратора. Parents - сведения о родителях. SummaryLedger - сведения об академической успеваемости. StudentParentRaletionship - сведения о связи студента и его родителя. Curator - сведения о кураторе. Teachers – информация о преподователях.

Связи программы с другими программами: Связь между СУБД MS SQL Server и Средой разработки Microsoft Visual Studio, между Средой разработки Microsoft Visual Studio и Excel.

**Используемые технические средства**

Персональный компьютер:

Аппаратные требования

Процессор:

Минимальный: Intel Core i3 или аналогичный AMD.

Рекомендуемый: Intel Core i5/i7 или выше.

Оперативная память (RAM):

Минимальная: 4 ГБ.

Рекомендуемая: 8 ГБ или более.

Жесткий диск:

Минимальный объем свободного места: 5 ГБ.

Рекомендуемый тип диска: SSD для ускорения работы.

Видеокарта:

Минимальные требования: поддержка OpenGL 2.0.

Рекомендуемые: дискретная видеокарта с поддержкой DirectX 11 для работы с графикой.

Программные требования

Операционная система:

Windows 10/11, macOS 10.15 или выше, Linux (Ubuntu 20.04 LTS или аналогичные дистрибутивы).

Среда выполнения:

Python 3.8 или выше (для скриптовых решений).

.NET Framework 4.7.2 или выше (для приложений на C#).

Базы данных:

Поддержка MySQL, PostgreSQL или SQLite.

Дополнительное ПО:

Установленные библиотеки: NumPy, Pandas, Matplotlib (для анализа данных).

Браузер с поддержкой HTML5 для веб-интерфейса (Google Chrome, Mozilla Firefox).

**Вызов и загрузка**

Windows:

Через меню "Пуск": Нажмите кнопку "Пуск", найдите название программы в списке установленных приложений и щелкните по нему.

Через ярлык на рабочем столе: Дважды щелкните по значку программы.

Через командную строку: Введите имя исполняемого файла (например, program.exe) в командной строке.

macOS:

Через Finder: Откройте папку "Программы", найдите приложение и дважды щелкните по нему.

Через Spotlight: Нажмите Cmd + Space, введите название программы и выберите её из результатов поиска.

Linux:

Через терминал: Введите команду запуска (например, ./program или python script.py).

Через графический интерфейс: Найдите программу в меню приложений и щелкните по ней.

Входные точки в программу: Окно авторизации и кнопка выхода

**Входные и выходные данные**

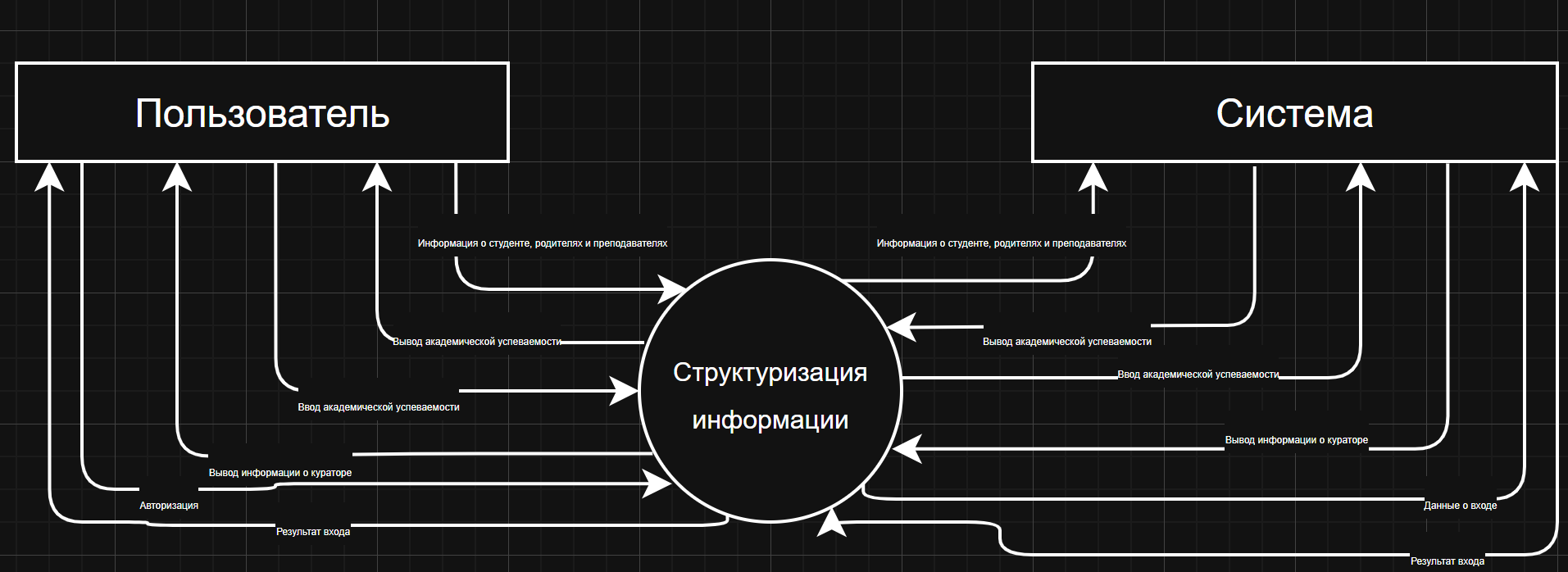
****

Рисунок 4 – Контекстная диаграмма

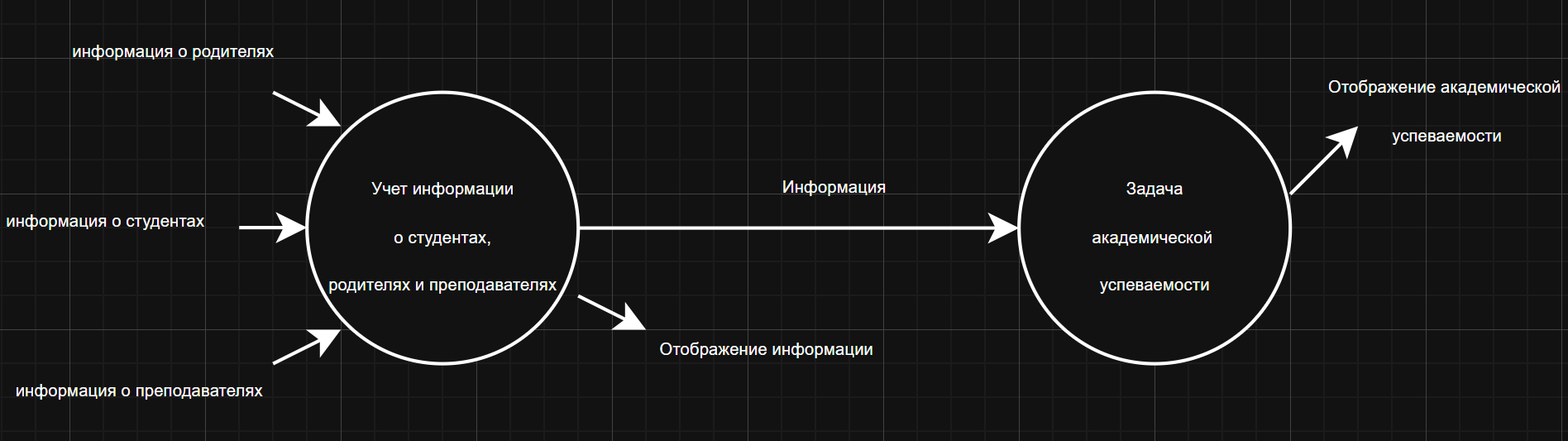


Рисунок 5 – Детализация процесса учета информации

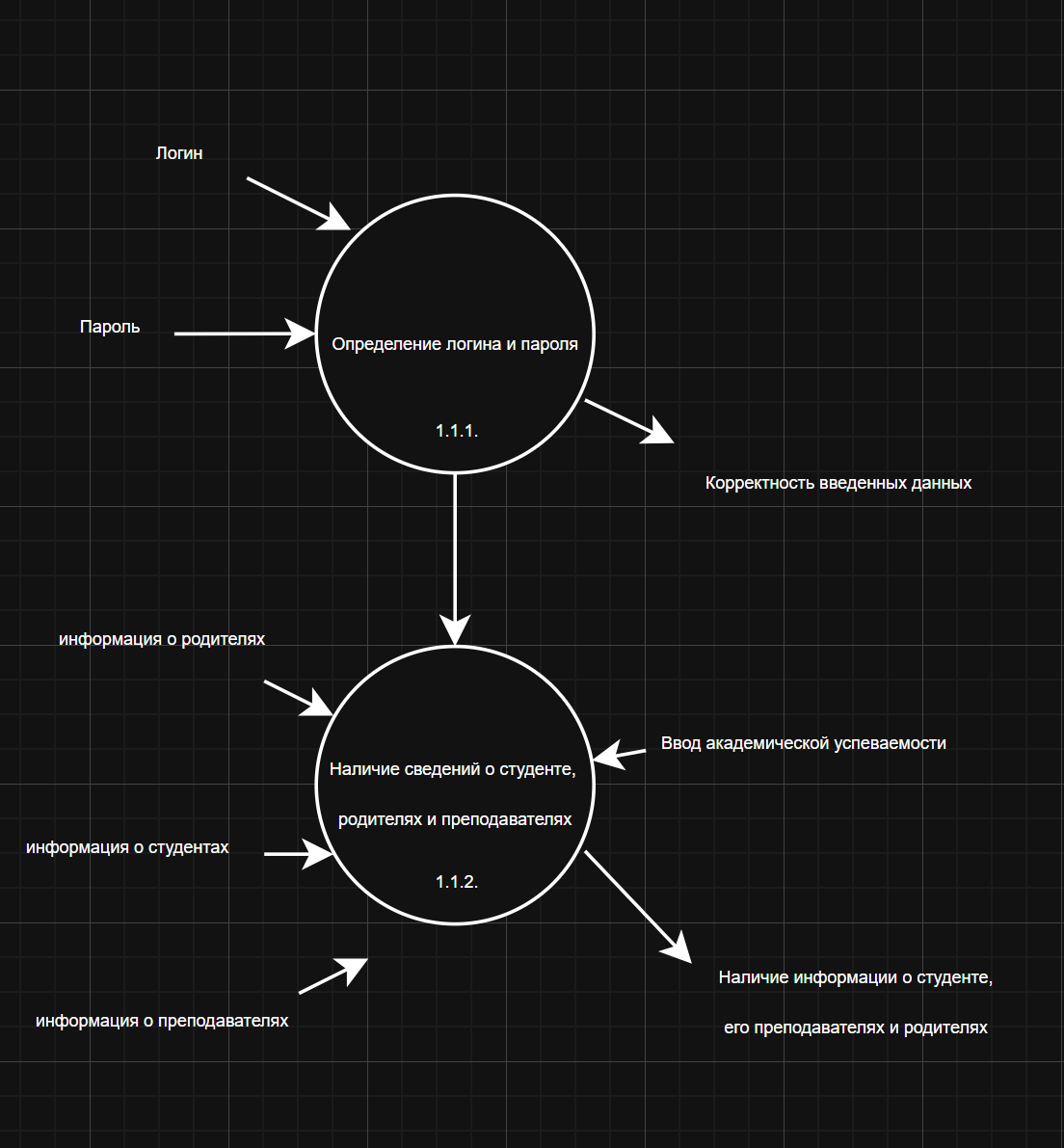


Рисунок 6 – Детализация процесса авторизации и обработки запроса на наличие сведений о студенте, преподавателях и родителях

**Руководство пользователя**

**Назначение программы**

Виды деятельности и функции, которые автоматизированы:

- Управление данными о студентах, преподавателях и родителях (удаление, добавление, изменение)

- Управление академической успеваемостью (добавление и изменение значения)

- Управление ФИО студентов (реализация поиска)

**Условия выполнения программы**

- Минимальный состав аппаратных средств: Компьютер с ОС Windows/Linux, 4 ГБ ОЗУ

- Минимальный состав программных средств: Установленная СУБД

- Требования к персоналу: Базовые навыки работы с ПК

**Выполнение программы**

- Загрузка и запуск программы

- Выполнение программы (функции программы, процессы работы с данными, выполнение конкретных задач пользователя):

Регистрация пользователя

Авторизация

Вывод информации о текущем пользователе

Для каждой таблицы в базе данных реализовано:

Загрузка данных из БД

Поиск записи по каким либо критериям

Добавление записи в БД

Изменения записи в БД

Удаление записи в БД

- Завершение работы программы

**Сообщения оператору**

- “Успешный вход”

- “Неверный логин или пароль”

- “Вы вышли из системы”

- “Запись успешно добавлена”

- “Запись успешно изменена”

- “Запись успешно удалена”

- “Что пошло не так”

* **Приложение**

Окно предназначено для работы с кураторами. Включает поля для ввода имени и фамилии, номера и почты. Кнопки "Добавить", "Изменить" и "Удалить" позволяют управлять записями. Рис 7 и 8

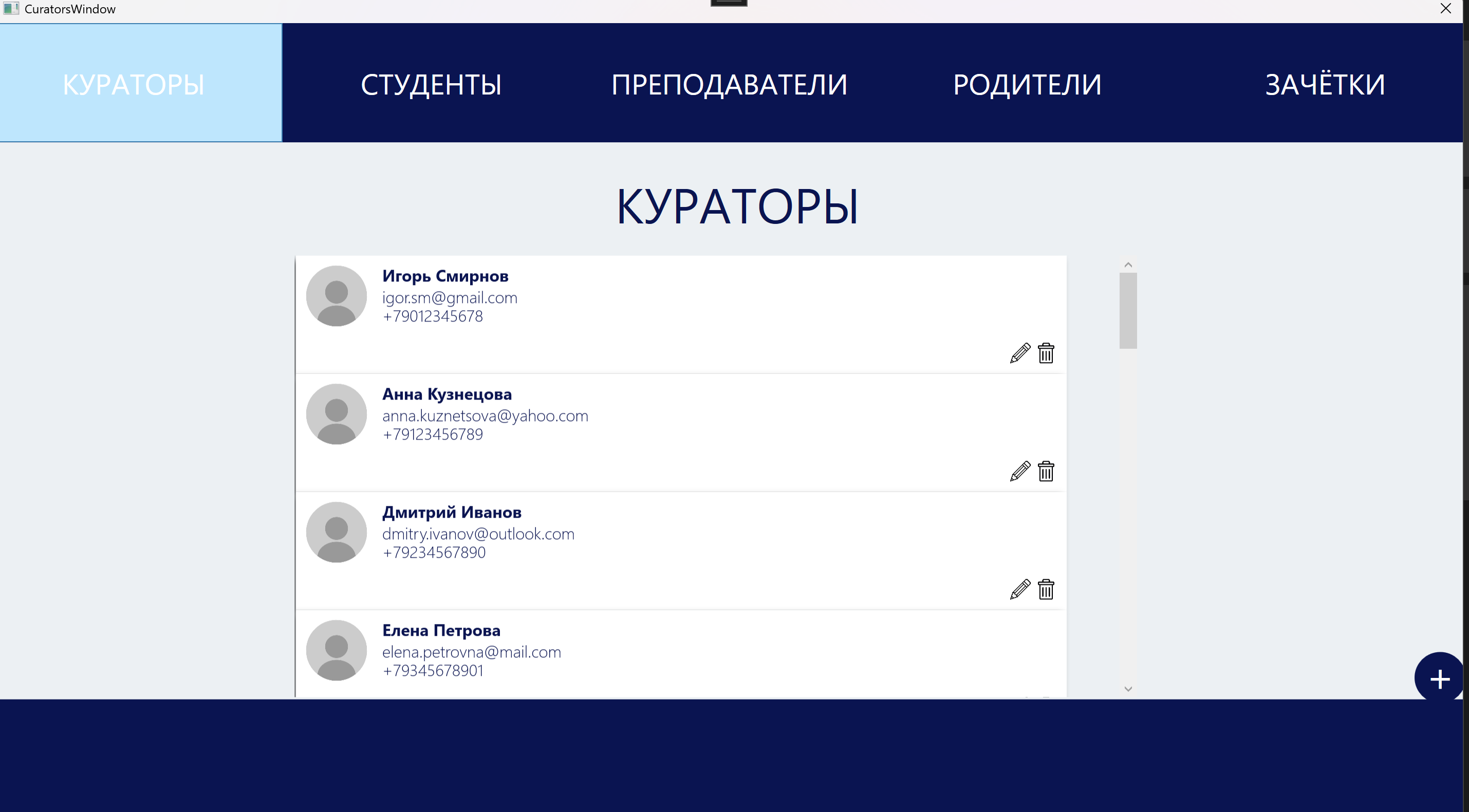


Рисунок 7 – Окно «Управление данными кураторов»

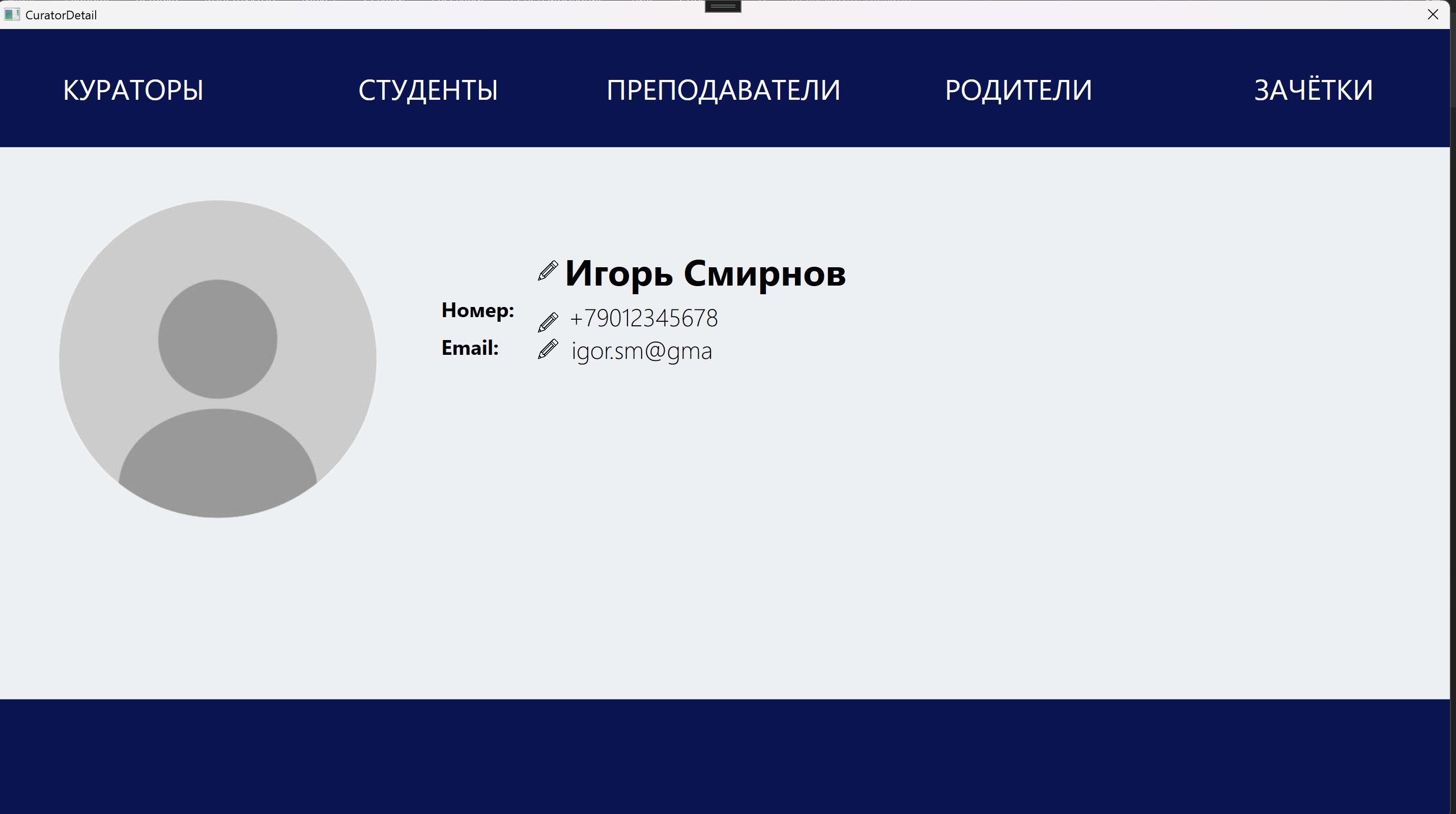


Рисунок 8 – Окно «Изменение информации»

Интерфейс для работы с информацией о студентах. Поля ввода содержат название имя и фамилию, телефон, дату рождения, группу, куратора, родителей. Данные отображаются, а кнопки обеспечивают добавление, сохранение и удаление записей. Рис 9 и 10

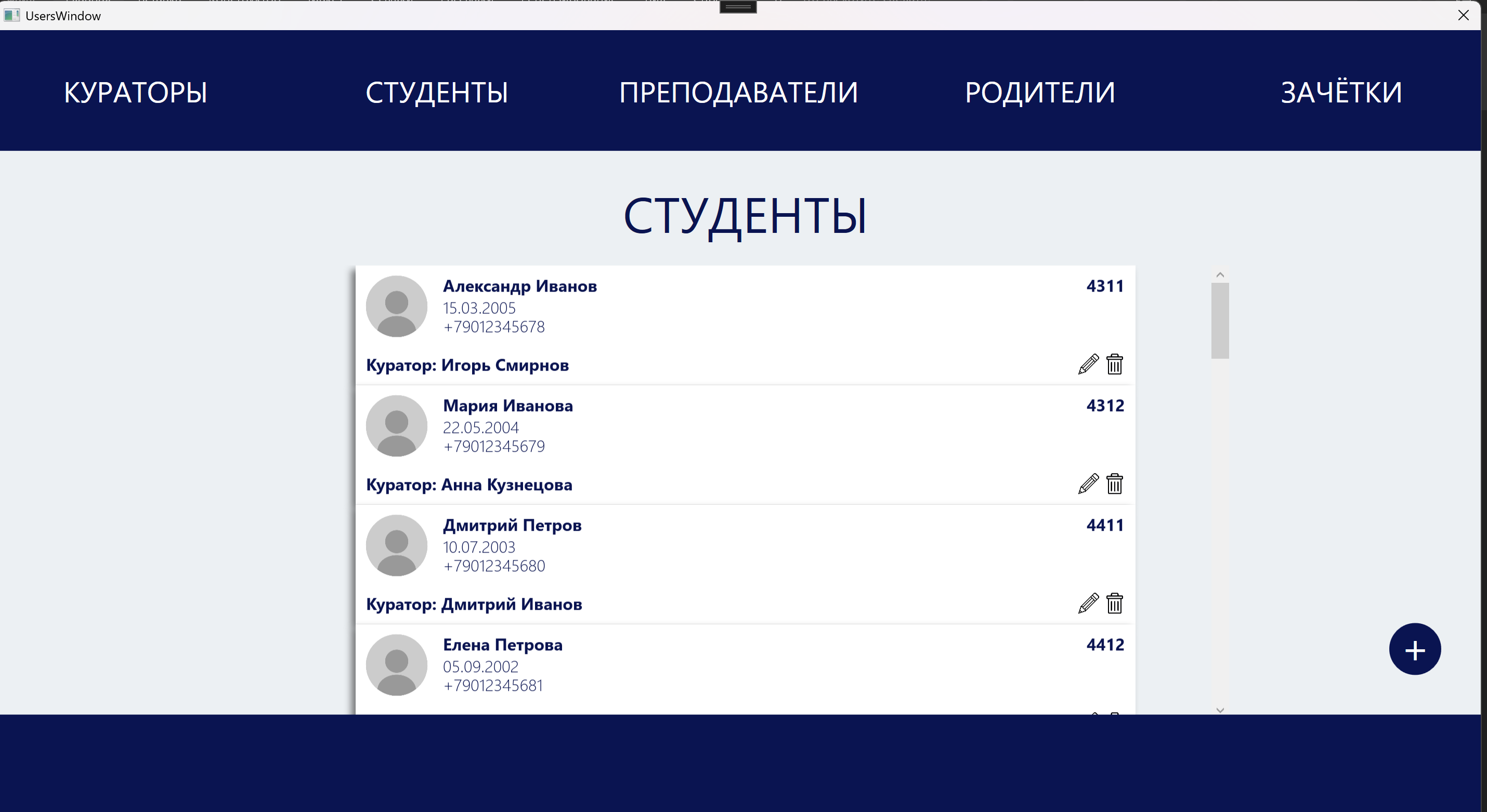


Рисунок 9 – Окно «Управление данными студентов»

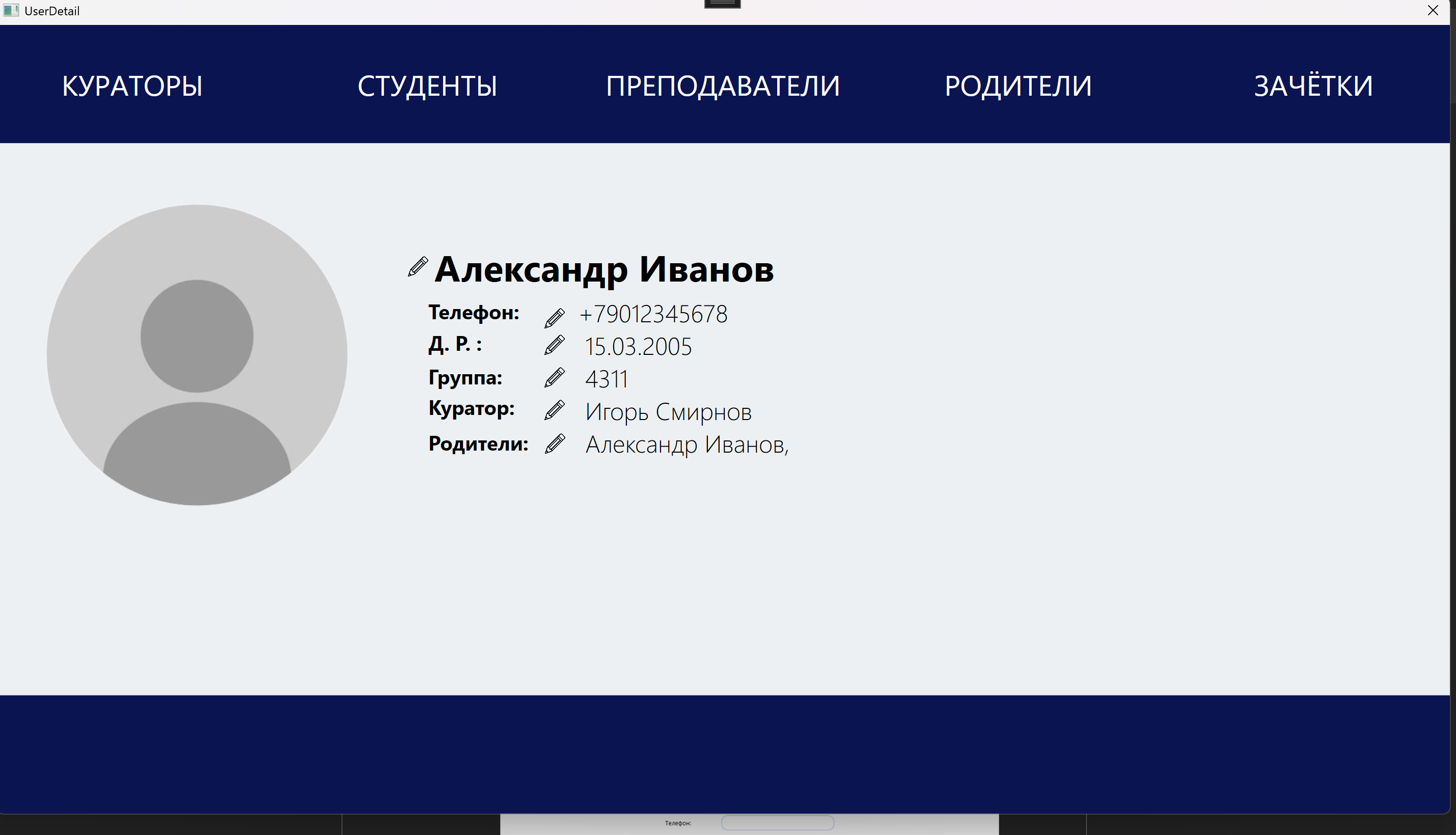


Рисунок 10 – Окно «Изменение информации о студенте»

Интерфейс для работы с информацией о преподавателях. Поля ввода содержат имя и фамилию, предмет. Данные отображаются, а кнопки обеспечивают добавление, сохранение и удаление записей. Рис 11 и 12

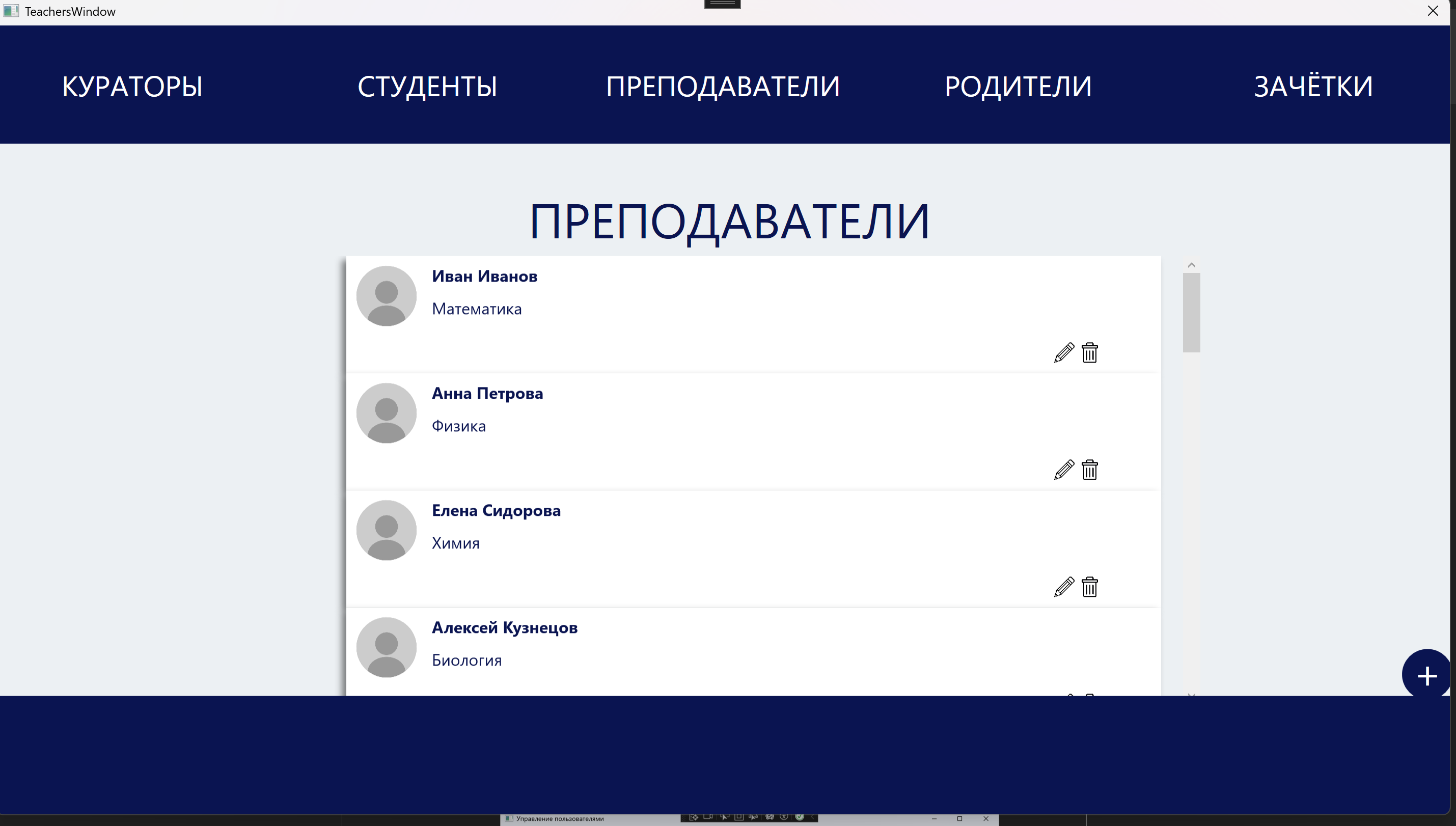


Рисунок 11 – Окно «Управление данными преподавателей»

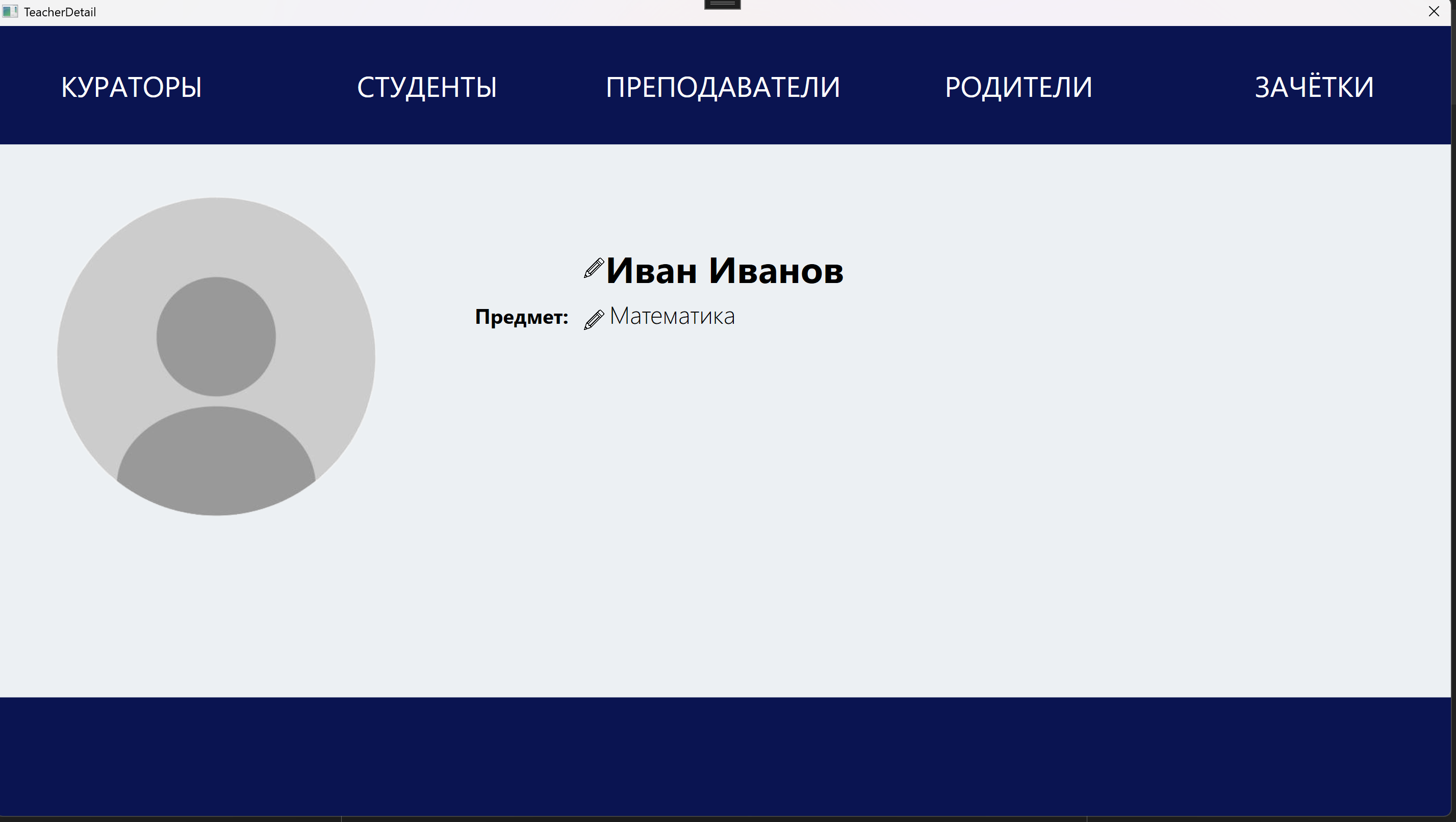


Рисунок 12 – Окно «Изменение информации»

Демонстрирует процесс ввода данных о родителе: имя и фамилию, номер телефона, почту и место работы. Поля подсвечиваются при корректном заполнении, что облегчает работу пользователя. Рис 13 и 14

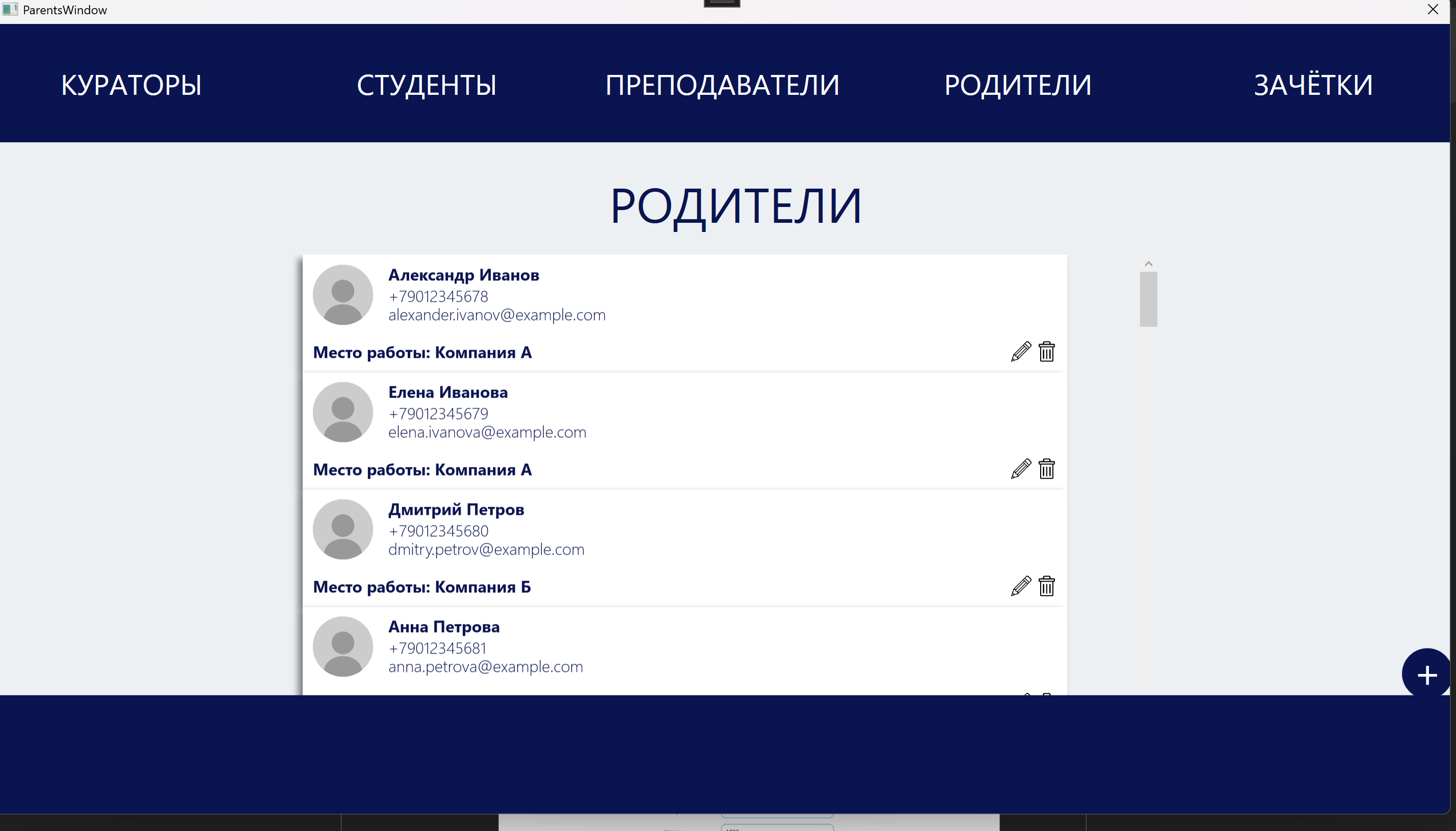


Рисунок 13 – Окно «Управление данными родителей»

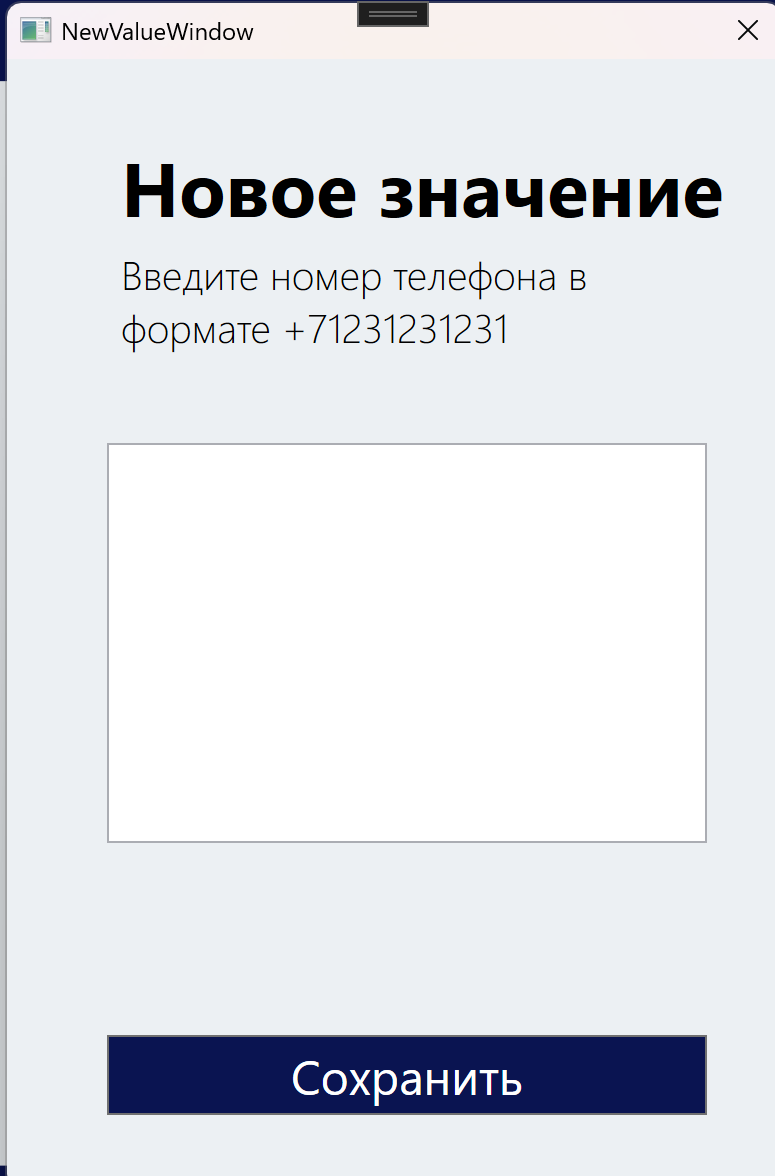


Рисунок 14 – «Заполнение карточки»

Интерфейс для работы с академической успеваемостью студентов. Поля ввода содержат имя и фамилию студента, имя и фамилию преподавателя, предмет, семестр и оценку за него. Данные отображаются, а кнопки обеспечивают добавление, сохранение и удаление записей. Рис 15 и 16

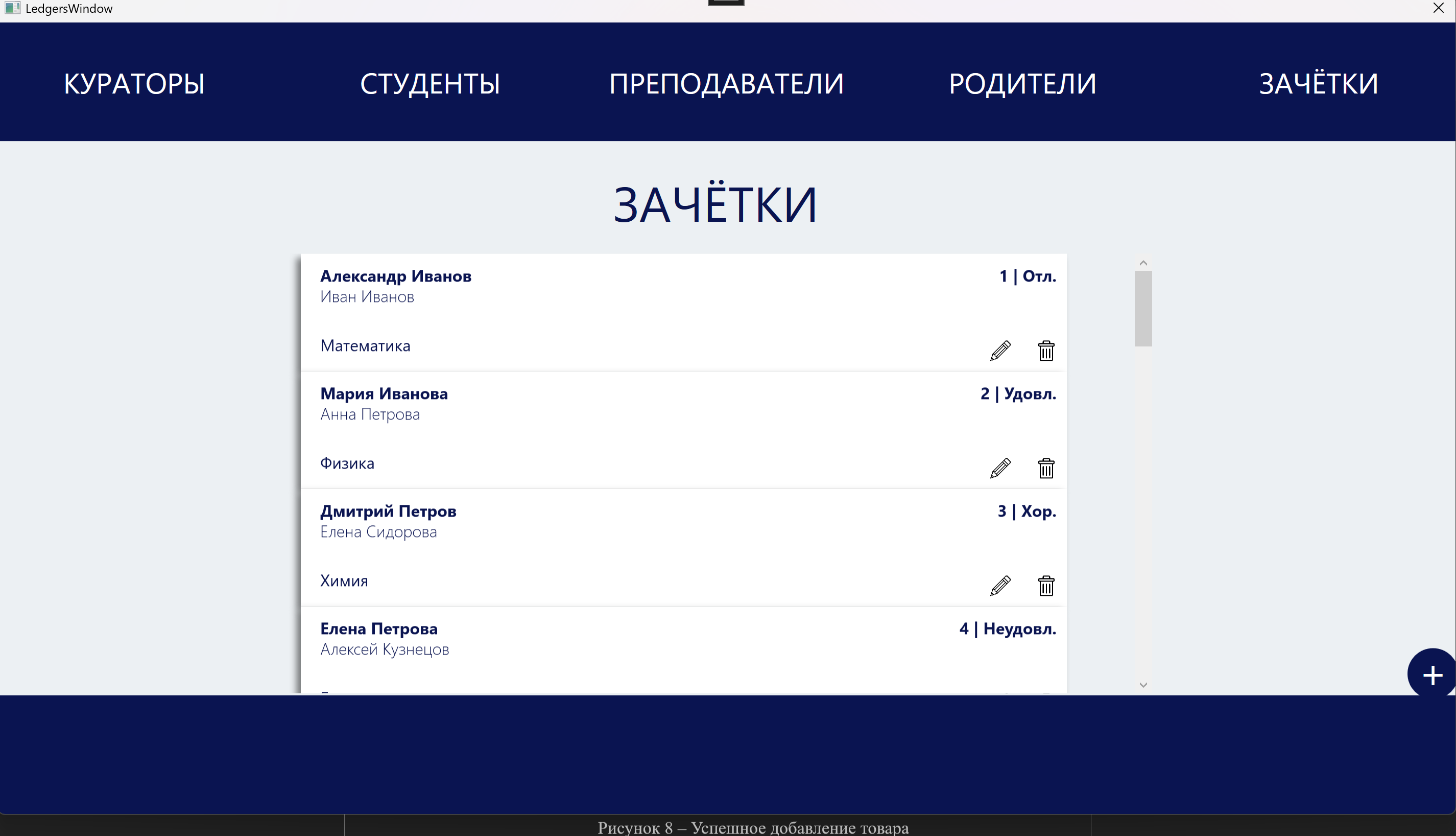


Рисунок 15 – «Отображение списка успеваемости»

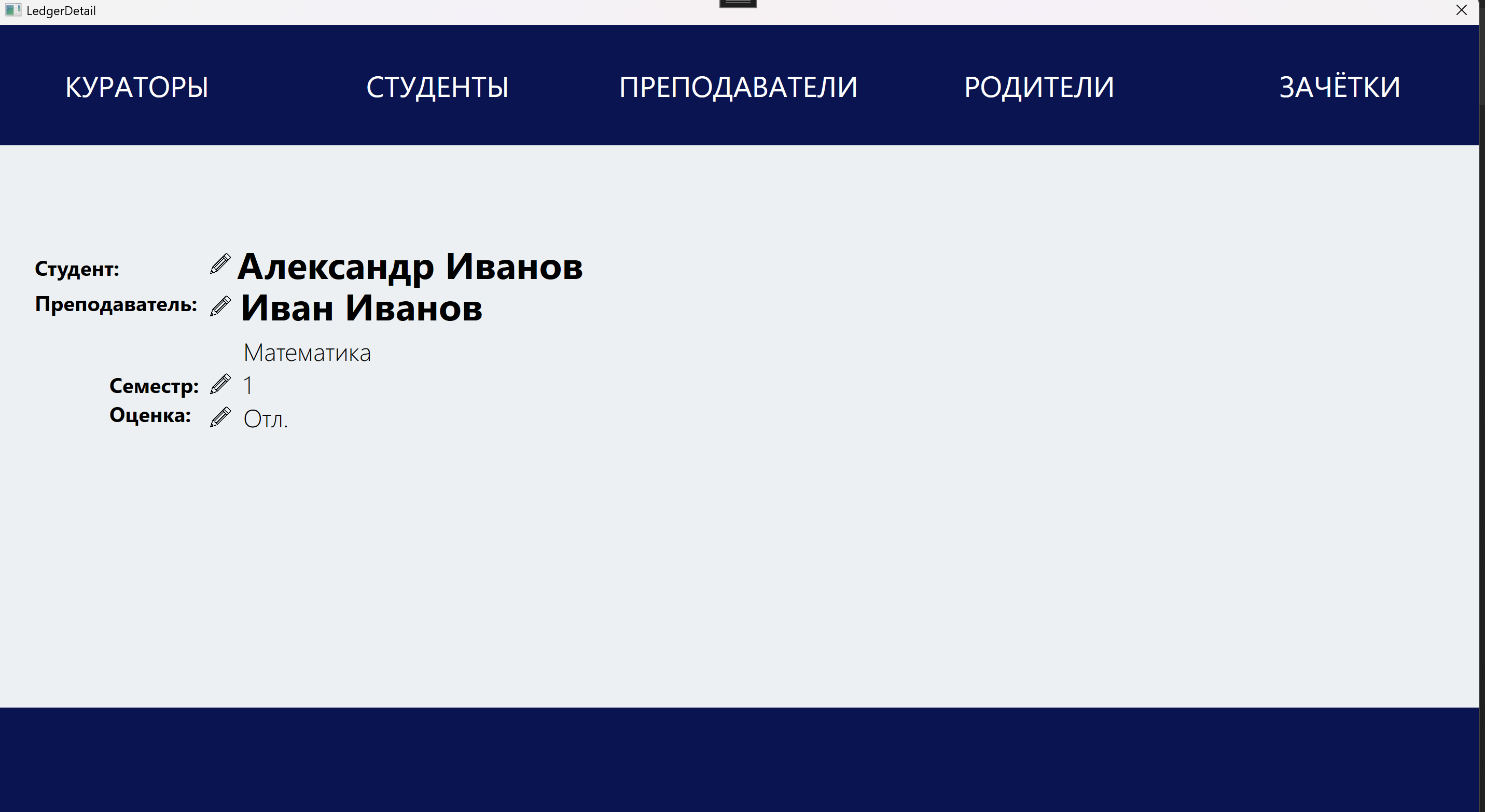


Рисунок 16 – «Добавление данных»

Показывает подтверждение изменения данных карточки. Система выводит сообщение, указывающее на успешное действие. Рис 17

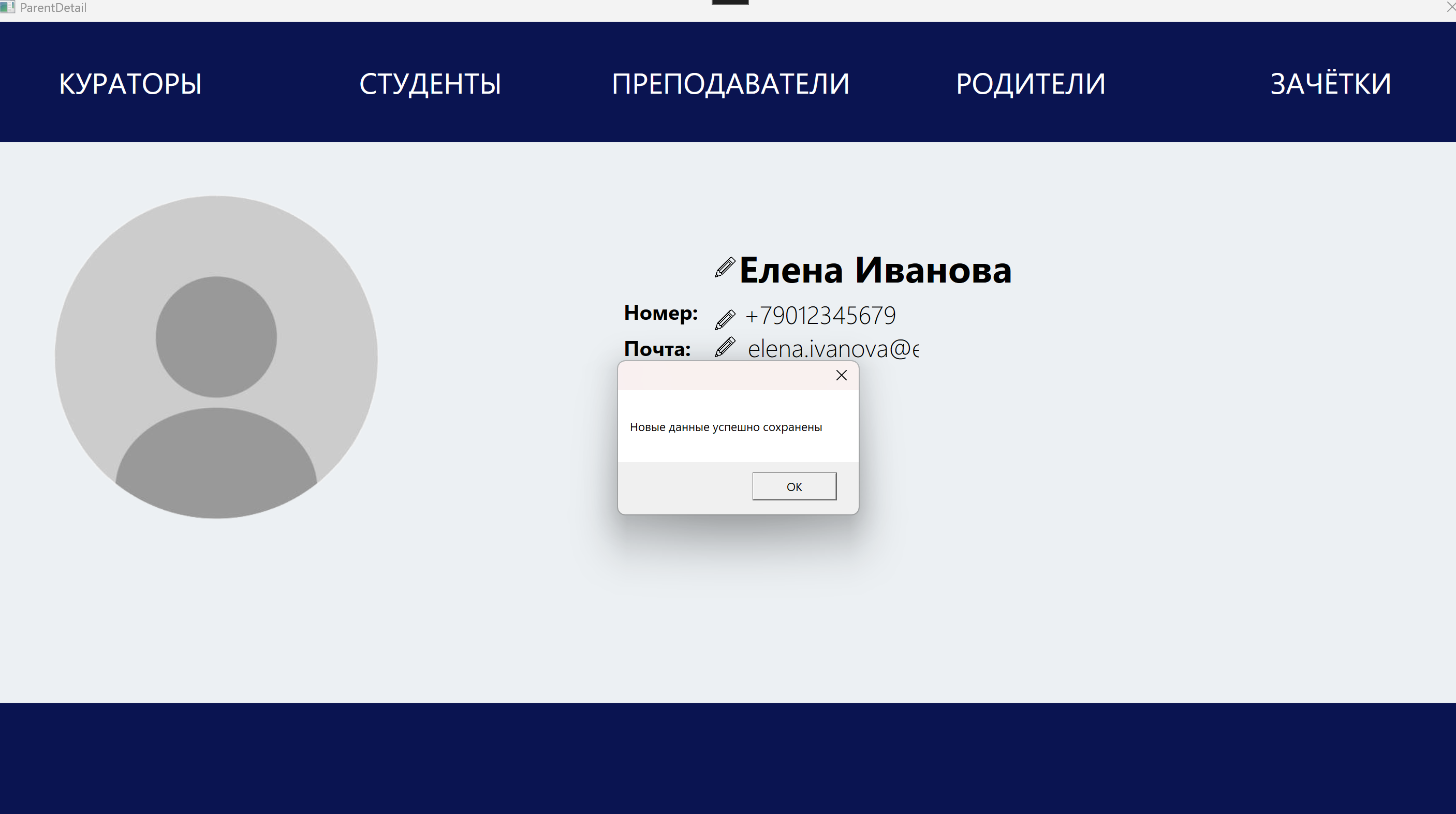


Рисунок 17 – «Успешное редактирование данных»

**Приложения**

<Window x:Class="WpfApp1.CuratorDetail"

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"

ResizeMode="NoResize"

Height="810" Width="1440"

Background="#ECF0F3"

WindowStartupLocation="CenterScreen"

Title="CuratorDetail">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Width" Value="290"/>

<Setter Property="Background" Value="#0a1451"/>

<Setter Property="BorderBrush" Value="#0a1451"/>

<Setter Property="FontSize" Value="28"/>

<Setter Property="Foreground" Value="#FFFFFF"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=".15\*"/>

<RowDefinition Height=".7\*"/>

<RowDefinition Height=".15\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Background="#0A1451" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="КУРАТОРЫ" Click="OpenCurators"/>

<Button Content="СТУДЕНТЫ" Click="OpenStudents"/>

<Button Content="ПРЕПОДАВАТЕЛИ" Click="OpenTeachers"/>

<Button Content="РОДИТЕЛИ" Click="OpenParents"/>

<Button Content="ЗАЧЁТКИ"/>

</StackPanel>

<Canvas Grid.Row="1">

<Image Source="Source/avatar guest.png" Height="416" Width="433"/>

<TextBlock Name="Name" FontSize="36" FontWeight="Bold" Text="Бусов Владислав" Canvas.Left="557" Canvas.Top="97" HorizontalAlignment="Left" VerticalAlignment="Center" Width="396"/>

<Image Cursor="Hand" MouseUp="EditName" Source="Source/edit.png" Width="28" Height="20" Canvas.Left="531" Canvas.Top="111" HorizontalAlignment="Left" VerticalAlignment="Center" RenderTransformOrigin="0.6,0.117"/>

<TextBlock Name="Phone" FontSize="24" FontWeight="Light" Text="+79196395387" Canvas.Left="561" Canvas.Top="150" HorizontalAlignment="Left" VerticalAlignment="Center" Width="193"/>

<Image Cursor="Hand" MouseUp="EditPhone" Source="Source/edit.png" Width="28" Height="20" Canvas.Left="531" Canvas.Top="162" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock Name="Email" FontSize="24" FontWeight="Light" Text="15.07.2006" Canvas.Left="564" Canvas.Top="182" HorizontalAlignment="Left" VerticalAlignment="Center" Width="140"/>

<Image Cursor="Hand" MouseUp="EditEmail" Source="Source/edit.png" Width="28" Height="20" Canvas.Left="531" Canvas.Top="188" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock x:Name="Name\_Копировать" FontSize="20" FontWeight="Bold" Canvas.Left="436" Canvas.Top="145" Width="90" HorizontalAlignment="Left" VerticalAlignment="Center" Height="36"><Run Language="ru-ru" Text="Номер:"/></TextBlock>

<TextBlock x:Name="Name\_Копировать1" FontSize="20" FontWeight="Bold" Canvas.Left="436" Canvas.Top="182" Width="78" HorizontalAlignment="Left" VerticalAlignment="Center" Text="Email:"/>

</Canvas>

<Rectangle Grid.Row="2">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

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.Shapes;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для CuratorDetail.xaml

/// </summary>

public partial class CuratorDetail : Window

{

public Curator curator;

CuratorsEntities1 db;

public CuratorDetail(Curator curator)

{

this.curator = curator;

this.db = CuratorsEntities1.GetContext();

InitializeComponent();

UpdateData();

}

public void UpdateData()

{

Name.Text = $"{this.curator.FirstName} {this.curator.LastName}";

Phone.Text = this.curator.CellPhone;

Email.Text = this.curator.Email;

}

private void OpenCurators(object sender, RoutedEventArgs e)

{

(new CuratorsWindow()).Show();

this.Close();

}

private void OpenStudents(object sender, RoutedEventArgs e)

{

(new UsersWindow()).Show();

this.Close();

}

private void OpenParents(object sender, RoutedEventArgs e)

{

(new ParentsWindow()).Show();

this.Close();

}

private void OpenTeachers(object sender, RoutedEventArgs e)

{

(new TeachersWindow()).Show();

this.Close();

}

private void EditName(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите данные в формате Имя Фамилия");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.Length != 0 && new\_value.Split(' ').Length == 2)

{

curator.FirstName = new\_value.Split(' ')[0];

curator.LastName = new\_value.Split(' ')[1];

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! '{new\_value}' должно состоять из двух слов: Имя Фамилия, разделенных пробелом");

UpdateData();

}

private void EditPhone(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите номер телефона в формате +71231231231");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.StartsWith("+7") && new\_value.Length == 12 && long.TryParse(new\_value.Replace("+", ""), out \_))

{

curator.CellPhone = new\_value;

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! '{new\_value}' должен начинаться с +7 и состоять из 11 цифр");

UpdateData();

}

private void EditEmail(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите электронную почту в формате test@test.test");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.Contains("@") && new\_value.Contains("."))

{

curator.Email = new\_value.ToString();

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! '{new\_value}' не подходит под требования к формату данных");

UpdateData();

}

}

}  
<Window x:Class="WpfApp1.CuratorsWindow"

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"

ResizeMode="NoResize"

Height="810" Width="1440"

Background="#ECF0F3"

WindowStartupLocation="CenterScreen"

Title="CuratorsWindow">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Width" Value="290"/>

<Setter Property="Background" Value="#0a1451"/>

<Setter Property="BorderBrush" Value="#0a1451"/>

<Setter Property="FontSize" Value="28"/>

<Setter Property="Foreground" Value="#FFFFFF"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=".15\*"/>

<RowDefinition Height=".7\*"/>

<RowDefinition Height=".15\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Background="#0A1451" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="КУРАТОРЫ" Click="OpenCurators"/>

<Button Content="СТУДЕНТЫ" Click="OpenStudents"/>

<Button Content="ПРЕПОДАВАТЕЛИ" Click="OpenTeachers"/>

<Button Content="РОДИТЕЛИ" Click="OpenParents"/>

<Button Content="ЗАЧЁТКИ" Click="OpenLedgers"/>

</StackPanel>

<Canvas Grid.Row="1" Cursor="Hand">

<TextBlock Text="КУРАТОРЫ"

Canvas.Left="602"

Canvas.Top="27"

FontSize="48"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Foreground="#0A1451"/>

<ScrollViewer Height="430" Canvas.Left="290" Canvas.Top="110" Width="820" VerticalScrollBarVisibility="Auto">

<StackPanel Name="CardsList">

<WrapPanel Grid.Column="0" Width="800" Name="UserCard0">

<Canvas Background="#FFFFFF" Width="750" Height="115">

<Image Source="Source/avatar guest.png" Height="87" Width="79" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBlock Text="Александр Шумилкин" FontSize="16" Foreground="#0A1451" FontWeight="Bold" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="8"/>

<TextBlock Text="test@test.com" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="29"/>

<TextBlock Text="+79196395387" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="47"/>

<Image Source="Source/edit.png" Width="20" Height="20" Canvas.Left="695" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<Image Source="Source/trash.png" Width="20" Height="20" Canvas.Left="720" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Canvas.Effect>

<DropShadowEffect Opacity=".5" RenderingBias="Performance" Direction="225" ShadowDepth="6" BlurRadius="10"/>

</Canvas.Effect>

</Canvas>

</WrapPanel>

</StackPanel>

</ScrollViewer>

<Button Click="CreateCurator" Height="50" Width="50" Canvas.Left="1380" Canvas.Top="496" Foreground="#FFFFFF" FontSize="40">

<Button.Style>

<Style TargetType="Button">

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="true">

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

</Trigger>

</Style.Triggers>

</Style>

</Button.Style>

<Button.Template>

<ControlTemplate>

<Border CornerRadius="25" Background="#0A1451">

<ContentPresenter Content="+" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0,-5,0,0"/>

</Border>

</ControlTemplate>

</Button.Template>

</Button>

</Canvas>

<Rectangle Grid.Row="2">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

</Grid>

</Window>

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.Markup;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для CuratorsWindow.xaml

/// </summary>

public partial class CuratorsWindow : Window

{

public CuratorsWindow()

{

InitializeComponent();

ShowData();

}

public void ShowData()

{

var db = CuratorsEntities1.GetContext();

List<Curator> curators = db.Curator.ToList();

if (curators.Count == 0) this.CardsList.Children.Clear();

else if (curators.Count > 1)

{

for (int i = 1; i < curators.Count; i++)

{

WrapPanel el = XamlReader.Parse(

XamlWriter.Save(

this.UserCard0

)

) as WrapPanel;

this.RegisterName($"UserCard{i}", el);

this.CardsList.Children.Add(el);

}

}

for (int i = 0; i < curators.Count; i++)

{

var curator = curators[i];

int TBCounter = 0;

int ICounter = 0;

Canvas container = (this.FindName($"UserCard{i}") as WrapPanel).Children[0] as Canvas;

foreach (UIElement el in container.Children)

{

string val = "";

if (el.GetType().ToString() == "System.Windows.Controls.TextBlock")

{

if (TBCounter == 0) val = $"{curator.FirstName} {curator.LastName}";

else if (TBCounter == 1) val = curator.Email;

else if (TBCounter == 2) val = curator.CellPhone;

el.SetValue(TextBlock.TextProperty, val);

TBCounter++;

}

if (el.GetType().ToString() == "System.Windows.Controls.Image")

{

if (ICounter == 1)

{

el.MouseUp += (o, e) =>

{

var win = new CuratorDetail(curator);

win.Show();

this.Close();

};

}

else if (ICounter == 2)

{

el.MouseUp += (o, e) =>

{

int count\_students = db.Student.Where(s => s.CuratorId == curator.CuratorID).ToList().Count;

if (count\_students == 0)

{

db.Curator.Remove(curator);

db.SaveChanges();

var win = new CuratorsWindow();

win.Show();

this.Close();

} else

{

MessageBox.Show("Ошибка! Вы не можете удалить куратора, пока к нему привязаны студенты");

}

};

}

ICounter++;

}

}

}

}

private void OpenCurators(object sender, RoutedEventArgs e)

{

(new CuratorsWindow()).Show();

this.Close();

}

private void OpenStudents(object sender, RoutedEventArgs e)

{

(new UsersWindow()).Show();

this.Close();

}

private void OpenParents(object sender, RoutedEventArgs e)

{

(new ParentsWindow()).Show();

this.Close();

}

private void OpenTeachers(object sender, RoutedEventArgs e)

{

(new TeachersWindow()).Show();

this.Close();

}

private void OpenLedgers(object sender, RoutedEventArgs e)

{

(new LedgersWindows()).Show();

this.Close();

}

private void CreateCurator(object sender, RoutedEventArgs e)

{

var db = CuratorsEntities1.GetContext();

Curator curator = new Curator()

{

FirstName = "\_\_\_\_\_\_\_\_\_\_",

LastName = "\_\_\_\_\_\_\_\_\_\_",

CellPhone = "+78917264893",

Email = "\_\_.\_\_.\_\_\_\_",

};

db.Curator.Add(curator);

db.SaveChanges();

var win = new CuratorDetail(curator);

win.Show();

this.Close();

}

}

}

<Window x:Class="WpfApp1.LedgerDetail"

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"

ResizeMode="NoResize"

Height="810" Width="1440"

Background="#ECF0F3"

WindowStartupLocation="CenterScreen"

Title="LedgerDetail">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Width" Value="290"/>

<Setter Property="Background" Value="#0a1451"/>

<Setter Property="BorderBrush" Value="#0a1451"/>

<Setter Property="FontSize" Value="28"/>

<Setter Property="Foreground" Value="#FFFFFF"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=".15\*"/>

<RowDefinition Height=".7\*"/>

<RowDefinition Height=".15\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Background="#0A1451" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="КУРАТОРЫ" Click="OpenCurators"/>

<Button Content="СТУДЕНТЫ" Click="OpenStudents"/>

<Button Content="ПРЕПОДАВАТЕЛИ" Click="OpenTeachers"/>

<Button Content="РОДИТЕЛИ" Click="OpenParents"/>

<Button Content="ЗАЧЁТКИ" Click="OpenLedgers"/>

</StackPanel>

<Canvas Grid.Row="1">

<TextBlock Name="StudentName" FontSize="36" FontWeight="Bold" Text="Бусов Владислав" Canvas.Left="230" Canvas.Top="90"/>

<Image Cursor="Hand" MouseUp="EditStudent" Source="Source/edit.png" Width="20" Height="20" Canvas.Left="204" Canvas.Top="104" HorizontalAlignment="Center" VerticalAlignment="Top" RenderTransformOrigin="0.6,0.117"/>

<TextBlock Name="TeacherName" FontSize="36" FontWeight="Bold" Text="+79196395387" Canvas.Left="233" Canvas.Top="130"/>

<Image Cursor="Hand" MouseUp="EditTeacher" Source="Source/edit.png" Width="20" Height="20" Canvas.Left="204" Canvas.Top="145" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBlock Name="SubjectName" FontSize="24" FontWeight="Light" Text="15.07.2006" Canvas.Left="236" Canvas.Top="182" HorizontalAlignment="Center" VerticalAlignment="Top"/>

<TextBlock Name="Semester" FontSize="24" FontWeight="Light" Text="4338" Canvas.Left="236" Canvas.Top="214" HorizontalAlignment="Center" VerticalAlignment="Top"/>

<Image Cursor="Hand" MouseUp="EditSemester" Source="Source/edit.png" Width="20" Height="20" Canvas.Left="204" Canvas.Top="220" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBlock Name="Grade" FontSize="24" FontWeight="Light" Text="Гимадеев И.Ф." Canvas.Left="236" Canvas.Top="246" HorizontalAlignment="Center" VerticalAlignment="Top"/>

<Image Cursor="Hand" MouseUp="EditGrade" Source="Source/edit.png" Width="20" Height="20" Canvas.Left="204" Canvas.Top="252" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock x:Name="TeacherName\_Копировать" FontSize="20" FontWeight="Bold" Canvas.Left="107" Canvas.Top="217" HorizontalAlignment="Left" VerticalAlignment="Center" Width="97"><Run Text="Семестр"/><Run Language="ru-ru" Text=":"/></TextBlock>

<TextBlock x:Name="TeacherName\_Копировать1" FontSize="20" FontWeight="Bold" Canvas.Left="107" Canvas.Top="245" Width="97" HorizontalAlignment="Center" VerticalAlignment="Top"><Run Language="ru-ru" Text="Оценка:"/></TextBlock>

<TextBlock x:Name="TeacherName\_Копировать4" FontSize="20" FontWeight="Bold" Canvas.Left="35" Canvas.Top="138" Width="169" HorizontalAlignment="Left" VerticalAlignment="Center" Text="Преподаватель:"/>

<TextBlock x:Name="TeacherName\_Копировать5" FontSize="20" FontWeight="Bold" Canvas.Left="35" Canvas.Top="104" Width="169" HorizontalAlignment="Center" VerticalAlignment="Top"><Run Language="ru-ru" Text="Студент:"/></TextBlock>

</Canvas>

<Rectangle Grid.Row="2">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Policy;

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.Shapes;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для LedgerDetail.xaml

/// </summary>

public partial class LedgerDetail : Window

{

public SummaryLedger ledger;

public Student student;

public Teacher teacher;

CuratorsEntities1 db;

public LedgerDetail(SummaryLedger ledger)

{

this.ledger = ledger;

this.db = CuratorsEntities1.GetContext();

this.student = db.Student.FirstOrDefault(s => s.StudentId == ledger.StudentId);

this.teacher = db.Teacher.FirstOrDefault(t => t.TeacherId == ledger.TeacherId);

InitializeComponent();

UpdateData();

}

public void UpdateData()

{

this.StudentName.Text = $"{student.FirstName} {student.LastName}";

this.TeacherName.Text = $"{teacher.FirstName} {teacher.LastName}";

this.SubjectName.Text = teacher.Subject;

this.Semester.Text = ledger.Semester.ToString();

this.Grade.Text = ledger.Grade.ToString();

}

private void OpenCurators(object sender, RoutedEventArgs e)

{

(new CuratorsWindow()).Show();

this.Close();

}

private void OpenStudents(object sender, RoutedEventArgs e)

{

(new UsersWindow()).Show();

this.Close();

}

private void OpenParents(object sender, RoutedEventArgs e)

{

(new ParentsWindow()).Show();

this.Close();

}

private void OpenTeachers(object sender, RoutedEventArgs e)

{

(new TeachersWindow()).Show();

this.Close();

}

private void OpenLedgers(object sender, RoutedEventArgs e)

{

(new LedgersWindows()).Show();

this.Close();

}

private void EditStudent(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите уникальный идентификатор студента");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (int.TryParse(new\_value, out int st\_id))

{

Student student = db.Student.FirstOrDefault(s => s.StudentId == st\_id);

if (student != null)

{

ledger.StudentId = st\_id;

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! Студента с Id {st\_id} не существует!");

}

else MessageBox.Show($"Ошибка! '{new\_value}' должно быть целым числом");

UpdateData();

}

private void EditTeacher(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите уникальный идентификатор преподавателя");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (int.TryParse(new\_value, out int teacher\_id))

{

Teacher teacher = db.Teacher.FirstOrDefault(t => t.TeacherId == teacher\_id);

if (teacher != null)

{

ledger.TeacherId = teacher\_id;

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! Преподавателя с Id {teacher\_id} не существует!");

}

else MessageBox.Show($"Ошибка! '{new\_value}' должно быть целым числом");

UpdateData();

}

private void EditSemester(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите семестр обучения одним числом");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (int.TryParse(new\_value, out int semester) && semester > 0)

{

ledger.Semester = semester;

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! Данные должны быть целым числом");

UpdateData();

}

private void EditGrade(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите оценку за предмет");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (int.TryParse(new\_value, out int grade))

{

ledger.Grade = grade.ToString();

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! Данные должны быть целым числом");

UpdateData();

}

}

}

<Window x:Class="WpfApp1.LedgersWindows"

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"

ResizeMode="NoResize"

Height="810" Width="1440"

Background="#ECF0F3"

WindowStartupLocation="CenterScreen"

Title="LedgersWindow">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Width" Value="290"/>

<Setter Property="Background" Value="#0a1451"/>

<Setter Property="BorderBrush" Value="#0a1451"/>

<Setter Property="FontSize" Value="28"/>

<Setter Property="Foreground" Value="#FFFFFF"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=".15\*"/>

<RowDefinition Height=".7\*"/>

<RowDefinition Height=".15\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Background="#0A1451" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="КУРАТОРЫ" Click="OpenCurators"/>

<Button Content="СТУДЕНТЫ" Click="OpenStudents"/>

<Button Content="ПРЕПОДАВАТЕЛИ" Click="OpenTeachers"/>

<Button Content="РОДИТЕЛИ" Click="OpenParents"/>

<Button Content="ЗАЧЁТКИ" Click="OpenLedgers"/>

</StackPanel>

<Canvas Grid.Row="1" Cursor="Hand">

<TextBlock

Canvas.Left="602"

Canvas.Top="27"

FontSize="48"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Foreground="#0A1451"><Run Language="ru-ru" Text="ЗАЧЁТКИ"/></TextBlock>

<ScrollViewer Width="850" Height="430" VerticalScrollBarVisibility="Visible" Canvas.Left="280" Canvas.Top="110">

<StackPanel Name="CardsList">

<WrapPanel Grid.Column="0" Width="800" Name="UserCard0">

<Canvas Background="#FFFFFF" Width="750" Height="115">

<TextBlock FontSize="16" Foreground="#0A1451" FontWeight="Bold" HorizontalAlignment="Left" Canvas.Left="19" VerticalAlignment="Center" Canvas.Top="10" Width="247"><Run Language="ru-ru" Text="Насыйбуллин Амир"/></TextBlock>

<TextBlock FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="19" VerticalAlignment="Center" Canvas.Top="30" Width="210"><Run Language="ru-ru" Text="Тарасов Данил"/></TextBlock>

<TextBlock TextWrapping="Wrap" FontSize="16" FontWeight="Regular" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="19" VerticalAlignment="Center" Canvas.Top="78" Height="23" Width="576"><Run Language="ru-ru" Text="Проектирование и разработка веб приложений"/></TextBlock>

<TextBlock TextAlignment="Right" FontSize="16" FontWeight="Bold" Foreground="#0A1451" Canvas.Left="554" Width="186" HorizontalAlignment="Left" VerticalAlignment="Center" Text="3" Canvas.Top="10"/>

<Image Source="Source/edit.png" Width="20" Height="20" Canvas.Left="675" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<Image Source="Source/trash.png" Width="20" Height="20" Canvas.Left="720" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Canvas.Effect>

<DropShadowEffect Opacity=".5" RenderingBias="Performance" Direction="225" ShadowDepth="6" BlurRadius="10"/>

</Canvas.Effect>

</Canvas>

</WrapPanel>

</StackPanel>

</ScrollViewer>

<Button Click="CreateLedger" Height="50" Width="50" Canvas.Left="1380" Canvas.Top="496" Foreground="#FFFFFF" FontSize="40">

<Button.Style>

<Style TargetType="Button">

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="true">

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

</Trigger>

</Style.Triggers>

</Style>

</Button.Style>

<Button.Template>

<ControlTemplate>

<Border CornerRadius="25" Background="#0A1451">

<ContentPresenter Content="+" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0,-5,0,0"/>

</Border>

</ControlTemplate>

</Button.Template>

</Button>

</Canvas>

<Rectangle Grid.Row="2">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

</Grid>

</Window>

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.Markup;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для LedgersWindows.xaml

/// </summary>

public partial class LedgersWindows : Window

{

public LedgersWindows()

{

InitializeComponent();

ShowData();

}

public void ShowData()

{

var db = CuratorsEntities1.GetContext();

List<SummaryLedger> ledgers = db.SummaryLedger.ToList();

if (ledgers.Count == 0) this.CardsList.Children.Clear();

else if (ledgers.Count > 1)

{

for (int i = 1; i < ledgers.Count; i++)

{

WrapPanel el = XamlReader.Parse(

XamlWriter.Save(

this.UserCard0

)

) as WrapPanel;

this.RegisterName($"UserCard{i}", el);

this.CardsList.Children.Add(el);

}

}

for (int i = 0; i < ledgers.Count; i++)

{

var ledger = ledgers[i];

int TBCounter = 0;

int ICounter = 0;

Canvas container = (this.FindName($"UserCard{i}") as WrapPanel).Children[0] as Canvas;

foreach (UIElement el in container.Children)

{

string val = "";

if (el.GetType().ToString() == "System.Windows.Controls.TextBlock")

{

Student student = db.Student.FirstOrDefault(s => s.StudentId == ledger.StudentId);

Teacher teacher = db.Teacher.FirstOrDefault(t => t.TeacherId == ledger.TeacherId);

if (TBCounter == 0) val = $"{student.FirstName} {student.LastName}";

else if (TBCounter == 1) val = $"{teacher.FirstName} {teacher.LastName}";

else if (TBCounter == 2) val = teacher.Subject;

else if (TBCounter == 3) val = $"{ledger.Semester} | {ledger.Grade}";

el.SetValue(TextBlock.TextProperty, val);

TBCounter++;

}

if (el.GetType().ToString() == "System.Windows.Controls.Image")

{

if (ICounter == 0)

{

el.MouseUp += (o, e) =>

{

var win = new LedgerDetail(ledger);

win.Show();

this.Close();

};

}

else if (ICounter == 1)

{

el.MouseUp += (o, e) =>

{

db.SummaryLedger.Remove(ledger);

db.SaveChanges();

var win = new LedgersWindows();

win.Show();

this.Close();

};

}

ICounter++;

}

}

}

}

private void OpenCurators(object sender, RoutedEventArgs e)

{

(new CuratorsWindow()).Show();

this.Close();

}

private void OpenStudents(object sender, RoutedEventArgs e)

{

(new UsersWindow()).Show();

this.Close();

}

private void OpenParents(object sender, RoutedEventArgs e)

{

(new ParentsWindow()).Show();

this.Close();

}

private void OpenTeachers(object sender, RoutedEventArgs e)

{

(new TeachersWindow()).Show();

this.Close();

}

private void OpenLedgers(object sender, RoutedEventArgs e)

{

(new LedgersWindows()).Show();

this.Close();

}

private void CreateLedger(object sender, RoutedEventArgs e)

{

var db = CuratorsEntities1.GetContext();

SummaryLedger ledger = new SummaryLedger()

{

Grade = "0",

Semester = 0,

TeacherId = db.Teacher.First().TeacherId,

StudentId = db.Student.First().StudentId,

};

db.SummaryLedger.Add(ledger);

db.SaveChanges();

var win = new LedgerDetail(ledger);

win.Show();

this.Close();

}

}

}

<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"

ResizeMode="NoResize"

Title="MainWindow" Height="810" Width="1440"

WindowStartupLocation="CenterScreen"

Background="#ECF0F3">

<Window.Resources>

<Style TargetType="TextBox">

<Setter Property="Width" Value="300"/>

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

<Setter Property="Margin" Value="0 0 0 10"/>

<Setter Property="Foreground" Value="#0A1451"/>

</Style>

<Style TargetType="Button">

<Setter Property="FontSize" Value="20"/>

<Setter Property="Width" Value="300"/>

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

<Setter Property="Margin" Value="0 20 0 0"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=".15\*"/>

<RowDefinition Height=".7\*"/>

<RowDefinition Height=".15\*"/>

</Grid.RowDefinitions>

<Rectangle Grid.Row="0">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

<WrapPanel Orientation="Vertical" Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Center">

<TextBlock

Text="ВХОД В СИСТЕМУ"

FontSize="64"

Foreground="#0A1451"

Margin="0 0 0 100"/>

<TextBox Name="Login" TabIndex="1">

<TextBox.Template>

<ControlTemplate>

<Border CornerRadius="8" Padding="2" BorderThickness="1" BorderBrush="#000000">

<ScrollViewer x:Name="PART\_ContentHost" Background="#FFFFFF" Foreground="#0A1451"/>

</Border>

</ControlTemplate>

</TextBox.Template>

</TextBox>

<TextBox Name="Password" TabIndex="2">

<TextBox.Template>

<ControlTemplate>

<Border CornerRadius="8" Padding="2" BorderThickness="1" BorderBrush="#000000">

<ScrollViewer x:Name="PART\_ContentHost" Background="#FFFFFF" Foreground="#0A1451"/>

</Border>

</ControlTemplate>

</TextBox.Template>

</TextBox>

<Button Click="AuthenticateUser" Content="Войти" Background="#1E2D85" Foreground="#FFFFFF"/>

</WrapPanel>

<Rectangle Grid.Row="2">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Policy;

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

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

//var win = new UsersWindow();

//win.Show();

//

//var win = new CuratorsWindow();

//win.Show();

//

//var win = new ParentsWindow();

//win.Show();

//

//var win = new TeachersWindow();

//win.Show();

//var win = new Window1();

//win.Show();

}

private void AuthenticateUser(object sender, EventArgs e)

{

string login = this.Login.Text;

string psswd = this.Password.Text;

if (login == "admin" && psswd == "admin")

{

(new UsersWindow()).Show();

this.Close();

}

}

}

}

//------------------------------------------------------------------------------

// <auto-generated>

// Этот код создан по шаблону.

//

// Изменения, вносимые в этот файл вручную, могут привести к непредвиденной работе приложения.

// Изменения, вносимые в этот файл вручную, будут перезаписаны при повторном создании кода.

// </auto-generated>

//------------------------------------------------------------------------------

namespace WpfApp1

{

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

public partial class CuratorsEntities1 : DbContext

{

public static CuratorsEntities1 db;

public CuratorsEntities1()

: base("name=CuratorsEntities11")

{

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

public static CuratorsEntities1 GetContext()

{

if (db == null)

{

db = new CuratorsEntities1();

}

return db;

}

public virtual DbSet<Curator> Curator { get; set; }

public virtual DbSet<HealtCard> HealtCard { get; set; }

public virtual DbSet<Parent> Parent { get; set; }

public virtual DbSet<Student> Student { get; set; }

public virtual DbSet<Student\_Parent\_Relationship> Student\_Parent\_Relationship { get; set; }

public virtual DbSet<SummaryLedger> SummaryLedger { get; set; }

public virtual DbSet<Teacher> Teacher { get; set; }

}

}

<Window x:Class="WpfApp1.NewValueWindow"

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"

ResizeMode="NoResize"

Height="600" Width="400"

WindowStartupLocation="CenterScreen"

Background="#ECF0F3"

Title="NewValueWindow">

<Canvas>

<TextBlock Text="Новое значение" FontSize="38" FontWeight="Bold" Canvas.Left="57" Canvas.Top="38"/>

<TextBox Name="ValueTB" Width="300" Height="200" Canvas.Left="50" Canvas.Top="192" FontSize="24"/>

<Button Content="Сохранить" FontSize="24" Background="#0A1451" Width="300" Height="40" Foreground="#FFFFFF" Canvas.Left="50" Canvas.Top="488" Click="Button\_Click"/>

<TextBlock TextWrapping="Wrap" Name="Description" Text="Новое значение" FontSize="20" Canvas.Left="57" FontWeight="Light" Canvas.Top="94" HorizontalAlignment="Left" VerticalAlignment="Top" Height="93" Width="293"/>

</Canvas>

</Window>

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 WpfApp1

{

/// <summary>

/// Логика взаимодействия для NewValueWindow.xaml

/// </summary>

public partial class NewValueWindow : Window

{

public string new\_value;

public NewValueWindow(string desc)

{

InitializeComponent();

this.Description.Text = desc;

}

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

{

new\_value = this.ValueTB.Text;

this.Close();

}

}

}

<Window x:Class="WpfApp1.ParentDetail"

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"

ResizeMode="NoResize"

Height="810" Width="1440"

Background="#ECF0F3"

WindowStartupLocation="CenterScreen"

Title="ParentDetail">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Width" Value="290"/>

<Setter Property="Background" Value="#0a1451"/>

<Setter Property="BorderBrush" Value="#0a1451"/>

<Setter Property="FontSize" Value="28"/>

<Setter Property="Foreground" Value="#FFFFFF"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=".15\*"/>

<RowDefinition Height=".7\*"/>

<RowDefinition Height=".15\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Background="#0A1451" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="КУРАТОРЫ" Click="OpenCurators"/>

<Button Content="СТУДЕНТЫ" Click="OpenStudents"/>

<Button Content="ПРЕПОДАВАТЕЛИ" Click="OpenTeachers"/>

<Button Content="РОДИТЕЛИ" Click="OpenParents"/>

<Button Content="ЗАЧЁТКИ"/>

</StackPanel>

<Canvas Grid.Row="1">

<Image Source="Source/avatar guest.png" Height="416" Width="433"/>

<TextBlock Name="Name" FontSize="36" FontWeight="Bold" Text="Бусов Владислав" Canvas.Left="721" Canvas.Top="97" HorizontalAlignment="Left" VerticalAlignment="Center" Width="470"/>

<Image Cursor="Hand" MouseUp="EditName" Source="Source/edit.png" Width="22" Height="20" Canvas.Left="698" Canvas.Top="111" HorizontalAlignment="Left" VerticalAlignment="Center" RenderTransformOrigin="0.6,0.117"/>

<TextBlock Name="Phone" FontSize="24" FontWeight="Light" Text="+79196395387" Canvas.Left="726" Canvas.Top="150" HorizontalAlignment="Left" VerticalAlignment="Center" Width="230"/>

<Image Cursor="Hand" MouseUp="EditPhone" Source="Source/edit.png" Width="22" Height="20" Canvas.Left="698" Canvas.Top="162" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock Name="Email" FontSize="24" FontWeight="Light" Text="15.07.2006" Canvas.Left="730" Canvas.Top="182" HorizontalAlignment="Left" VerticalAlignment="Center" Width="165"/>

<Image Cursor="Hand" MouseUp="EditEmail" Source="Source/edit.png" Width="22" Height="20" Canvas.Left="698" Canvas.Top="188" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock Name="Workplace" FontSize="24" FontWeight="Light" Text="4338" Canvas.Left="730" Canvas.Top="214" HorizontalAlignment="Left" VerticalAlignment="Center" Width="79"/>

<Image Cursor="Hand" MouseUp="EditWorkplace" Source="Source/edit.png" Width="22" Height="20" Canvas.Left="698" Canvas.Top="220" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock x:Name="Name\_Копировать" FontSize="20" FontWeight="Bold" Canvas.Left="610" Canvas.Top="150" Width="83" HorizontalAlignment="Left" VerticalAlignment="Center"><Run Language="ru-ru" Text="Номер:"/></TextBlock>

<TextBlock x:Name="Name\_Копировать1" FontSize="20" FontWeight="Bold" Canvas.Left="610" Canvas.Top="185" Width="72" HorizontalAlignment="Left" VerticalAlignment="Center"><Run Text="Почта"/><Run Language="ru-ru" Text=":"/></TextBlock>

<TextBlock x:Name="Name\_Копировать2" FontSize="20" FontWeight="Bold" Canvas.Left="610" Canvas.Top="214" Width="74" HorizontalAlignment="Left" VerticalAlignment="Center"><Run Language="ru-ru" Text="Группа:"/></TextBlock>

</Canvas>

<Rectangle Grid.Row="2">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

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.Shapes;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для ParentDetail.xaml

/// </summary>

public partial class ParentDetail : Window

{

public Parent parent;

CuratorsEntities1 db;

public ParentDetail(Parent parent)

{

this.parent = parent;

this.db = CuratorsEntities1.GetContext();

InitializeComponent();

UpdateData();

}

public void UpdateData()

{

Name.Text = $"{parent.FirstName} {parent.LastName}";

Phone.Text = parent.CellPhone;

Email.Text = parent.Email;

Workplace.Text = parent.Workplace;

}

private void OpenCurators(object sender, RoutedEventArgs e)

{

(new CuratorsWindow()).Show();

this.Close();

}

private void OpenStudents(object sender, RoutedEventArgs e)

{

(new UsersWindow()).Show();

this.Close();

}

private void OpenParents(object sender, RoutedEventArgs e)

{

(new ParentsWindow()).Show();

this.Close();

}

private void OpenTeachers(object sender, RoutedEventArgs e)

{

(new TeachersWindow()).Show();

this.Close();

}

private void EditName(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите данные в формате Имя Фамилия");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.Length != 0 && new\_value.Split(' ').Length == 2)

{

parent.FirstName = new\_value.Split(' ')[0];

parent.LastName = new\_value.Split(' ')[1];

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! '{new\_value}' должно состоять из двух слов: Имя Фамилия, разделенных пробелом");

UpdateData();

}

private void EditPhone(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите номер телефона в формате +71231231231");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.StartsWith("+7") && new\_value.Length == 12 && long.TryParse(new\_value.Replace("+", ""), out \_))

{

parent.CellPhone = new\_value;

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! '{new\_value}' должен начинаться с +7 и состоять из 11 цифр");

UpdateData();

}

private void EditEmail(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите электронную почту в формате test@test.test");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.Contains("@") && new\_value.Contains("."))

{

parent.Email = new\_value.ToString();

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! '{new\_value}' не подходит под требования к формату почты");

UpdateData();

}

private void EditWorkplace(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите название места работы родителя");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.Length > 0)

{

parent.Workplace = new\_value;

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка!");

UpdateData();

}

}

}

<Window x:Class="WpfApp1.ParentsWindow"

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"

ResizeMode="NoResize"

Height="810" Width="1440"

Background="#ECF0F3"

WindowStartupLocation="CenterScreen"

Title="ParentsWindow">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Width" Value="290"/>

<Setter Property="Background" Value="#0a1451"/>

<Setter Property="BorderBrush" Value="#0a1451"/>

<Setter Property="FontSize" Value="28"/>

<Setter Property="Foreground" Value="#FFFFFF"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=".15\*"/>

<RowDefinition Height=".7\*"/>

<RowDefinition Height=".15\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Background="#0A1451" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="КУРАТОРЫ" Click="OpenCurators"/>

<Button Content="СТУДЕНТЫ" Click="OpenStudents"/>

<Button Content="ПРЕПОДАВАТЕЛИ" Click="OpenTeachers"/>

<Button Content="РОДИТЕЛИ" Click="OpenParents"/>

<Button Content="ЗАЧЁТКИ" Click="OpenLedgers"/>

</StackPanel>

<Canvas Grid.Row="1" Cursor="Hand">

<TextBlock Text="РОДИТЕЛИ"

Canvas.Left="602"

Canvas.Top="27"

FontSize="48"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Foreground="#0A1451"/>

<ScrollViewer Width="860" VerticalScrollBarVisibility="Visible" Canvas.Left="280" Canvas.Top="110" Height="463">

<StackPanel Name="CardsList">

<WrapPanel Grid.Column="0" Width="800" Name="UserCard0">

<Canvas Background="#FFFFFF" Width="750" Height="115">

<Image Source="Source/avatar guest.png" Height="87" Width="79" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBlock Text="Владислав Бусов" FontSize="16" Foreground="#0A1451" FontWeight="Bold" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="8"/>

<TextBlock Text="15.07.2006" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="29"/>

<TextBlock Text="+79196395387" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="47"/>

<TextBlock Text="Место работы: Гимадеев И.Ф." FontSize="16" FontWeight="Bold" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="10" VerticalAlignment="Top" Canvas.Top="84"/>

<Image Source="Source/edit.png" Width="20" Height="20" Canvas.Left="695" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<Image Source="Source/trash.png" Width="20" Height="20" Canvas.Left="720" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Canvas.Effect>

<DropShadowEffect Opacity=".5" RenderingBias="Performance" Direction="225" ShadowDepth="6" BlurRadius="10"/>

</Canvas.Effect>

</Canvas>

</WrapPanel>

</StackPanel>

</ScrollViewer>

<Button Click="CreateParent" Height="50" Width="50" Canvas.Left="1380" Canvas.Top="496" Foreground="#FFFFFF" FontSize="40">

<Button.Style>

<Style TargetType="Button">

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="true">

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

</Trigger>

</Style.Triggers>

</Style>

</Button.Style>

<Button.Template>

<ControlTemplate>

<Border CornerRadius="25" Background="#0A1451">

<ContentPresenter Content="+" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0,-5,0,0"/>

</Border>

</ControlTemplate>

</Button.Template>

</Button>

</Canvas>

<Rectangle Grid.Row="2">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

</Grid>

</Window>

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.Markup;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для ParentsWindow.xaml

/// </summary>

public partial class ParentsWindow : Window

{

public ParentsWindow()

{

InitializeComponent();

ShowData();

}

public void ShowData()

{

var db = CuratorsEntities1.GetContext();

List<Parent> parents = db.Parent.ToList();

if (parents.Count == 0) this.CardsList.Children.Clear();

else if (parents.Count > 1)

{

for (int i = 1; i < parents.Count; i++)

{

WrapPanel el = XamlReader.Parse(

XamlWriter.Save(

this.UserCard0

)

) as WrapPanel;

this.RegisterName($"UserCard{i}", el);

this.CardsList.Children.Add(el);

}

}

for (int i = 0; i < parents.Count; i++)

{

var parent = parents[i];

int TBCounter = 0;

int ICounter = 0;

Canvas container = (this.FindName($"UserCard{i}") as WrapPanel).Children[0] as Canvas;

foreach (UIElement el in container.Children)

{

string val = "";

if (el.GetType().ToString() == "System.Windows.Controls.TextBlock")

{

if (TBCounter == 0) val = $"{parent.FirstName} {parent.LastName}";

else if (TBCounter == 1) val = parent.CellPhone;

else if (TBCounter == 2) val = parent.Email;

else if (TBCounter == 3) val = $"Место работы: {parent.Workplace}";

el.SetValue(TextBlock.TextProperty, val);

TBCounter++;

}

if (el.GetType().ToString() == "System.Windows.Controls.Image")

{

if (ICounter == 1)

{

el.MouseUp += (o, e) =>

{

var win = new ParentDetail(parent);

win.Show();

this.Close();

};

}

else if (ICounter == 2)

{

el.MouseUp += (o, e) =>

{

db.Parent.Remove(parent);

db.SaveChanges();

var win = new ParentsWindow();

win.Show();

this.Close();

};

}

ICounter++;

}

}

}

}

private void OpenCurators(object sender, RoutedEventArgs e)

{

(new CuratorsWindow()).Show();

this.Close();

}

private void OpenStudents(object sender, RoutedEventArgs e)

{

(new UsersWindow()).Show();

this.Close();

}

private void OpenParents(object sender, RoutedEventArgs e)

{

(new ParentsWindow()).Show();

this.Close();

}

private void OpenTeachers(object sender, RoutedEventArgs e)

{

(new TeachersWindow()).Show();

this.Close();

}

private void OpenLedgers(object sender, RoutedEventArgs e)

{

(new LedgersWindows()).Show();

this.Close();

}

private void CreateParent(object sender, RoutedEventArgs e)

{

var db = CuratorsEntities1.GetContext();

Parent parent = new Parent()

{

FirstName = "\_\_\_\_\_\_\_\_\_\_",

LastName = "\_\_\_\_\_\_\_\_\_\_",

CellPhone = "+7\_\_\_\_\_\_\_\_\_\_",

Email = "\_\_\_@\_\_\_.\_\_",

Workplace = "\_\_\_\_\_"

};

db.Parent.Add(parent);

db.SaveChanges();

var win = new ParentDetail(parent);

win.Show();

this.Close();

}

}

}

<Window x:Class="WpfApp1.TeacherDetail"

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"

ResizeMode="NoResize"

Height="810" Width="1440"

Background="#ECF0F3"

WindowStartupLocation="CenterScreen"

Title="TeacherDetail">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Width" Value="290"/>

<Setter Property="Background" Value="#0a1451"/>

<Setter Property="BorderBrush" Value="#0a1451"/>

<Setter Property="FontSize" Value="28"/>

<Setter Property="Foreground" Value="#FFFFFF"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=".15\*"/>

<RowDefinition Height=".7\*"/>

<RowDefinition Height=".15\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Background="#0A1451" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="КУРАТОРЫ" Click="OpenCurators"/>

<Button Content="СТУДЕНТЫ" Click="OpenStudents"/>

<Button Content="ПРЕПОДАВАТЕЛИ" Click="OpenTeachers"/>

<Button Content="РОДИТЕЛИ" Click="OpenParents"/>

<Button Content="ЗАЧЁТКИ"/>

</StackPanel>

<Canvas Grid.Row="1">

<Image Source="Source/avatar guest.png" Height="416" Width="433"/>

<TextBlock Name="Name" FontSize="36" FontWeight="Bold" Text="Бусов Владислав" Canvas.Left="598" Canvas.Top="97" HorizontalAlignment="Left" VerticalAlignment="Center" Width="448"/>

<Image Cursor="Hand" MouseUp="EditName" Source="Source/edit.png" Width="20" Height="20" Canvas.Left="577" Canvas.Top="111" HorizontalAlignment="Left" VerticalAlignment="Center" RenderTransformOrigin="0.6,0.117"/>

<TextBlock x:Name="Subject" FontSize="24" FontWeight="Light" Canvas.Left="602" Canvas.Top="150" HorizontalAlignment="Left" VerticalAlignment="Center" Width="184"><Run Language="ru-ru" Text="Математика"/></TextBlock>

<Image Cursor="Hand" MouseUp="EditSubject" Source="Source/edit.png" Width="20" Height="20" Canvas.Left="577" Canvas.Top="162" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock x:Name="Subject\_Копировать" FontSize="20" FontWeight="Bold" Canvas.Left="470" Canvas.Top="154" Width="107" HorizontalAlignment="Center" VerticalAlignment="Top"><Run Text="Предмет"/><Run Language="ru-ru" Text=":"/></TextBlock>

</Canvas>

<Rectangle Grid.Row="2">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Policy;

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.Shapes;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для TeacherDetail.xaml

/// </summary>

public partial class TeacherDetail : Window

{

public Teacher teacher;

CuratorsEntities1 db;

public TeacherDetail(Teacher teacher)

{

this.teacher = teacher;

this.db = CuratorsEntities1.GetContext();

InitializeComponent();

UpdateData();

}

public void UpdateData()

{

Name.Text = $"{teacher.FirstName} {teacher.LastName}";

Subject.Text = teacher.Subject;

}

private void OpenCurators(object sender, RoutedEventArgs e)

{

(new CuratorsWindow()).Show();

this.Close();

}

private void OpenStudents(object sender, RoutedEventArgs e)

{

(new UsersWindow()).Show();

this.Close();

}

private void OpenParents(object sender, RoutedEventArgs e)

{

(new ParentsWindow()).Show();

this.Close();

}

private void OpenTeachers(object sender, RoutedEventArgs e)

{

(new TeachersWindow()).Show();

this.Close();

}

private void EditName(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите данные в формате Имя Фамилия");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.Length != 0 && new\_value.Split(' ').Length == 2)

{

teacher.FirstName = new\_value.Split(' ')[0];

teacher.LastName = new\_value.Split(' ')[1];

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! '{new\_value}' должно состоять из двух слов: Имя Фамилия, разделенных пробелом");

UpdateData();

}

private void EditSubject(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите название предмета одной строкой");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.Length != 0)

{

teacher.Subject = new\_value;

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! Пустое значение");

UpdateData();

}

}

}

<Window x:Class="WpfApp1.TeachersWindow"

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"

ResizeMode="NoResize"

Height="810" Width="1440"

Background="#ECF0F3"

WindowStartupLocation="CenterScreen"

Title="TeachersWindow">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Width" Value="290"/>

<Setter Property="Background" Value="#0a1451"/>

<Setter Property="BorderBrush" Value="#0a1451"/>

<Setter Property="FontSize" Value="28"/>

<Setter Property="Foreground" Value="#FFFFFF"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=".15\*"/>

<RowDefinition Height=".7\*"/>

<RowDefinition Height=".15\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Background="#0A1451" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="КУРАТОРЫ" Click="OpenCurators"/>

<Button Content="СТУДЕНТЫ" Click="OpenStudents"/>

<Button Content="ПРЕПОДАВАТЕЛИ" Click="OpenTeachers"/>

<Button Content="РОДИТЕЛИ" Click="OpenParents"/>

<Button Content="ЗАЧЁТКИ" Click="OpenLedgers"/>

</StackPanel>

<Canvas Grid.Row="1">

<TextBlock Text="ПРЕПОДАВАТЕЛИ"

Canvas.Left="522"

Canvas.Top="41"

FontSize="48"

HorizontalAlignment="Left"

VerticalAlignment="Top"

Foreground="#0A1451"/>

<ScrollViewer Height="440" VerticalScrollBarVisibility="Visible" Canvas.Left="322" Canvas.Top="110" HorizontalAlignment="Left" VerticalAlignment="Center" Width="860">

<StackPanel Name="CardsList">

<WrapPanel Grid.Column="0" Width="800" Name="UserCard0">

<Canvas Background="#FFFFFF" Width="800" Height="115">

<Image Source="Source/avatar guest.png" Height="87" Width="79" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBlock Text="Владислав Бусов" FontSize="16" Foreground="#0A1451" FontWeight="Bold" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="8"/>

<TextBlock Text="Куратор: Гимадеев И.Ф." FontSize="16" FontWeight="Normal" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="40"/>

<Image Source="Source/edit.png" Width="20" Height="20" Canvas.Left="695" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<Image Source="Source/trash.png" Width="20" Height="20" Canvas.Left="720" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Canvas.Effect>

<DropShadowEffect Opacity=".5" RenderingBias="Performance" Direction="225" ShadowDepth="6" BlurRadius="10"/>

</Canvas.Effect>

</Canvas>

</WrapPanel>

</StackPanel>

</ScrollViewer>

<Button Click="CreateTeacher" Height="50" Width="50" Canvas.Left="1380" Canvas.Top="496" Foreground="#FFFFFF" FontSize="40">

<Button.Style>

<Style TargetType="Button">

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="true">

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

</Trigger>

</Style.Triggers>

</Style>

</Button.Style>

<Button.Template>

<ControlTemplate>

<Border CornerRadius="25" Background="#0A1451">

<ContentPresenter Content="+" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0,-5,0,0"/>

</Border>

</ControlTemplate>

</Button.Template>

</Button>

</Canvas>

<Rectangle Grid.Row="2">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

</Grid>

</Window>

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.Markup;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для TeachersWindow.xaml

/// </summary>

public partial class TeachersWindow : Window

{

public TeachersWindow()

{

InitializeComponent();

ShowData();

}

public void ShowData()

{

var db = CuratorsEntities1.GetContext();

List<Teacher> teachers = db.Teacher.ToList();

if (teachers.Count == 0) this.CardsList.Children.Clear();

else if (teachers.Count > 1)

{

for (int i = 1; i < teachers.Count; i++)

{

WrapPanel el = XamlReader.Parse(

XamlWriter.Save(

this.UserCard0

)

) as WrapPanel;

//el.SetValue(Grid.ColumnProperty, i % 3);

//el.SetValue(Grid.RowProperty, i / 3);

this.RegisterName($"UserCard{i}", el);

this.CardsList.Children.Add(el);

}

}

for (int i = 0; i < teachers.Count; i++)

{

var teacher = teachers[i];

int TBCounter = 0;

int ICounter = 0;

Canvas container = (this.FindName($"UserCard{i}") as WrapPanel).Children[0] as Canvas;

foreach (UIElement el in container.Children)

{

string val = "";

if (el.GetType().ToString() == "System.Windows.Controls.TextBlock")

{

if (TBCounter == 0) val = $"{teacher.FirstName} {teacher.LastName}";

else if (TBCounter == 1) val = teacher.Subject;

el.SetValue(TextBlock.TextProperty, val);

TBCounter++;

}

if (el.GetType().ToString() == "System.Windows.Controls.Image")

{

if (ICounter == 1)

{

el.MouseUp += (o, e) =>

{

var win = new TeacherDetail(teacher);

win.Show();

this.Close();

};

}

else if (ICounter == 2)

{

el.MouseUp += (o, e) =>

{

db.Teacher.Remove(teacher);

db.SaveChanges();

var win = new TeachersWindow();

win.Show();

this.Close();

};

}

ICounter++;

}

}

}

}

private void OpenCurators(object sender, RoutedEventArgs e)

{

(new CuratorsWindow()).Show();

this.Close();

}

private void OpenStudents(object sender, RoutedEventArgs e)

{

(new UsersWindow()).Show();

this.Close();

}

private void OpenParents(object sender, RoutedEventArgs e)

{

(new ParentsWindow()).Show();

this.Close();

}

private void OpenTeachers(object sender, RoutedEventArgs e)

{

(new TeachersWindow()).Show();

this.Close();

}

private void OpenLedgers(object sender, RoutedEventArgs e)

{

(new LedgersWindows()).Show();

this.Close();

}

private void CreateTeacher(object sender, RoutedEventArgs e)

{

var db = CuratorsEntities1.GetContext();

Teacher teacher = new Teacher()

{

FirstName = "\_\_\_\_\_\_\_\_\_\_",

LastName = "\_\_\_\_\_\_\_\_\_\_",

Subject = "\_\_\_\_\_"

};

db.Teacher.Add(teacher);

db.SaveChanges();

var win = new TeacherDetail(teacher);

win.Show();

this.Close();

}

}

}

<Window x:Class="WpfApp1.UserDetail"

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"

ResizeMode="NoResize"

Height="810" Width="1440"

Background="#ECF0F3"

WindowStartupLocation="CenterScreen"

Title="UserDetail">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Width" Value="290"/>

<Setter Property="Background" Value="#0a1451"/>

<Setter Property="BorderBrush" Value="#0a1451"/>

<Setter Property="FontSize" Value="28"/>

<Setter Property="Foreground" Value="#FFFFFF"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=".15\*"/>

<RowDefinition Height=".7\*"/>

<RowDefinition Height=".15\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Background="#0A1451" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="КУРАТОРЫ" Click="OpenCurators"/>

<Button Content="СТУДЕНТЫ" Click="OpenStudents"/>

<Button Content="ПРЕПОДАВАТЕЛИ" Click="OpenTeachers"/>

<Button Content="РОДИТЕЛИ" Click="OpenParents"/>

<Button Content="ЗАЧЁТКИ"/>

</StackPanel>

<Canvas Grid.Row="1">

<Image Source="Source/avatar guest.png" Height="416" Width="394" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock Name="Name" FontSize="36" FontWeight="Bold" Text="Бусов Владислав" Canvas.Left="430" Canvas.Top="97"/>

<Image Cursor="Hand" MouseUp="EditName" Source="Source/edit.png" Width="20" Height="20" Canvas.Left="404" Canvas.Top="111" HorizontalAlignment="Center" VerticalAlignment="Top" RenderTransformOrigin="0.6,0.117"/>

<TextBlock Name="Phone" FontSize="24" FontWeight="Light" Text="+79196395387" Canvas.Left="571" Canvas.Top="150" HorizontalAlignment="Left" VerticalAlignment="Center" Width="320"/>

<Image Cursor="Hand" MouseUp="EditPhone" Source="Source/edit.png" Width="21" Height="20" Canvas.Left="538" Canvas.Top="162" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock Name="BDate" FontSize="24" FontWeight="Light" Text="15.07.2006" Canvas.Left="578" Canvas.Top="182" HorizontalAlignment="Left" VerticalAlignment="Center" Width="229"/>

<Image Cursor="Hand" MouseUp="EditBDate" Source="Source/edit.png" Width="21" Height="20" Canvas.Left="538" Canvas.Top="188" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock Name="Group" FontSize="24" FontWeight="Light" Text="4338" Canvas.Left="578" Canvas.Top="214" HorizontalAlignment="Left" VerticalAlignment="Center" Width="108"/>

<Image Cursor="Hand" MouseUp="EditGroup" Source="Source/edit.png" Width="21" Height="20" Canvas.Left="538" Canvas.Top="220" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock Name="Curator" FontSize="24" FontWeight="Light" Text="Гимадеев И.Ф." Canvas.Left="578" Canvas.Top="246" HorizontalAlignment="Left" VerticalAlignment="Center" Width="323"/>

<Image Cursor="Hand" MouseUp="EditCurator" Source="Source/edit.png" Width="21" Height="20" Canvas.Left="538" Canvas.Top="252" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock Name="Parent" FontSize="24" FontWeight="Light" Text="Гимадеев И.Ф." Canvas.Left="578" Canvas.Top="278" Width="712" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<Image Cursor="Hand" MouseUp="EditParent" Source="/Source/edit.png" Width="21" Height="20" Canvas.Left="538" Canvas.Top="285" HorizontalAlignment="Center" VerticalAlignment="Center"/>

<TextBlock x:Name="Phone\_Копировать" FontSize="20" FontWeight="Bold" Canvas.Left="424" Canvas.Top="151" Width="114" HorizontalAlignment="Center" VerticalAlignment="Center"><Run Language="ru-ru" Text="Телефон:"/></TextBlock>

<TextBlock x:Name="Phone\_Копировать1" FontSize="20" FontWeight="Bold" Canvas.Left="424" Canvas.Top="182" Width="114" HorizontalAlignment="Center" VerticalAlignment="Top"><Run Language="ru-ru" Text="Д. Р. :"/></TextBlock>

<TextBlock x:Name="Phone\_Копировать2" FontSize="20" FontWeight="Bold" Canvas.Left="424" Canvas.Top="215" Width="114" HorizontalAlignment="Center" VerticalAlignment="Top"><Run Language="ru-ru" Text="Группа:"/></TextBlock>

<TextBlock x:Name="Phone\_Копировать3" FontSize="20" FontWeight="Bold" Canvas.Left="424" Canvas.Top="245" Width="114" HorizontalAlignment="Center" VerticalAlignment="Top"><Run Language="ru-ru" Text="Куратор:"/></TextBlock>

<TextBlock x:Name="Phone\_Копировать4" FontSize="20" FontWeight="Bold" Canvas.Left="424" Canvas.Top="280" Width="114" HorizontalAlignment="Center" VerticalAlignment="Top"><Run Language="ru-ru" Text="Родители"/><Run Text=":"/></TextBlock>

</Canvas>

<Rectangle Grid.Row="2">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Data.Entity.Migrations.Model;

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 WpfApp1

{

/// <summary>

/// Логика взаимодействия для UserDetail.xaml

/// </summary>

public partial class UserDetail : Window

{

public Student student;

CuratorsEntities1 db;

public UserDetail(Student student)

{

this.student = student;

this.db = CuratorsEntities1.GetContext();

InitializeComponent();

UpdateData();

}

public void UpdateData()

{

Name.Text = $"{student.FirstName} {student.LastName}";

Phone.Text = student.CellPhone;

Group.Text = student.GroupNumber;

BDate.Text = student.BDate;

var parents\_list = db.Student\_Parent\_Relationship.Where(spr => spr.StudentId == student.StudentId);

string parents = "";

foreach (var spr in parents\_list)

{

var parent = db.Parent.FirstOrDefault(p => p.ParentId == spr.ParentId);

parents += $"{parent.FirstName} {parent.LastName}, ";

}

this.Parent.Text = parents;

Curator curator = db.Curator.FirstOrDefault(c => c.CuratorID == student.CuratorId);

Curator.Text = $"{curator.FirstName} {curator.LastName}";

}

private void OpenCurators(object sender, RoutedEventArgs e)

{

(new CuratorsWindow()).Show();

this.Close();

}

private void OpenStudents(object sender, RoutedEventArgs e)

{

(new UsersWindow()).Show();

this.Close();

}

private void OpenParents(object sender, RoutedEventArgs e)

{

(new ParentsWindow()).Show();

this.Close();

}

private void OpenTeachers(object sender, RoutedEventArgs e)

{

(new TeachersWindow()).Show();

this.Close();

}

private void EditName(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите данные в формате Имя Фамилия");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.Length != 0 && new\_value.Split(' ').Length == 2)

{

student.FirstName = new\_value.Split(' ')[0];

student.LastName = new\_value.Split(' ')[1];

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! '{new\_value}' должно состоять из двух слов: Имя Фамилия, разделенных пробелом");

UpdateData();

}

private void EditPhone(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите номер телефона в формате +71231231231");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.StartsWith("+7") && new\_value.Length == 12 && long.TryParse(new\_value.Replace("+", ""), out \_))

{

student.CellPhone = new\_value;

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! '{new\_value}' должен начинаться с +7 и состоять из 11 цифр");

UpdateData();

}

private void EditBDate(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите дату рождения в формате дд.мм.гггг");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (DateTime.TryParse(new\_value, out DateTime date))

{

student.BDate = new\_value.ToString();

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! '{new\_value}' не подходит под требования к формату данных");

UpdateData();

}

private void EditGroup(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите группу числом из четырех цифр");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

if (new\_value.Length == 4 && int.TryParse(new\_value, out int group) && group > 0)

{

student.GroupNumber = new\_value;

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! Группа '{new\_value}' должна состоять из четырех цифр. Пример: 4338");

UpdateData();

}

private void EditCurator(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите уникальный идентификатор куратора (одно целое число)");

win.ShowDialog();

string new\_value = win.new\_value;

if (new\_value == null) return;

string fname = new\_value.Split(' ')[0];

string lname = new\_value.Split(' ')[1];

if (new\_value.Split(' ').Length == 2)

{

Curator curator = db.Curator.FirstOrDefault(

c => c.LastName == lname &&

c.FirstName == fname);

if (curator != null)

{

student.CuratorId = curator.CuratorID;

db.SaveChanges();

MessageBox.Show("Новые данные успешно сохранены");

}

else MessageBox.Show($"Ошибка! Куратор с именем {new\_value.Split(' ')[0]} и фамилией {new\_value.Split(' ')[1]} не найден!");

}

else MessageBox.Show($"Ошибка! Введите имя и фамилию куратора через пробел");

UpdateData();

}

private void EditParent(object sender, RoutedEventArgs e)

{

var win = new NewValueWindow("Введите уникальные идентификаторы родителей через пробел");

win.ShowDialog();

string new\_value = win.new\_value;

var ids = new\_value.Split(' ');

if (ids.Length == 2 || ids.Length == 1)

{

if (int.TryParse(ids[0], out int id1) && int.TryParse(ids[1], out int id2))

{

var parents = db.Student\_Parent\_Relationship.Where(spr => spr.StudentId == id1);

foreach (Student\_Parent\_Relationship parent in parents) db.Student\_Parent\_Relationship.Remove(parent);

db.SaveChanges();

db.Student\_Parent\_Relationship.Add( new Student\_Parent\_Relationship()

{

StudentId = student.StudentId,

ParentId = id1

});

db.Student\_Parent\_Relationship.Add(new Student\_Parent\_Relationship()

{

StudentId = student.StudentId,

ParentId = id2

});

db.SaveChanges();

}

else MessageBox.Show("Ошибка! Введены не числовые значения");

}

else MessageBox.Show("Ошибка! Введите 1 или 2 идентификатора");

UpdateData();

}

}

}

<Window x:Class="WpfApp1.UsersWindow"

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"

ResizeMode="NoResize"

Height="810" Width="1440"

Background="#ECF0F3"

WindowStartupLocation="CenterScreen"

Title="UsersWindow">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Width" Value="290"/>

<Setter Property="Background" Value="#0a1451"/>

<Setter Property="BorderBrush" Value="#0a1451"/>

<Setter Property="FontSize" Value="28"/>

<Setter Property="Foreground" Value="#FFFFFF"/>

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height=".15\*"/>

<RowDefinition Height=".7\*"/>

<RowDefinition Height=".15\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Background="#0A1451" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="КУРАТОРЫ" Click="OpenCurators"/>

<Button Content="СТУДЕНТЫ" Click="OpenStudents"/>

<Button Content="ПРЕПОДАВАТЕЛИ" Click="OpenTeachers"/>

<Button Content="РОДИТЕЛИ" Click="OpenParents"/>

<Button Content="ЗАЧЁТКИ" Click="OpenLedgers"/>

</StackPanel>

<Canvas Grid.Row="1">

<TextBlock Text="СТУДЕНТЫ"

Canvas.Left="602"

Canvas.Top="27"

FontSize="48"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Foreground="#0A1451"/>

<ScrollViewer Width="863" Height="436" Canvas.Left="322" Canvas.Top="110" VerticalScrollBarVisibility="Visible" HorizontalAlignment="Left" VerticalAlignment="Center">

<StackPanel Name="CardsList">

<WrapPanel Grid.Column="0" Width="800" Name="UserCard0">

<Canvas Background="#FFFFFF" Width="750" Height="115">

<Image Source="Source/avatar guest.png" Height="87" Width="79" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBlock Text="Владислав Бусов" FontSize="16" Foreground="#0A1451" FontWeight="Bold" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="8"/>

<TextBlock Text="4338" FontSize="16" FontWeight="Bold" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="703" VerticalAlignment="Center" Canvas.Top="8"/>

<TextBlock Text="15.07.2006" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="29"/>

<TextBlock Text="+79196395387" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="47"/>

<TextBlock Text="Куратор: Гимадеев И.Ф." FontSize="16" FontWeight="Bold" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="10" VerticalAlignment="Top" Canvas.Top="84"/>

<Image Source="Source/edit.png" Width="20" Height="20" Canvas.Left="695" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Image Source="Source/trash.png" Width="20" Height="20" Canvas.Left="720" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Canvas.Effect>

<DropShadowEffect Opacity=".5" RenderingBias="Performance" Direction="225" ShadowDepth="6" BlurRadius="10"/>

</Canvas.Effect>

</Canvas>

</WrapPanel>

</StackPanel>

</ScrollViewer>

<Button Click="CreateUser" Height="50" Width="50" Canvas.Left="1339" Canvas.Top="454" Foreground="#FFFFFF" FontSize="40">

<Button.Style>

<Style TargetType="Button">

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="true">

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

</Trigger>

</Style.Triggers>

</Style>

</Button.Style>

<Button.Template>

<ControlTemplate>

<Border CornerRadius="25" Background="#0A1451">

<ContentPresenter Content="+" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0,-5,0,0"/>

</Border>

</ControlTemplate>

</Button.Template>

</Button>

</Canvas>

<Rectangle Grid.Row="2">

<Rectangle.Fill>

#0A1451

</Rectangle.Fill>

</Rectangle>

</Grid>

</Window>

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.Markup;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для UsersWindow.xaml

/// </summary>

public partial class UsersWindow : Window

{

public UsersWindow()

{

InitializeComponent();

ShowData();

}

public void ShowData()

{

var db = CuratorsEntities1.GetContext();

List<Student> students = db.Student.ToList();

if (students.Count == 0) this.CardsList.Children.Clear();

else if (students.Count > 1)

{

for (int i = 1; i < students.Count; i++)

{

WrapPanel el = XamlReader.Parse(

XamlWriter.Save(

this.UserCard0

)

) as WrapPanel;

//el.SetValue(Grid.ColumnProperty, i % 3);

//el.SetValue(Grid.RowProperty, i / 3);

this.RegisterName($"UserCard{i}", el);

this.CardsList.Children.Add(el);

}

}

for (int i = 0; i < students.Count; i++)

{

var student = students[i];

int TBCounter = 0;

int ICounter = 0;

Canvas container = (this.FindName($"UserCard{i}") as WrapPanel).Children[0] as Canvas;

foreach (UIElement el in container.Children)

{

string val = "";

if (el.GetType().ToString() == "System.Windows.Controls.TextBlock")

{

if (TBCounter == 0) val = $"{student.FirstName} {student.LastName}";

else if (TBCounter == 1) val = student.GroupNumber;

else if (TBCounter == 2) val = student.BDate;

else if (TBCounter == 3) val = student.CellPhone;

else if (TBCounter == 4)

{

Curator curator = db.Curator.FirstOrDefault(c => c.CuratorID == student.CuratorId);

val = $"Куратор: {curator.FirstName} {curator.LastName}";

}

el.SetValue(TextBlock.TextProperty, val);

TBCounter++;

}

if (el.GetType().ToString() == "System.Windows.Controls.Image")

{

if (ICounter == 1)

{

el.MouseUp += (o, e) =>

{

var win = new UserDetail(student);

win.Show();

this.Close();

};

}

else if (ICounter == 2)

{

el.MouseUp += (o, e) =>

{

db.Student.Remove(student);

db.SaveChanges();

var win = new UsersWindow();

win.Show();

this.Close();

//ShowData();

};

}

ICounter++;

}

}

}

}

private void OpenCurators(object sender, RoutedEventArgs e)

{

(new CuratorsWindow()).Show();

this.Close();

}

private void OpenStudents(object sender, RoutedEventArgs e)

{

(new UsersWindow()).Show();

this.Close();

}

private void OpenParents(object sender, RoutedEventArgs e)

{

(new ParentsWindow()).Show();

this.Close();

}

private void OpenTeachers(object sender, RoutedEventArgs e)

{

(new TeachersWindow()).Show();

this.Close();

}

private void OpenLedgers(object sender, RoutedEventArgs e)

{

(new LedgersWindows()).Show();

this.Close();

}

private void CreateUser(object sender, RoutedEventArgs e)

{

var db = CuratorsEntities1.GetContext();

Student student = new Student()

{

FirstName = "\_\_\_\_\_\_\_\_\_\_",

LastName = "\_\_\_\_\_\_\_\_\_\_",

CellPhone = "+78917264893",

BDate = "\_\_.\_\_.\_\_\_\_",

GroupNumber = "\_\_\_\_",

CuratorId = db.Curator.First().CuratorID

};

db.Student.Add(student);

db.SaveChanges();

var win = new UserDetail(student);

win.Show();

this.Close();

}

}

}

<Window x:Class="WpfApp1.Window1"

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="Window1" Height="450" Width="1700">

<ListView >

<StackPanel Orientation="Horizontal">

<WrapPanel Width="400">

<Canvas Background="#FFFFFF" Width="350" Height="115">

<Image Source="Source/avatar guest.png" Height="87" Width="79" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBlock Text="Владислав Бусов" FontSize="16" Foreground="#0A1451" FontWeight="Bold" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="8"/>

<TextBlock Text="4338" FontSize="16" FontWeight="Bold" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="303" VerticalAlignment="Center" Canvas.Top="8"/>

<TextBlock Text="15.07.2006" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="29"/>

<TextBlock Text="+79196395387" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="47"/>

<TextBlock Text="Куратор: Гимадеев И.Ф." FontSize="16" FontWeight="Bold" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="10" VerticalAlignment="Top" Canvas.Top="84"/>

<Image Source="Source/edit.png" Width="20" Height="20" Canvas.Left="295" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Image Source="Source/trash.png" Width="20" Height="20" Canvas.Left="320" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Canvas.Effect>

<DropShadowEffect Opacity=".5" RenderingBias="Performance" Direction="225" ShadowDepth="6" BlurRadius="10"/>

</Canvas.Effect>

</Canvas>

</WrapPanel>

<WrapPanel Grid.Column="0" Width="400">

<Canvas Background="#FFFFFF" Width="350" Height="115">

<Image Source="Source/avatar guest.png" Height="87" Width="79" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBlock Text="Владислав Бусов" FontSize="16" Foreground="#0A1451" FontWeight="Bold" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="8"/>

<TextBlock Text="4338" FontSize="16" FontWeight="Bold" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="303" VerticalAlignment="Center" Canvas.Top="8"/>

<TextBlock Text="15.07.2006" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="29"/>

<TextBlock Text="+79196395387" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="47"/>

<TextBlock Text="Куратор: Гимадеев И.Ф." FontSize="16" FontWeight="Bold" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="10" VerticalAlignment="Top" Canvas.Top="84"/>

<Image Source="Source/edit.png" Width="20" Height="20" Canvas.Left="295" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Image Source="Source/trash.png" Width="20" Height="20" Canvas.Left="320" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Canvas.Effect>

<DropShadowEffect Opacity=".5" RenderingBias="Performance" Direction="225" ShadowDepth="6" BlurRadius="10"/>

</Canvas.Effect>

</Canvas>

</WrapPanel>

<WrapPanel Grid.Column="0" Width="400">

<Canvas Background="#FFFFFF" Width="350" Height="115">

<Image Source="Source/avatar guest.png" Height="87" Width="79" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBlock Text="Владислав Бусов" FontSize="16" Foreground="#0A1451" FontWeight="Bold" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="8"/>

<TextBlock Text="4338" FontSize="16" FontWeight="Bold" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="303" VerticalAlignment="Center" Canvas.Top="8"/>

<TextBlock Text="15.07.2006" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="29"/>

<TextBlock Text="+79196395387" FontSize="16" FontWeight="Light" Foreground="#0A1451" HorizontalAlignment="Center" Canvas.Left="84" VerticalAlignment="Top" Canvas.Top="47"/>

<TextBlock Text="Куратор: Гимадеев И.Ф." FontSize="16" FontWeight="Bold" Foreground="#0A1451" HorizontalAlignment="Left" Canvas.Left="10" VerticalAlignment="Top" Canvas.Top="84"/>

<Image Source="Source/edit.png" Width="20" Height="20" Canvas.Left="295" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Image Source="Source/trash.png" Width="20" Height="20" Canvas.Left="320" Canvas.Top="85" HorizontalAlignment="Left" VerticalAlignment="Center" Cursor="Hand"/>

<Canvas.Effect>

<DropShadowEffect Opacity=".5" RenderingBias="Performance" Direction="225" ShadowDepth="6" BlurRadius="10"/>

</Canvas.Effect>

</Canvas>

</WrapPanel>

</StackPanel>

</ListView>

</Window>

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 WpfApp1

{

/// <summary>

/// Логика взаимодействия для Window1.xaml

/// </summary>

public partial class Window1 : Window

{

public Window1()

{

InitializeComponent();

}

}

}