ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО

ОБЕСПЕЧНИЯ

КП.181.

Министерство цифрового развития, связи и массовых коммуникаций РФ

Уральский технический институт связи и информатики (филиал)

ФГБОУ ВО «Сибирский государственный университет

телекоммуникаций и информатики»

в г. Екатеринбурге

(УрТИСИ СибГУТИ)

КУРСОВОЙ ПРОЕКТ

по междисциплинарному курсу: «Технология разработки программного обеспечения»

на тему: «Разработка электронного учебного пособия по теме «Построение минимального остовного графа» для АИС кафедры ИСТ УрТИСИ СибГУТИ»

Выполнил:

студент группы 181

Писниченко А. А.

Руководитель:

преподаватель

Ермоленко О. М.

г. Екатеринбург, 2023

Министерство цифрового развития, связи и массовых коммуникаций РФ

Уральский технический институт связи и информатики (филиал)

ФГБОУ ВО «Сибирский государственный университет

телекоммуникаций и информатики»

в г. Екатеринбурге

(УрТИСИ СибГУТИ)

Цикловая комиссия ИТ и АСУ кафедры ИСТ

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на курсовой проект

по дисциплине: «Технология разработки программного обеспечения»

на тему: «Разработка электронного учебного пособия для АИС кафедры ИСТ УрТИСИ СибГУТИ»

студента(ки) группы\_\_181 \_\_\_ Ф.И.О. Писниченко Александр Андреевич

ИСХОДНЫЕ ДАННЫЕ

ВАРИАНТ №13

В работе должно быть представлено приложение, предназначенное для предоставления информации и тестирования студентов на тему «Построение минимального остовного дерева графа, которое включает в себя – дружественный интерфейс программы для удобного использования приложением, авторизация.

При выполнении курсовой работы (проекта) на указанную тему должны быть представлены:

1. Пояснительная записка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Введение; Постановка задачи; Анализ предметной области; Аналогичные программные решения; Требования к программному продукту; Обоснование выбора инструментальных средств разработки; Разработка приложения: Структура программы; Реализация программы; Проектирование макетов интерфейса пользователя; Инструкция по работе с программой; Заключение; Библиография. 00000000

1. Графическая часть000000000000000000000000000000000000000000000

Дата выдачи \_\_\_\_\_\_\_\_ Дата окончания \_\_\_\_\_\_\_\_\_

Председатель ЦК ИТиАСУ кафедры ИСТ Ермоленко О.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель – руководитель преподаватель кафедры ИСТ Ермоленко О. М.

Студент Писниченко Александр Андреевич \_\_\_\_\_\_\_\_\_\_\_\_\_

Отзыв руководителя

Оглавление

[Введение 6](#_Toc152106813)

[1 Обоснование выбора темы курсового проекта 7](#_Toc152106814)

[1.1 Анализ предметной области 7](#_Toc152106815)

[1.2 Аналогичные программные решения 7](#_Toc152106816)

[1.3 Требования к программному продукту 8](#_Toc152106821)

[1.4 Обоснование выбора инструментальных средств разработки 10](#_Toc152106822)

[2 Разработка приложения 12](#_Toc152106823)

[2.1 Структура программы 12](#_Toc152106824)

[2.2 Реализация программы 13](#_Toc152106825)

2.3 Проектирование макетов интерфейса пользователя…………………………..16

[2.3 Инструкция по работе с программой 18](#_Toc152106826)

[Заключение 23](#_Toc152106827)

[Библиография 24](#_Toc152106828)

[Приложение А 25](#_Toc152106829)

# Введение

Успешное функционирование различных фирм, организаций и предприятий невозможно без развитой информационной системы, которая позволяет автоматизировать сбор и обработку данных. Помимо того, подобные информационные системы также могут быть использованы в целях упрощения и автоматизации обучения персонала, либо студентов.

В рамках данного курсового проекта рассматривается процесс обучения на кафедре ИСТ. Он предусматривает анализ предметной области и разработку электронного учебного пособия, с помощью которого будет упрощен процесс обучения и тестирования студентов.

Целью курсового проекта является применение технологии создания баз данных (БД) и приложений средствами SQL Server.

Техническое задание по созданию web-приложения для автоматизации процесса обучения и тестирования студентов предусматривает ряд задач, которые необходимо выполнить в процессе выполнения работы:

1. сформировать информационную и логическую структуру приложения на основе анализа предметной области и требуемых функций, реализующих хранение, поиск и обновление информации;

2. подключить базу данных к приложению;

3. реализовать авторизацию студента и преподавателя;

4. разработать функционал и интерфейс приложения.

# 1 Обоснование выбора темы курсового проекта

## Анализ предметной области

## Задание на курсовой проект было выдано АИС кафедрой ИСТ УрТИСИ СибГУТИ. На кафедре проводится обучение специальностям в сфере информационных технологий.

## Выбор темы проекта основан на нескольких важных факторах.

Во-первых, в данный момент на кафедре ИСТ имеются аналогичные программы, но они не соответствуют требованиям, и не способны предоставить полный объем информации на тему «Построение минимального остовного графа».

Во-вторых, на кафедре требуется такое программное средство, использование которого было бы бесплатным, ведь на данный момент аналогичные использующиеся преподавателями программы требуют платной подписки.

## Таким образом, выбор темы «Разработка электронного учебного пособия на тему «Построение минимального остовного графа» обоснован актуальностью и пользой такого проекта для обучения студентов и улучшения процесса образования на кафедре ИСТ.

## Аналогичные программные решения

Функционал учебного пособия, который будет использоваться в курсовом проекте, включен во многих других решениях.

## Для обучения персонала или студентов можно использовать специализированные программные приложения для учета успеваемости и изучения материала.

## Такие приложения позволяют удобно получать доступ к образовательным материалам, открывают возможности обучаться в любом месте и в любое время, а также интерактивно обучаться с использованием мультимедийных элементов.

## Приложения «Coassemble» и «Raptivity» предназначено для использования в различных компаниях или учебных заведениях.

## Training Manual Creator Software - Coassemble

Рисунок 1.1 – Приложение «Coassemble»



Рисунок 1.2 – Приложение «Raptivity»

## Эти приложения облегчают работу по обучению и тестированию, помогают избежать больших затрат времени, что важно для обеспечения эффективности образовательного процесса. Главным минусом этих приложений является высокая стоимость использования, из-за чего их использование становится нерациональным.

## 1.3 Требования к программному продукту

Обычно требования к программному продукту разрабатываются в начале проекта и обсуждаются с заказчиком или пользователем для уточнения и согласования. Весь процесс разработки и внесение изменений в требования требует тесного взаимодействия между заказчиками, пользователями и разработчиками.

Функциональные требования описывают, как должна работать система или программное обеспечение, какие функции она должна выполнять и какие задачи должна решать. Они формулируются в виде конкретных и мероприятий, которые должны быть выполнены. Функциональные требования помогают определить основные возможности и функциональность системы.

Нефункциональные требования, с другой стороны, определяют качественные атрибуты, ожидания и ограничения системы, которые не связаны с ее функционированием. Это может включать производительность, надежность, безопасность, доступность, интерфейс пользователя, масштабируемость, совместимость и т. д. Нефункциональные требования помогают определить ожидания пользователей в отношении системы или программного обеспечения и обеспечить ее успешную работу в соответствии с этими ожиданиями.

Оба типа требований необходимы для успешного проектирования, разработки и реализации системы или программного обеспечения. Функциональные требования определяют, что должно быть сделано, а нефункциональные требования - как оно должно быть сделано и какие требования должны быть учтены для обеспечения качественной работы системы или программного обеспечения.

Функциональные требования:

1. аутентификация пользователей – система должна позволять преподавателю создавать учетные записи, входить в систему и менять пароли;

2. предоставление студентам теоретической информации по теме;

3. предоставление студентам возможности пройти тестирование;

4. возможность преподавателю искать информацию о студентах.

Нефункциональные требования:

1. безопасность – система должна быть защищена от несанкционированного доступа, утечки данных;

2. производительность – система должна быть способна стабильно работать, без сбоев;

3. надежность – система должна быть надежной и соответствовать требованиям пользователя;

Минимальные технические требования к устройству для работы с приложением.

Минимальные требования:

- Операционная система: Windows 7 или более поздняя версия

- Процессор: 1 ГГц или более

- Оперативная память: 500 МБ для 32-разрядной версии или 2 ГБ для 64-разрядной версии

- Место на жестком диске: 200 МБ свободного места

- Дисплей: разрешение 1280 x 800 пикселей

## 

## 1.4 Обоснование выбора инструментальных средств разработки

Для создания макетов интерфейса пользователя и макета структуры программы будет использоваться Microsoft Visual Studio.

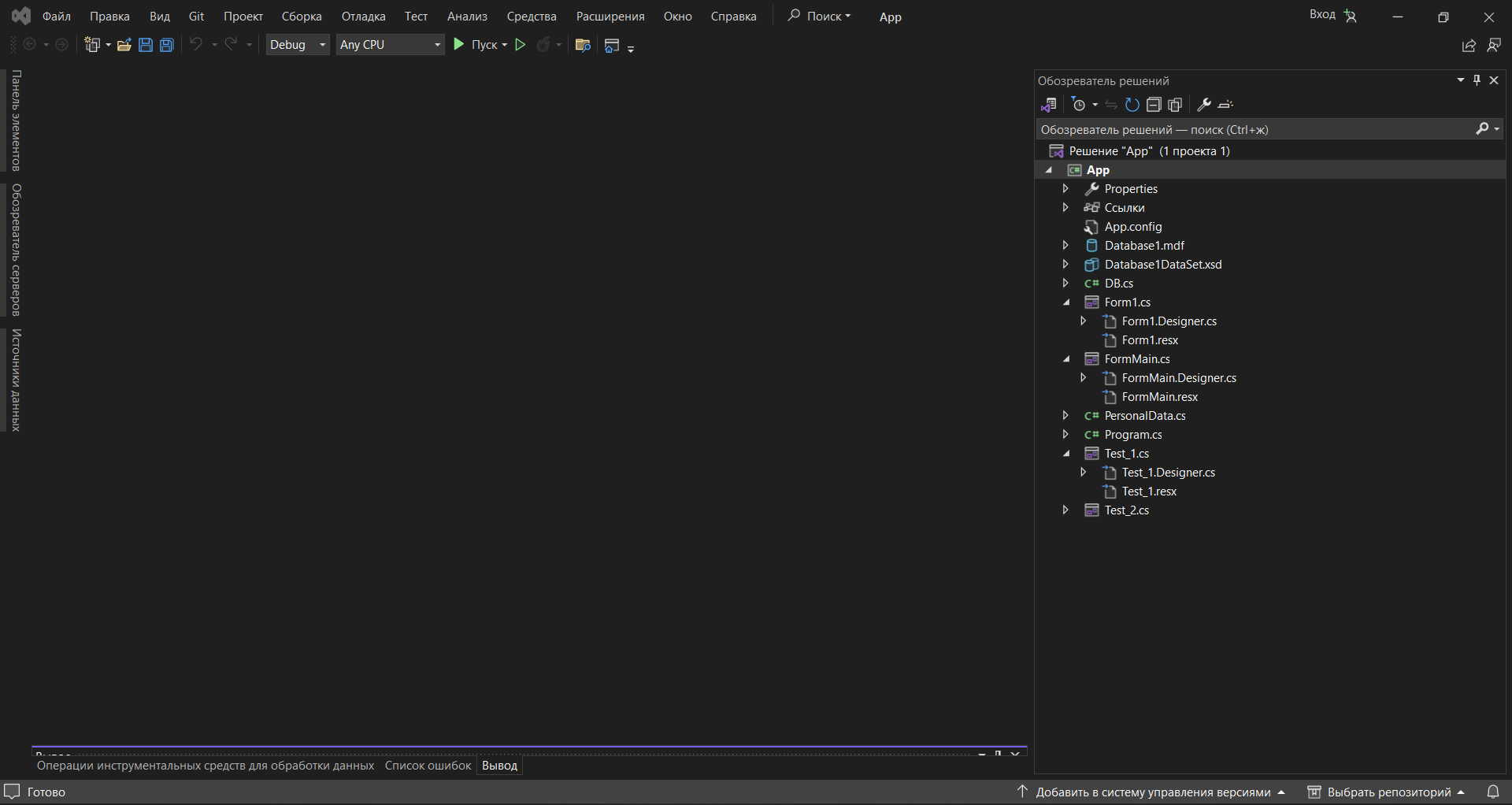


Рисунок 1.3 – Microsoft Visual Studio

Выбор инструментальных средств разработки «Microsoft Visual Studio» обоснован следующими причинами:

1. Удобство использования - Microsoft Visual Studio предоставляет удобную среду разработки с множеством инструментов и функций, таких как отладка, автодополнение кода, управление версиями и многие другие, что упрощает и ускоряет процесс разработки.

2. Поддержка различных платформ - Microsoft Visual Studio поддерживает разработку приложений для различных платформ, включая Windows, Android, iOS, веб-приложения и многие другие, что делает его универсальным инструментом для разработчиков.

3. Богатый функционал - Microsoft Visual Studio предоставляет широкий набор инструментов для разработки, тестирования и оптимизации приложений, что позволяет создавать высококачественные программные продукты.

4. Поддержка сообщества - Microsoft Visual Studio имеет активное сообщество разработчиков, где можно найти множество полезной информации, обучающих материалов и поддержку со стороны опытных специалистов.

Некоторые аналоги Microsoft Visual Studio включают в себя следующие инструменты разработки:

1. IntelliJ IDEA - это интегрированная среда разработки, которая предоставляет широкий набор инструментов для разработки приложений на Java, Kotlin, Groovy и других языках программирования.

2. Eclipse - это популярная интегрированная среда разработки, которая поддерживает различные языки программирования и технологии, такие как Java, C/C++, PHP, и другие.

3. Visual Studio Code - это легковесная и мощная среда разработки от Microsoft, которая поддерживает различные языки программирования и платформы.

4. Xcode - это интегрированная среда разработки от Apple, которая предназначена для создания приложений для iOS, macOS, watchOS и tvOS.

5. Android Studio - это официальная интегрированная среда разработки для создания приложений для платформы Android. Каждый из этих инструментов имеет свои особенности и преимущества, и выбор зависит от конкретных потребностей и предпочтений разработчика.

Исходя из вышеперечисленных преимуществ, выбор инструментальных средств разработки «Microsoft Visual Studio» является обоснованным для создания качественных и многофункциональных программных продуктов.

# 

# 2 Разработка приложения

## 2.1 Структура программы

Структура web-приложения необходима для организации и упорядочения компонентов приложения, чтобы оно было более поддерживаемым, масштабируемым и удобным в использовании. Она позволяет разделить различные функциональности приложения на модули, компоненты и слои, облегчая разработку, тестирование и сопровождение приложения. Выделяют ряд преимуществ:

1. Организация кода. Структура позволяет легко организовать и классифицировать код внутри приложения. Она может включать разделение кода на модули или компоненты, группировку функциональности по слоям (например, представление, логика приложения, доступ к данным) и использование стандартных паттернов разработки.

2. Разделение ответственности. Проектирование приложения с использованием структуры помогает разделить ответственность между различными частями кода. Например, модули представления могут отвечать за обработку пользовательского интерфейса, модули логики приложения могут выполнять бизнес-логику, а модули доступа к данным могут обрабатывать запросы к базе данных или другим источникам данных.

3. Легкость тестирования. Структура позволяет разделить приложение на отдельные компоненты, которые можно тестирует независимо друг от друга. Это упрощает написание и выполнение модульных тестов, функциональных тестов и интеграционных тестов.

4. Масштабируемость. Хорошо спроектированная структура позволяет легко масштабировать приложение по мере его роста. Модульная архитектура облегчает добавление новых функций или компонентов, а разделение кода на слои позволяет горизонтально и вертикально масштабировать приложение.

5. Понятность и удобство использования. Хорошо организованная структура делает код более понятным и удобным для использования. Разработчики могут быстро найти нужный код, понять его функциональность и легко работать с ним.

В целом, структура web-приложения является фундаментом для эффективной разработки и сопровождения приложения. Она помогает создать легко поддерживаемое, масштабируемое и удобное в использовании приложение.

Для создания структуры ПО будет использована интегрированная среда разработки Microsoft Visual Studio.

Опишем, какие страницы будут в приложении для удобства.

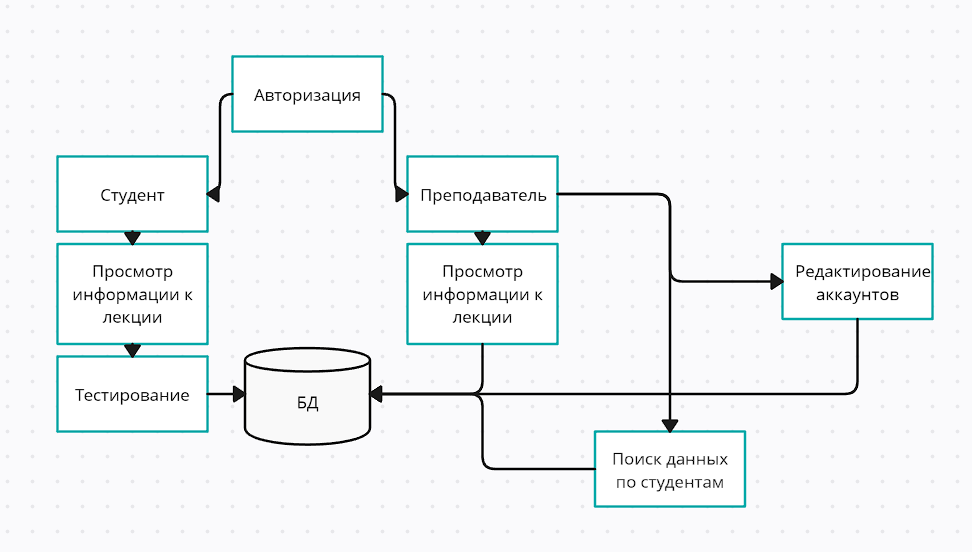
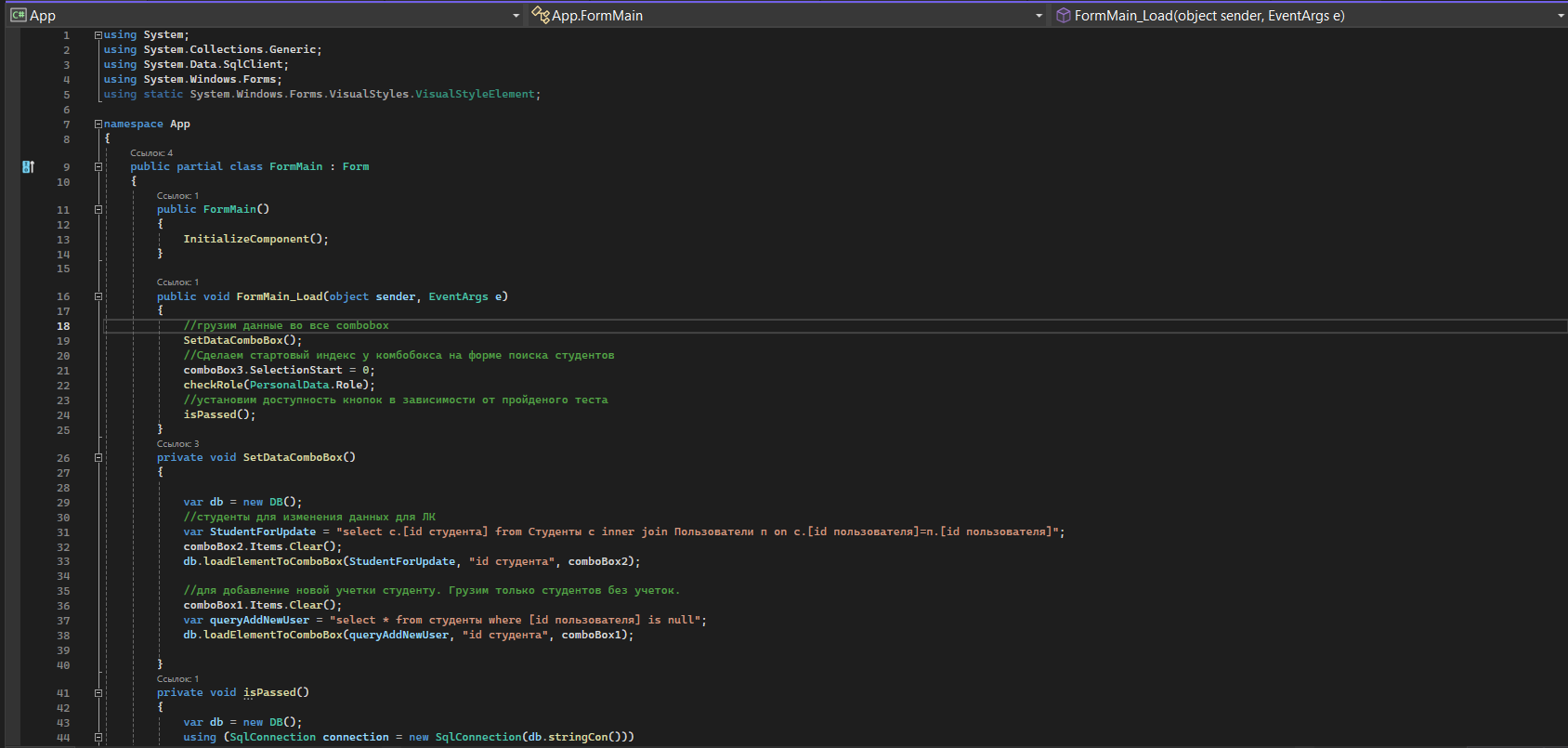
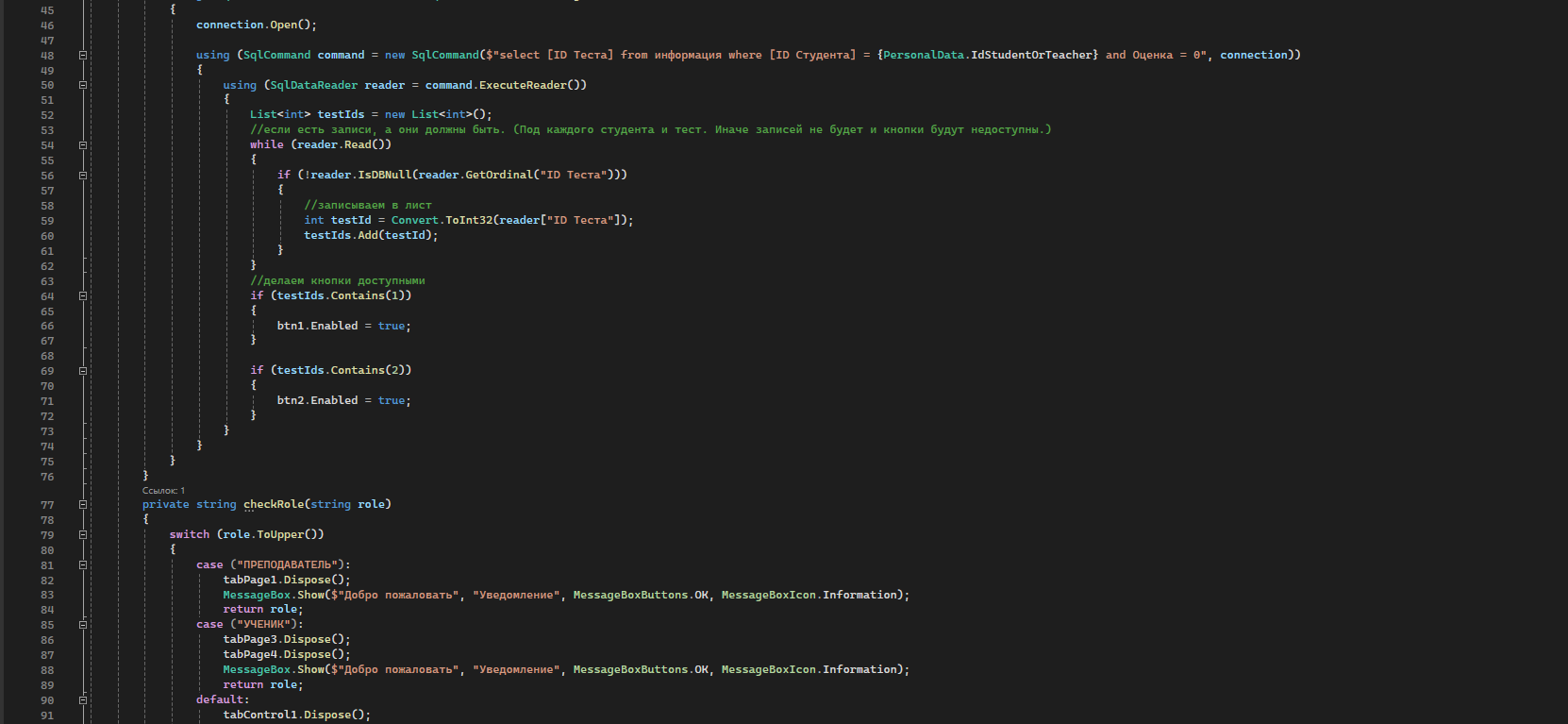


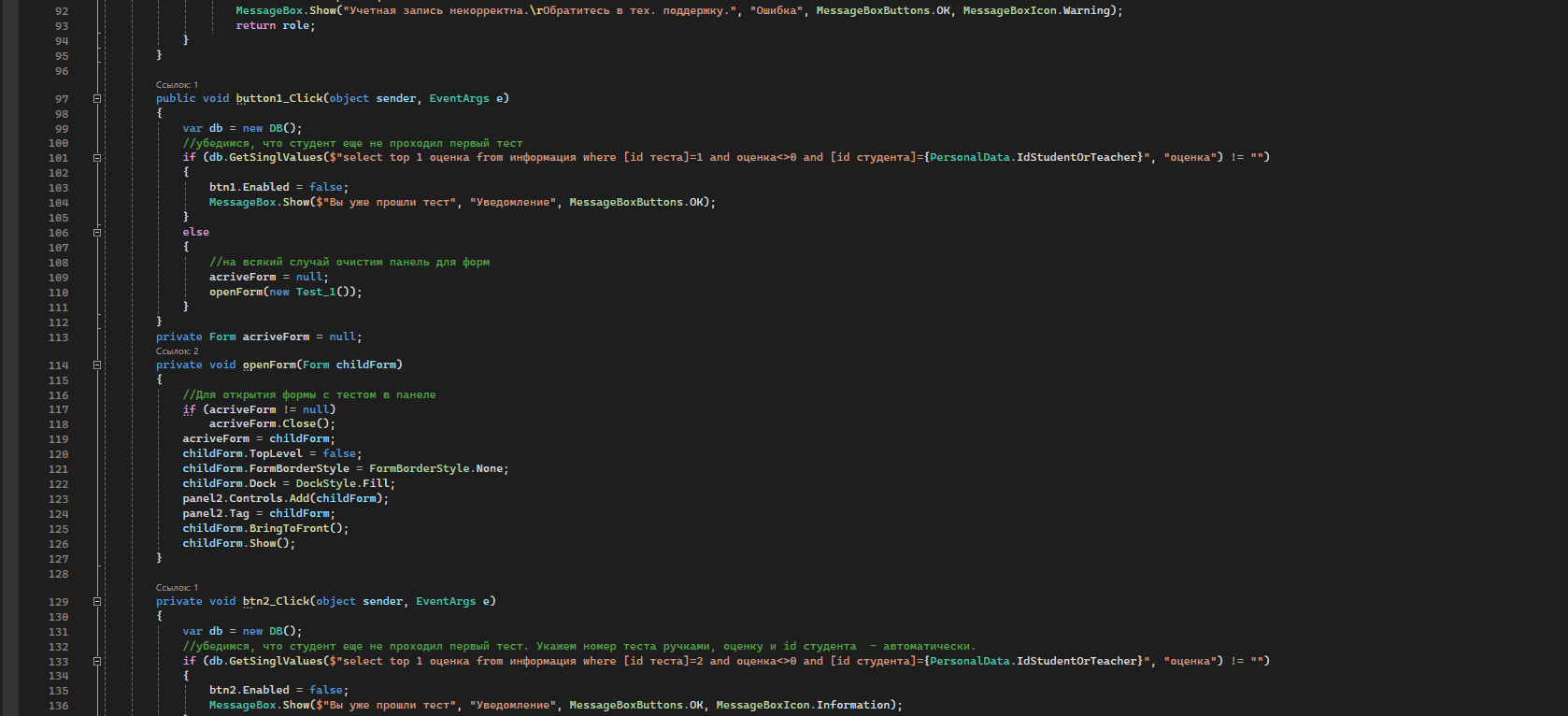
Рисунок 2.1 – Структура программы

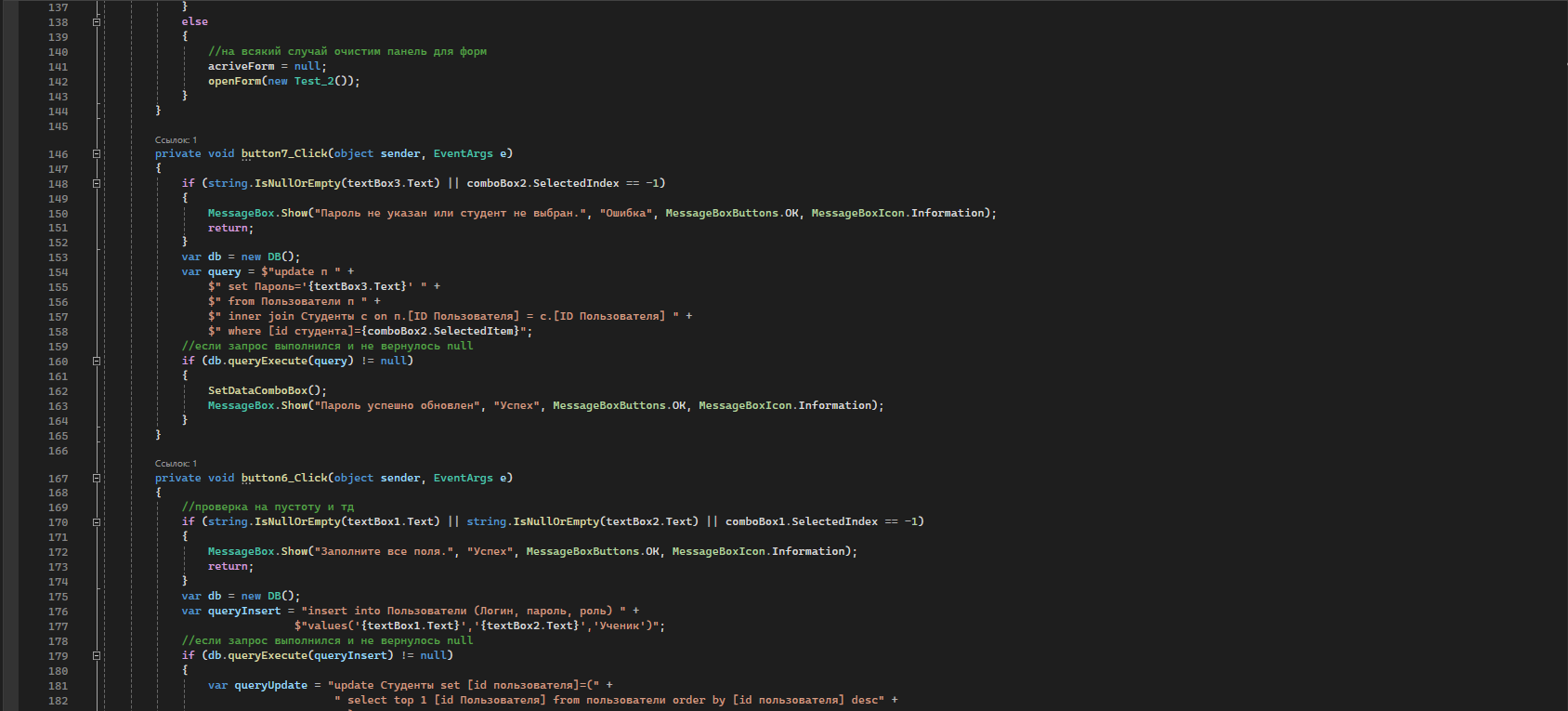
## Реализация программы

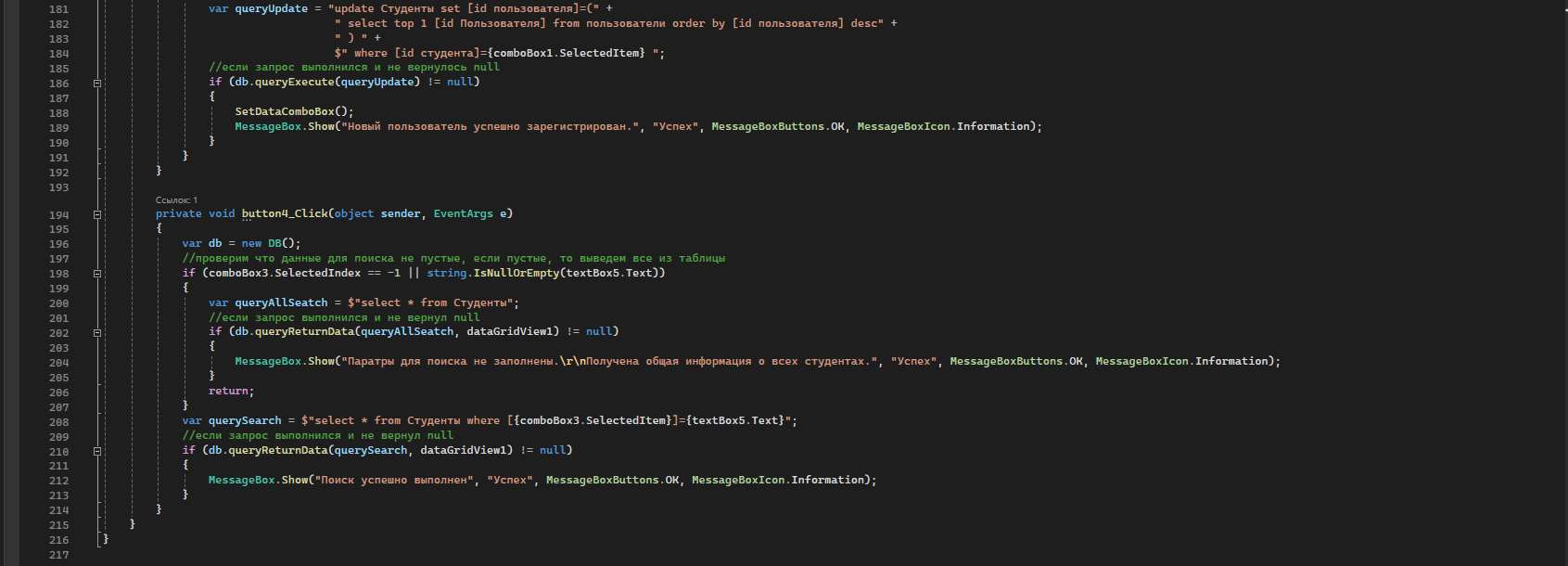
## Главная форма (FormMain) является ядром программы, включающим в себя всю основную логику базы данных. Ее код представлен на рисунках 2.2 – 2.6

Рисунок 2.2 – Структура главной формы, часть 1

Рисунок 2.3 – Структура главной формы, часть 2

Рисунок 2.4 – Структура главной формы, часть 3

Рисунок 2.5 – Структура главной формы, часть 4

 Рисунок 2.6 – Структура главной формы, часть 5

* 1. Проектирование макетов интерфейса пользователя

Проектирование макетов интерфейса пользователя является важной составляющей процесса разработки программного продукта или веб-сайта. Оно помогает разработчикам и дизайнерам создать наглядное представление о том, как будет выглядеть, и функционировать окончательный продукт.

Основные причины, по которым проектирование макетов интерфейса пользователя является необходимым.

Визуализация идеи. Проектирование макетов позволяет визуализировать идею и концепцию продукта. Оно помогает команде проекта представить, как будет выглядеть и работать окончательный продукт, и убедиться, что все заинтересованные стороны имеют единое представление о конечной цели.

Предотвращение ошибок. Макеты интерфейса пользователя позволяют идентифицировать потенциальные ошибки и проблемы до начала фактической разработки. Они позволяют проанализировать и оценить эффективность, удобство и понятность интерфейса пользователя и внести необходимые изменения, если это необходимо.

Определение пользовательского опыта. Макеты интерфейса пользователя помогают определить пользовательский опыт и удовлетворение. Они позволяют тестировать разные варианты пользовательского интерфейса для того, чтобы найти наиболее оптимальный и эффективный.

Экономия времени и ресурсов. Проектирование макетов интерфейса пользователя позволяет сэкономить время и ресурсы, поскольку позволяет идентифицировать и исправить проблемы на более ранних стадиях разработки, когда их исправление гораздо более дешево и проще.

Улучшенное сотрудничество. Макеты интерфейса пользователя являются эффективным средством коммуникации между участниками команды проекта. Они позволяют всем заинтересованным сторонам иметь ясное представление о конечном продукте и участвовать в его разработке и тестировании.

В целом, проектирование макетов интерфейса пользователя помогает обеспечить создание эффективного, удобного и привлекательного пользовательского опыта, что в конечном итоге может повысить удовлетворенность пользователей и успех продукта.

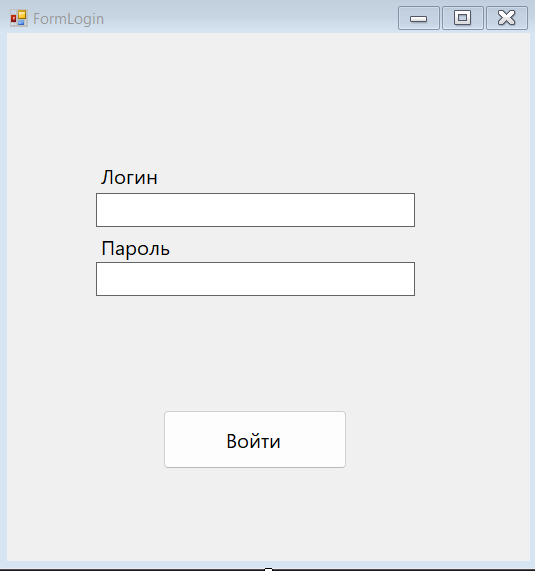


Рисунок 2.7 – Макет авторизации

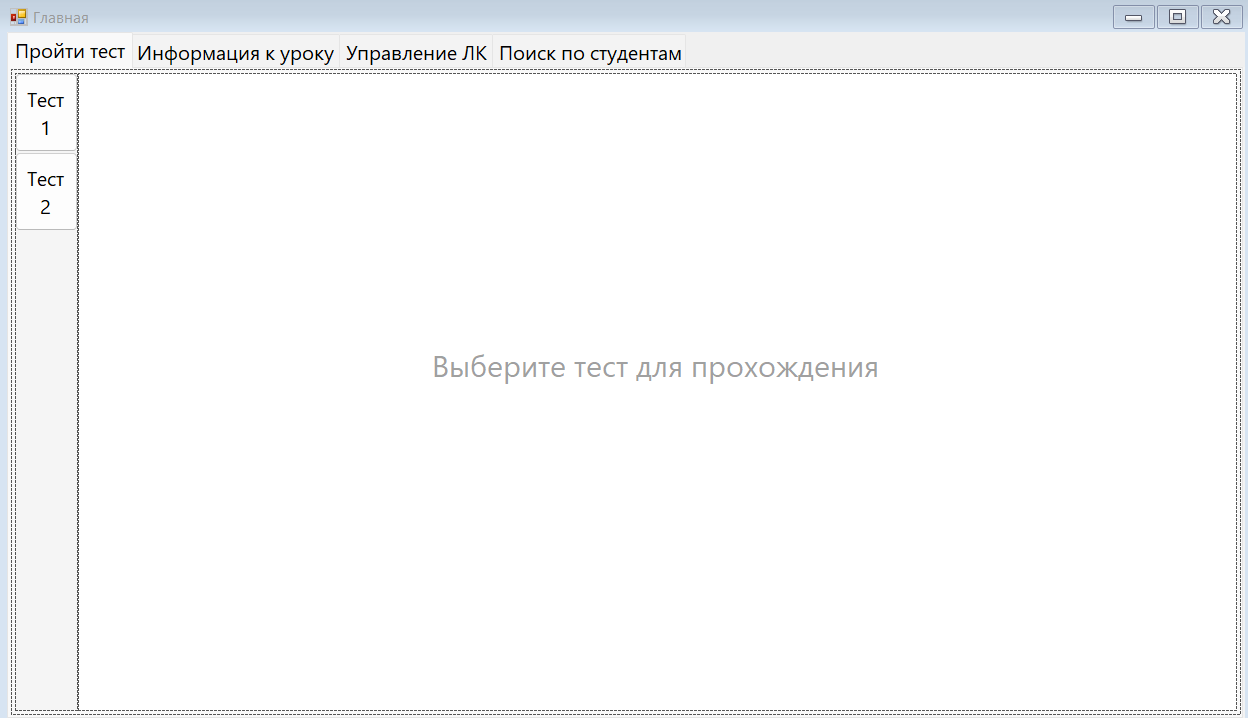


Рисунок 2.8 – Макет главной формы

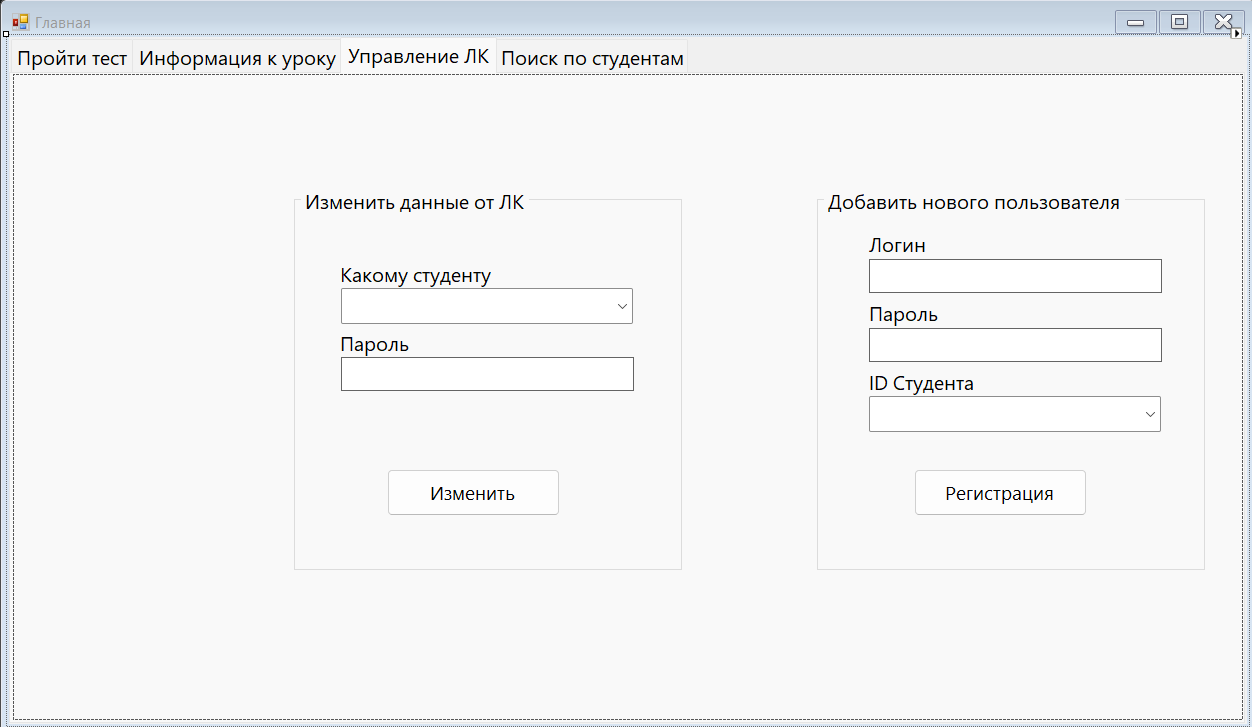


Рисунок 2.9 – Макет изменения данных

## Инструкция по работе с программой

При запуске программы, перед пользователем появляется окно авторизации. Предполагается, что данные для нее он уже имеет.

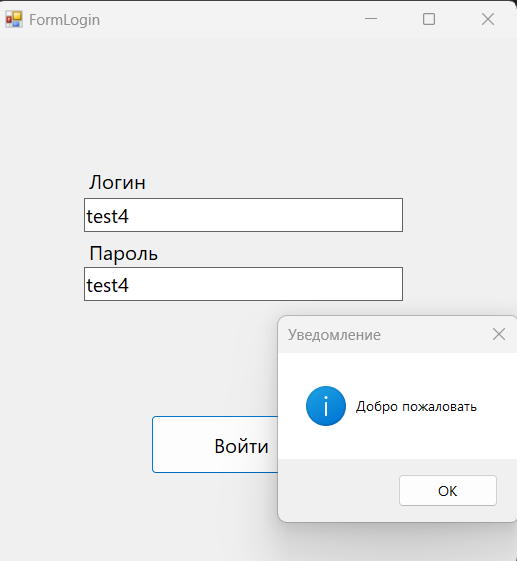


Рисунок 2.10 – Авторизация

В данном случае авторизовался студент. Перед ним появляется информация по теме, с которой необходимо ознакомиться, пролистывая с помощью ползунка справа.

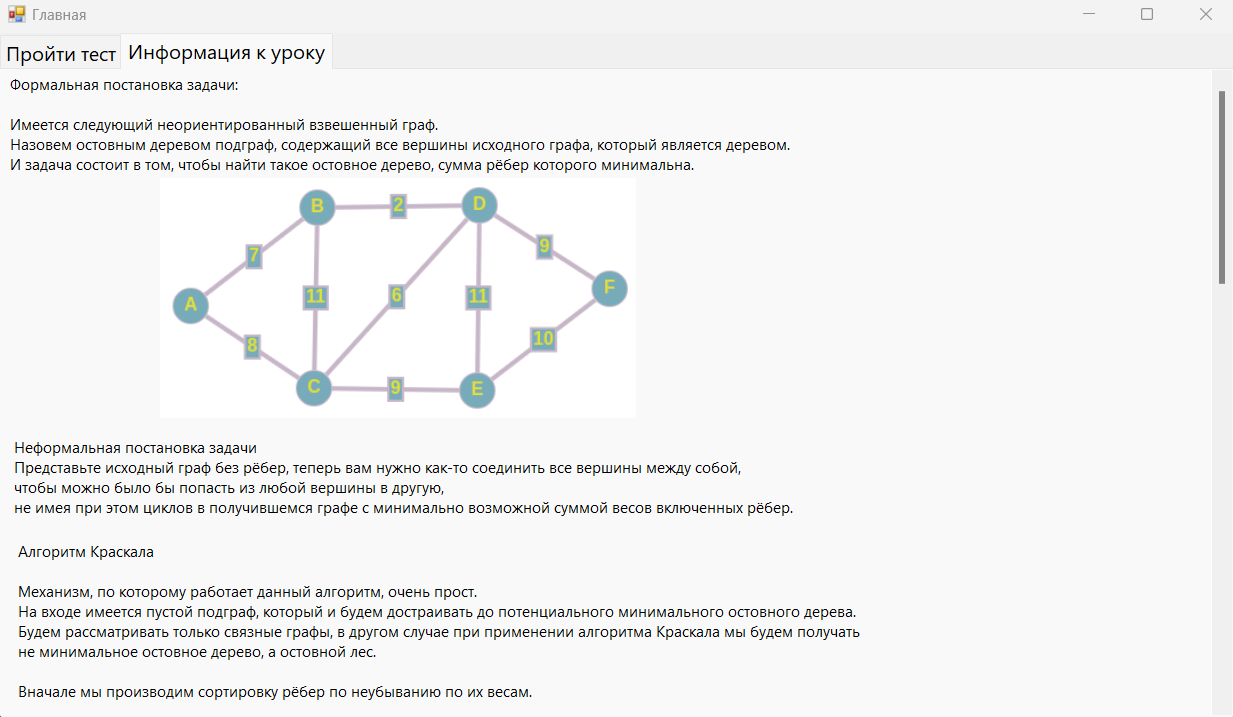


Рисунок 2.11 – Информация по теме, часть 1

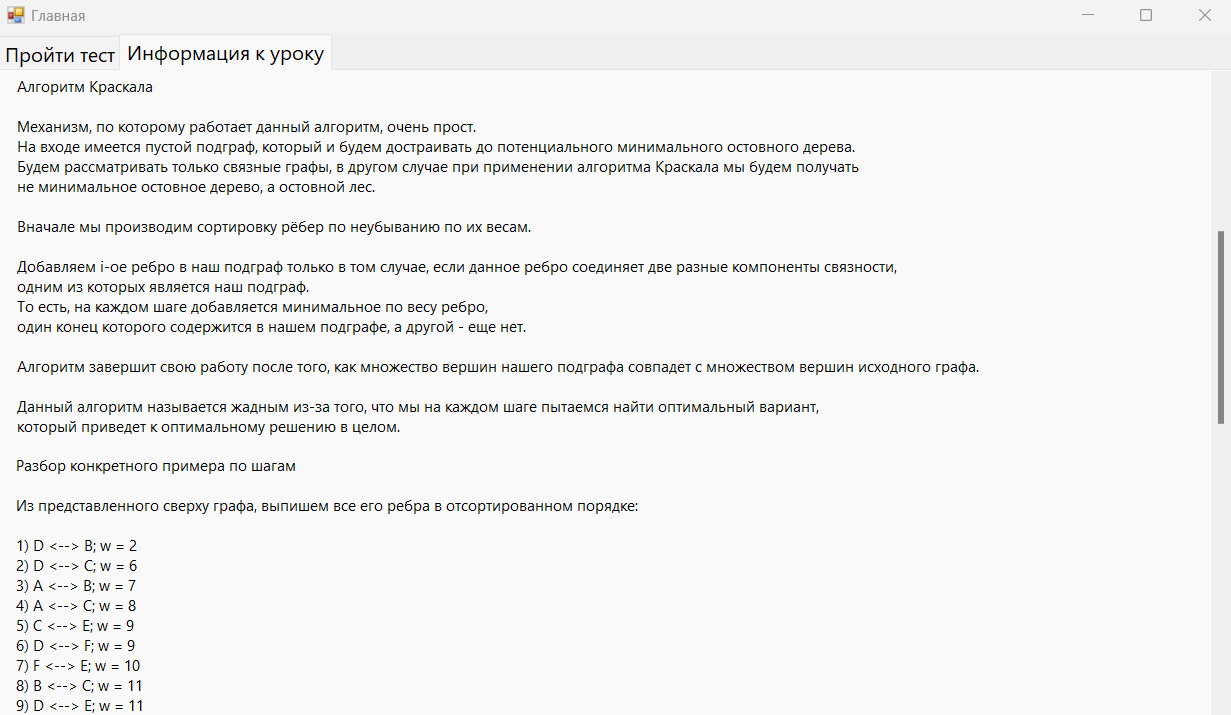


Рисунок 2.12 – Информация по теме, часть 2

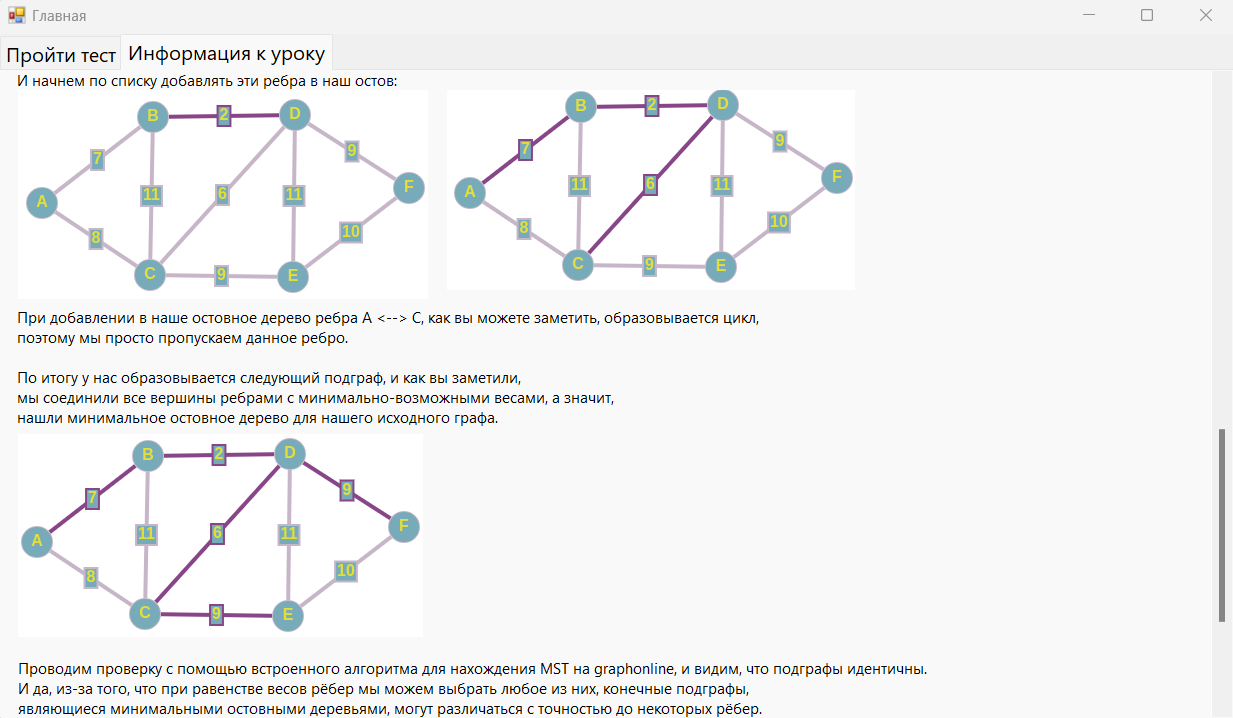


Рисунок 2.13 – Информация по теме, часть 3

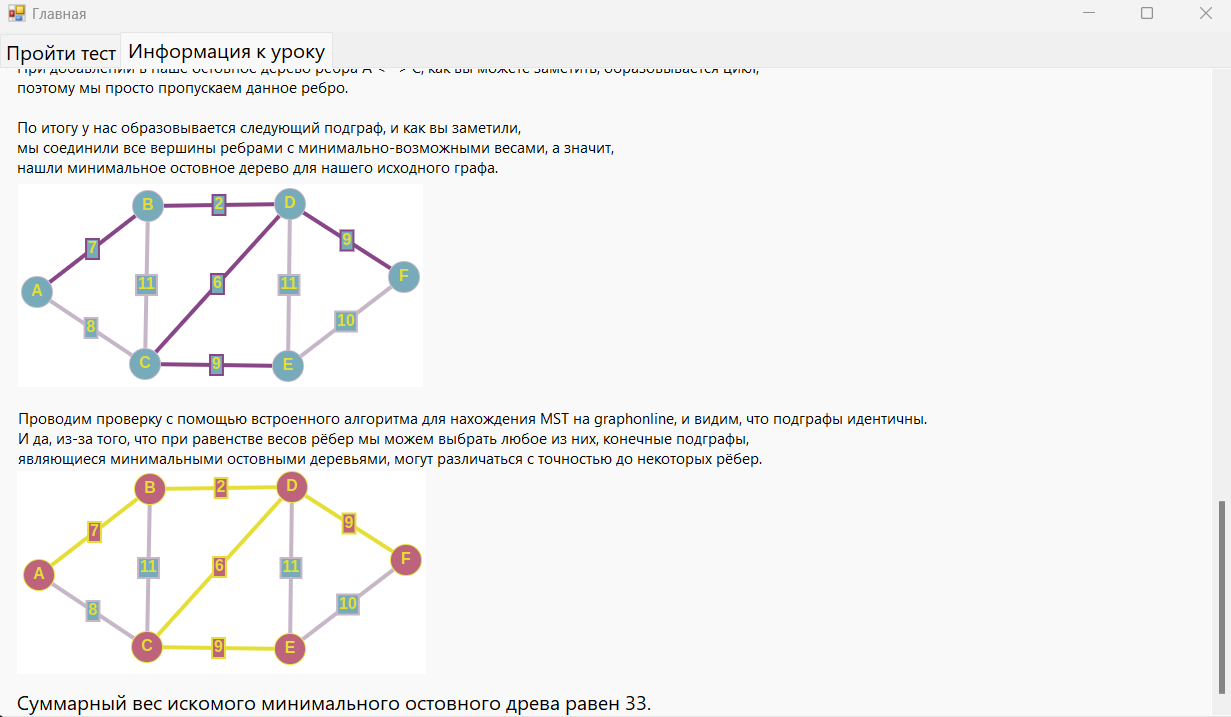


Рисунок 2.14 – Информация по теме, часть 4

Далее, нажав на кнопку «Пройти тест», студент попадет на соответствующую форму. Предполагается, что студент еще не проходил его. В противном случае он окажется недоступен, пока преподаватель его не разблокирует вновь.

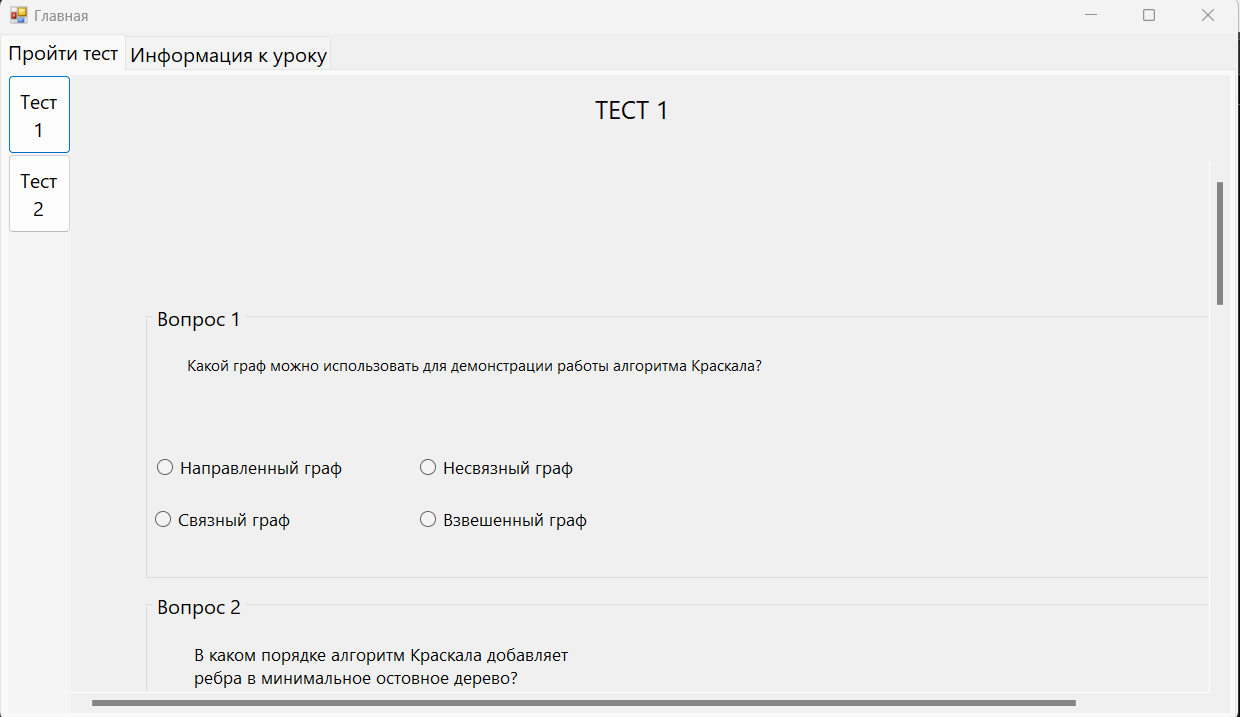


Рисунок 2.15 - Тест

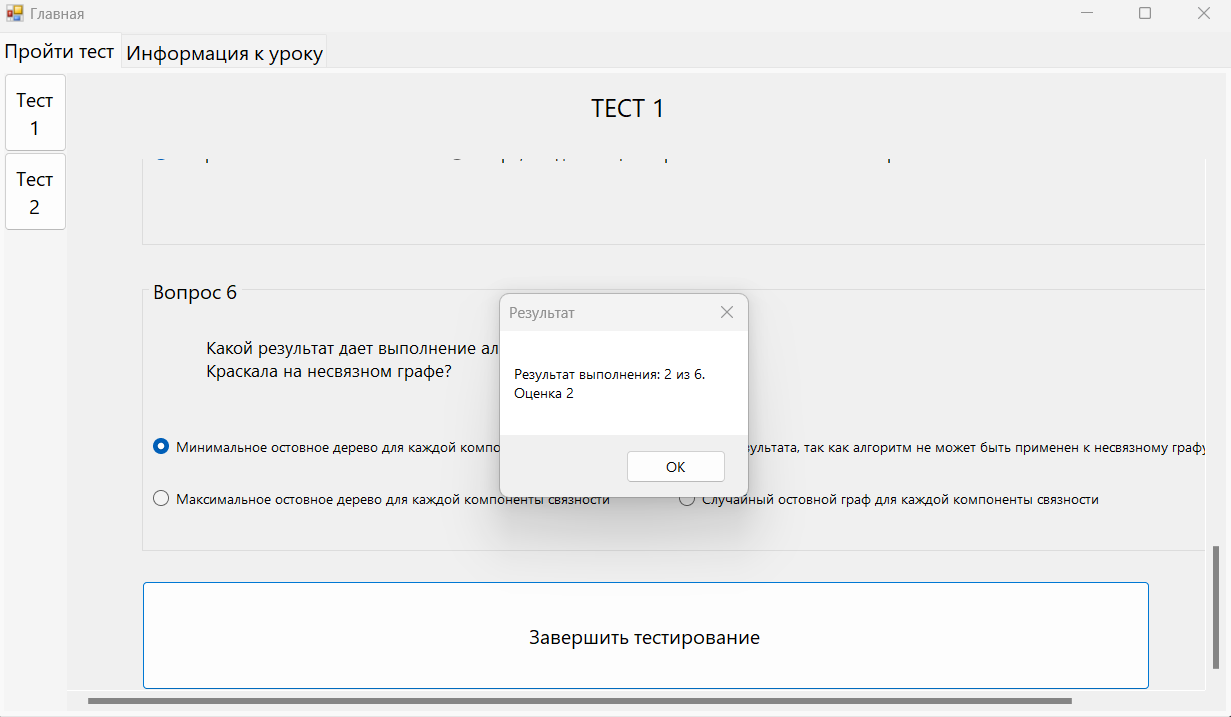


Рисунок 2.16 – Завершение теста, уведомление о его результате.

После прохождения студентом тестирования, его результат автоматически попадает в базу данных.

Рассмотрим ту ситуацию, когда авторизуется преподаватель. В этом случае он также имеет возможность изменить данные для авторизации студентов, нажав на кнопку «Управление ЛК», выбрав ID студента, и написав новый пароль в соответствующей форме, либо добавить нового пользователя.

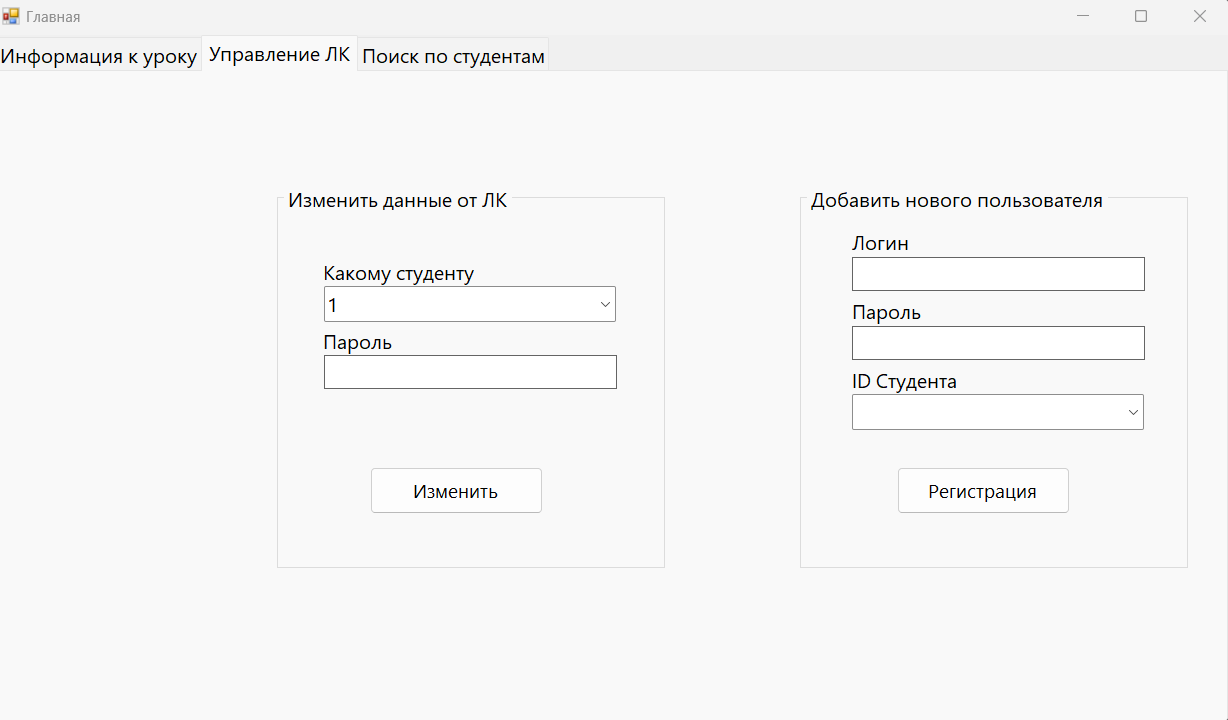


Рисунок 2.17 – Управление авторизацией

Кроме того, преподаватель имеет возможность просмотреть всех студентов, которые числятся в базе данных. Для этого ему следует перейти на форму «Поиск по студентам». По умолчанию, если преподаватель не выбирает конкретное ID, будет выведена информация по всем учащимся.

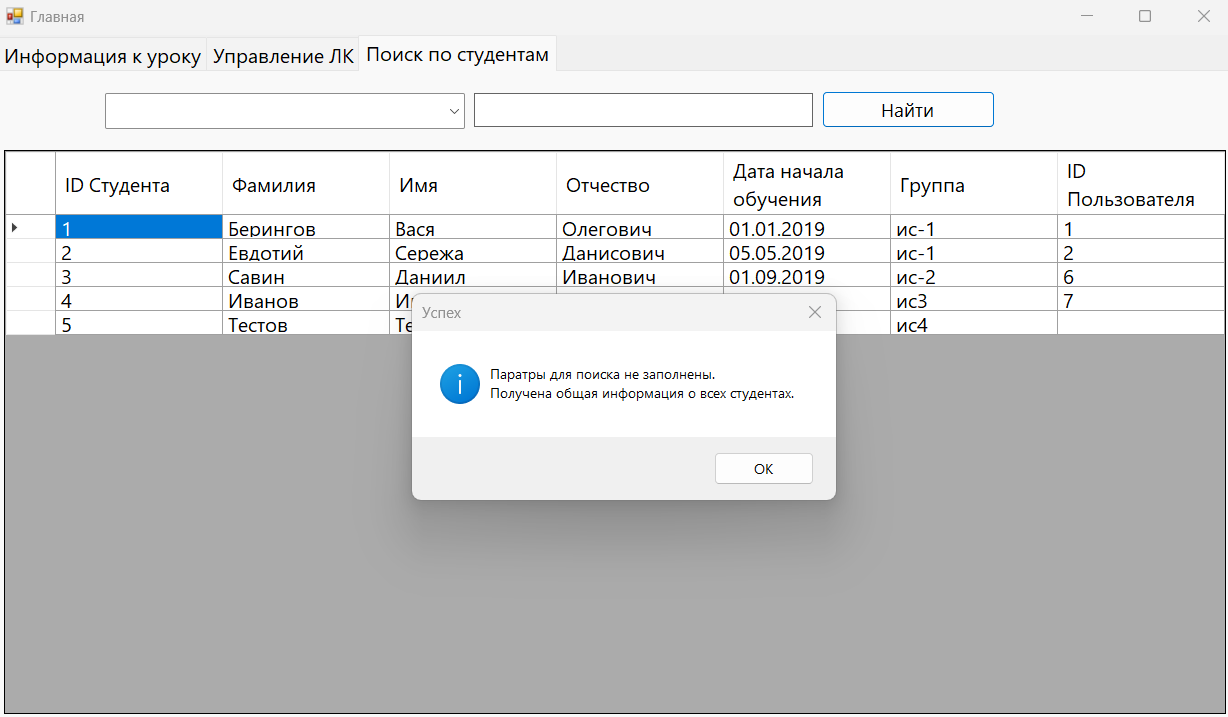


Рисунок 2.18 – Вывод информации обо всех студентах

В ином случае, при выборе конкретного ID, преподаватель получит информацию о соответствующем студенте.

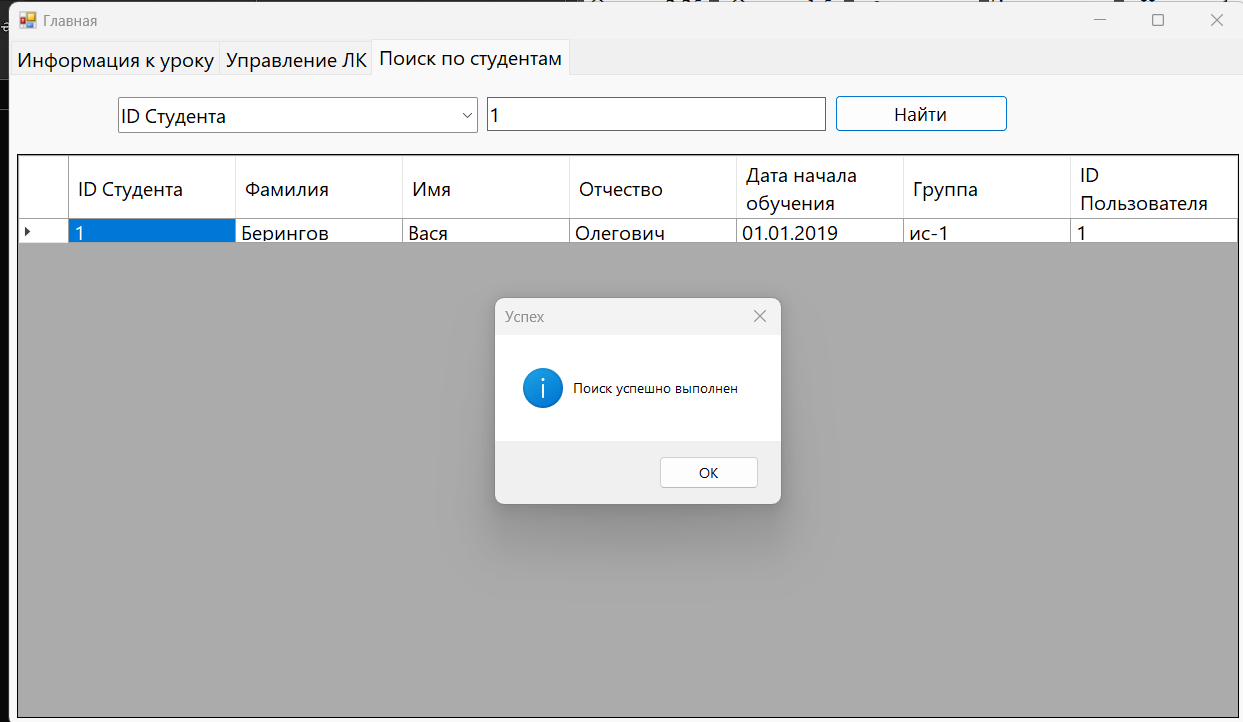


Рисунок 2.19 – Вывод информации о конкретном студенте

# Заключение

В данном курсовом проекте было разработано электронное учебное пособие для кафедры ИСТ. Целью курсового проекта было создание удобного и эффективного инструмента для обучения и тестирования студентов.

В ходе работы были проведены анализ и изучение требований учебного процесса. На основе этих данных была разработана концепция приложения, определены его функциональные возможности и архитектура.

В рамках курсового проекта была проведена разработка и реализация самого приложения. Была использована современная платформа для программирования Microsoft Visual Studio.

Данный проект включает в себя теоретическую информацию по заданной теме и тест, а также возможность авторизации студентов и преподавателей, которая позволит идентифицировать пользователей. Кроме того, поиск и добавление данных поможет быстро находить информацию и поддерживать ее актуальность.

В будущем проект будет дорабатываться, функционал будет расширен добавлением практической части, планируется проработка дизайна.

# Библиография

1) Анализ предметной области [Электронный ресурс] – Режим доступа: <https://metanit.com/sql/tutorial/1.1.php>

2) Предметная область [Электронный ресурс] – Режим доступа: <https://studbooks.net/2271976/informatika/model_predmetnoy_oblasti?ysclid=lb1dcfwm6o329311600>

3) Разработка web-приложений [Электронный ресурс] – Режим доступа: <https://bestprogrammer.ru/izuchenie/kak-sozdat-svoe-pervoe-prilozhenie-windows-forms-winforms-v-visual-studio?ysclid=lb1dh0n0xv116238613>

4) Информация по теме [Электронный ресурс] – Режим доступа: https://brestprog.by/topics/mst/

# Приложение А

(справочное)

Листинг программы

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace App

{

public partial class FormMain : Form

{

public FormMain()

{

InitializeComponent();

}

public void FormMain\_Load(object sender, EventArgs e)

{

//грузим данные во все combobox

SetDataComboBox();

//Сделаем стартовый индекс у комбобокса на форме поиска студентов

comboBox3.SelectionStart = 0;

checkRole(PersonalData.Role);

//установим доступность кнопок в зависимости от пройденого теста

isPassed();

}

private void SetDataComboBox()

{

var db = new DB();

//студенты для изменения данных для ЛК

var StudentForUpdate = "select с.[id студента] from Студенты с inner join Пользователи п on с.[id пользователя]=п.[id пользователя]";

comboBox2.Items.Clear();

db.loadElementToComboBox(StudentForUpdate, "id студента", comboBox2);

//для добавление новой учетки студенту. Грузим только студентов без учеток.

comboBox1.Items.Clear();

var queryAddNewUser = "select \* from студенты where [id пользователя] is null";

db.loadElementToComboBox(queryAddNewUser, "id студента", comboBox1);

}

private void isPassed()

{

var db = new DB();

using (SqlConnection connection = new SqlConnection(db.stringCon()))

{

connection.Open();

using (SqlCommand command = new SqlCommand($"select [ID Теста] from информация where [ID Студента] = {PersonalData.IdStudentOrTeacher} and Оценка = 0", connection))

{

using (SqlDataReader reader = command.ExecuteReader())

{

List<int> testIds = new List<int>();

//если есть записи, а они должны быть. (Под каждого студента и тест. Иначе записей не будет и кнопки будут недоступны.)

while (reader.Read())

{

if (!reader.IsDBNull(reader.GetOrdinal("ID Теста")))

{

//записываем в лист

int testId = Convert.ToInt32(reader["ID Теста"]);

testIds.Add(testId);

}

}

//делаем кнопки доступными

if (testIds.Contains(1))

{

btn1.Enabled = true;

}

if (testIds.Contains(2))

{

btn2.Enabled = true;

}

}

}

}

}

private string checkRole(string role)

{

switch (role.ToUpper())

{

case ("ПРЕПОДАВАТЕЛЬ"):

tabPage1.Dispose();

MessageBox.Show($"Добро пожаловать", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

return role;

case ("УЧЕНИК"):

tabPage3.Dispose();

tabPage4.Dispose();

MessageBox.Show($"Добро пожаловать", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

return role;

default:

tabControl1.Dispose();

MessageBox.Show("Учетная запись некорректна.\rОбратитесь в тех. поддержку.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return role;

}

}

public void button1\_Click(object sender, EventArgs e)

{

var db = new DB();

//убедимся, что студент еще не проходил первый тест

if (db.GetSinglValues($"select top 1 оценка from информация where [id теста]=1 and оценка<>0 and [id студента]={PersonalData.IdStudentOrTeacher}", "оценка") != "")

{

btn1.Enabled = false;

MessageBox.Show($"Вы уже прошли тест", "Уведомление", MessageBoxButtons.OK);

}

else

{

//на всякий случай очистим панель для форм

acriveForm = null;

openForm(new Test\_1());

}

}

private Form acriveForm = null;

private void openForm(Form childForm)

{

//Для открытия формы с тестом в панеле

if (acriveForm != null)

acriveForm.Close();

acriveForm = childForm;

childForm.TopLevel = false;

childForm.FormBorderStyle = FormBorderStyle.None;

childForm.Dock = DockStyle.Fill;

panel2.Controls.Add(childForm);

panel2.Tag = childForm;

childForm.BringToFront();

childForm.Show();

}

private void btn2\_Click(object sender, EventArgs e)

{

var db = new DB();

//убедимся, что студент еще не проходил первый тест. Укажем номер теста ручками, оценку и id студента - автоматически.

if (db.GetSinglValues($"select top 1 оценка from информация where [id теста]=2 and оценка<>0 and [id студента]={PersonalData.IdStudentOrTeacher}", "оценка") != "")

{

btn2.Enabled = false;

MessageBox.Show($"Вы уже прошли тест", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

//на всякий случай очистим панель для форм

acriveForm = null;

openForm(new Test\_2());

}

}

private void button7\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(textBox3.Text) || comboBox2.SelectedIndex == -1)

{

MessageBox.Show("Пароль не указан или студент не выбран.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

var db = new DB();

var query = $"update п " +

$" set Пароль='{textBox3.Text}' " +

$" from Пользователи п " +

$" inner join Студенты с on п.[ID Пользователя] = с.[ID Пользователя] " +

$" where [id студента]={comboBox2.SelectedItem}";

//если запрос выполнился и не вернулось null

if (db.queryExecute(query) != null)

{

SetDataComboBox();

MessageBox.Show("Пароль успешно обновлен", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

private void button6\_Click(object sender, EventArgs e)

{

//проверка на пустоту и тд

if (string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text) || comboBox1.SelectedIndex == -1)

{

MessageBox.Show("Заполните все поля.", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

var db = new DB();

var queryInsert = "insert into Пользователи (Логин, пароль, роль) " +

$"values('{textBox1.Text}','{textBox2.Text}','Ученик')";

//если запрос выполнился и не вернулось null

if (db.queryExecute(queryInsert) != null)

{

var queryUpdate = "update Студенты set [id пользователя]=(" +

" select top 1 [id Пользователя] from пользователи order by [id пользователя] desc" +

" ) " +

$" where [id студента]={comboBox1.SelectedItem} ";

//если запрос выполнился и не вернулось null

if (db.queryExecute(queryUpdate) != null)

{

SetDataComboBox();

MessageBox.Show("Новый пользователь успешно зарегистрирован.", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

private void button4\_Click(object sender, EventArgs e)

{

var db = new DB();

//проверим что данные для поиска не пустые, если пустые, то выведем все из таблицы

if (comboBox3.SelectedIndex == -1 || string.IsNullOrEmpty(textBox5.Text))

{

var queryAllSeatch = $"select \* from Студенты";

//если запрос выполнился и не вернул null

if (db.queryReturnData(queryAllSeatch, dataGridView1) != null)

{

MessageBox.Show("Паратры для поиска не заполнены.\r\nПолучена общая информация о всех студентах.", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

return;

}

var querySearch = $"select \* from Студенты where [{comboBox3.SelectedItem}]={textBox5.Text}";

//если запрос выполнился и не вернул null

if (db.queryReturnData(querySearch, dataGridView1) != null)

{

MessageBox.Show("Поиск успешно выполнен", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

}