|  |
| --- |
| **Министерство сельского хозяйства и продовольствия Республики Беларусь**  **УО «Столинский государственный аграрно-экономический колледж»**  **Отделение «Бухгалтерский учет и информационные технологии»**  ДОПУЩЕН К ЗАЩИТЕ  Заведующий учебно-производственной практикой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.С. Ярмошик  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.  **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  **ПРОЕКТА**  «  Тестирование знаний »  (Тема проекта)  Специальность  2-40 01 01 «Программное обеспечение информационных технологий»  Специализация  2-40 01 01 35 «Программное обеспечение обработки экономической и деловой информации»  Учащийся  группа П-16 Флорьянович В. Л.  (подпись) (Ф.И.О.)  Руководитель  (подпись) (Ф.И.О.)  Отметка  Столин 2022 |
| Министерство сельского хозяйства и продовольствия Республики Беларусь  УО «Столинский государственный аграрно-экономический колледж  **ЗАДАНИЕ**  на выполнение проекта учащемуся  *Флорьяновичу Владиславу Леонидовичу*  (Фамилия, Имя, Отчество)   1. Тема работы: *Тестирование знаний* 2. Исходные данные к проекту    1. *Создать программу, позволяющую составлять и проходить тесты с набором вопросов, имеющих свой рейтинг, составить тест можно автоматически на выбранную из списка тему, там и будет использоваться рейтинг.*    2. *оценить знания учащегося и запомнить результат* 3. Состав проекта    1. Содержание расчётно-пояснительной записки (перечень подлежащих разработке вопросов)   *Создать окно авторизации и регистрации учеников и учителей. Создать программный продукт, позволяющий составлять тесты. Создать функцию автоматического создания тестов, реализуя создание тестов по рейтингу вопросов. Дать возможность добавлять вопросы. Создать возможность прохождения тестов ученикам. Создать функцию оценки знаний теста. Предоставить меры безопасности от возможного жульничества.*   1. Дата выдачи задания \_\_\_\_\_\_\_\_\_\_\_2022г. 2. Срок сдачи учащимся законченной работы \_\_\_\_\_\_\_\_\_\_\_2022г.   Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ф.И.О.  (подпись)  Задание принял к исполнению  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_20\_\_г. Флорьянович В. Л.  *Ф.И.О* | |
| **РЕФЕРАТ**  Пояснительная записка проекта: 41 с, 30 рис., 6 табл.  Объектом разработки является: обучение учащихся.  Цель проекта: программа тестирования знаний учащихся.  В процессе проектирования выполнены следующие разработки:  - Проанализировано тестирование учащихся.  - Предусмотрено безопасность от жульничества.  Элементами практической значимости полученных результатов являются:  - Добавление дополнительных мер безопасности  - Структурирование интерфейса программы  Областью возможного практического применения являются:  - Разработка программы  - Внедрение особенностей тестирования в другие учебные заведения  В ходе проектирования прошли апробацию такие предложения, как:  - Создание 3 баз данных  - Разделение полномочий на учителя и ученика  - Сохранение оценок в БД  Результатами внедрения явились:  - Дискретная система БД  - Возможность создавать тесты учителю, не ученикам  - Мера защиты от жульничества  Учащийся-дипломник подтверждает, что приведенный в проекте расчетно-аналитический материал объективно отражает состояние разрабатываемого объекта, все заимствованные из литературных и других источников теоретические и методологические положения и концепции сопровождаются ссылками на их авторов. | |
| **СОДЕРЖАНИЕ**  Введение…………………………………………………………………………………………5  Анализ исходных данных и постановка цели и задач…………………. …………………………………6  Техническое задание……………………………………………………………………………………...7  Проектирование программного обеспечения………………………………...……………..10  Кодирование модулей системы………………………………………………………………20  Реализация …………………………………………………………………………………….30  Тестирование…………………………………………………………………………………..38  Заключение…………………………………………………………………………….……….41 | |

**ВВЕДЕНИЕ**

Основной причиной создания приложения является практическая работа.

Планируется выполнить курсовой проект на тему: «Тестирование знаний». Выбор данной темы курсового проекта основывается на востребованности пользования подобным программным обеспечением в учебных заведениях.

Автоматизация рабочего места заключается во внедрении средств, позволяющих в большей части заменить, ускорить и облегчить трудовую деятельность человека.

Автоматизации рабочего места учителя достигается путем его оснащения ЭВМ с соответствующим программным обеспечением, называемым «Тестирование знаний». Это программное обеспечение должно позволять: хранить тесты по заданным темам, создавать новые тесты как вручную так и автоматически и редактировать уже созданные, проходить тесты, оценивать тесты.

В основной круг пользователей программы будут входить учителя и ученики, оснащенные ЭВМ. Кроме этого, программа может применяться и в других учебных заведениях для прохождения теста и оценки знаний.

Основная цель проекта – создание автоматизированного рабочего места тестирования знаний.

* Для ее достижения необходимо решить следующие вопросы:
* создание базы данных тестов;
* разработка СУБД для осуществления работы с базой данных;
* обеспечение авторизации;
* создание редактора тестов;
* создание прохождения тестов;
* обеспечение объединения и взаимодействия всех разработанных средств в программе;
* построение несложного и удобного для работы интерфейса.

**Анализ исходных данных и постановка цели и задач.**

Программа создания универсальной тестирующей оболочки, содержащей рейтинговые вопросы и позволяющая по десятибалльной системе оценить знания учащегося – в данное время есть множество интернет аналогов тестов, но те что создают и проходят тесты я знаю всего 1. Есть одно, но, все они требуют интернет подключение. Моя же программа будет полностью автоматизирована. А значит ей могут пользоваться все учебные заведения установив его лишь раз.

Данная программа создается посредством решения следующих задач:

* Создать окно авторизации и регистрации учеников и учителей.
* Создать программный продукт, позволяющий составлять тесты.
* Создать функцию автоматического создания тестов, реализуя создание тестов по рейтингу вопросов.
* Дать возможность добавлять вопросы.
* Создать возможность прохождения тестов ученикам.
* Создать функцию оценки знаний теста.
* Предоставить меры безопасности от возможного жульничества.

Изучаемым объектом стал обычный тест в школах и их вариации: на листочке, онлайн. Повторяясь, ни один из них не мог автоматически проверять ответы и не использовать интернет. Оба варианта подверглись анализу.

Из предыдущего следует:

Решение данных задач способствует созданию заданной программы с учетом всех дополнительных условий

**Техническое задание**

**Формулировка задания:**

Разработать программный продукт (ПП), предназначенный для создания и прохождения тестов. ПП должен обеспечивать ведение базы данных, автоматическое создание тестов, рейтинг вопросов, оценку знаний учащихся.

**Категории пользователей:**

ПП должен быть рассчитан на преподавателя, учащегося.

**Наименование организации заказчика:**

Заказчиком на разработку данного ПП является УО «СГАЭК».

(220393 г. Столин, ул. Калинина 12/ 12, тел/ факс 278-65-42).

**Основания для проведения работ:**

Основанием для разработки данного ПП является практика «ТРПО».

**Описание процессов:**

* ПП поддерживает функции авторизации и регистрации.
* ПП должен иметь возможность создавать тесты и сохранять их в БД, также должна присутствовать возможность автоматически создавать тесты на основе рейтинга вопросов.
* Рейтинг вопроса вычисляется по кол-ву неправильных ответов, преподаватель может изменить рейтинг вручную и добавить вопросы вручную.
* Учащийся может пройти тест и получить оценку.

**Состав данных и алгоритмы обработки информации**:

Входными данными для разрабатываемого ПП являются:

* Пароль и имя преподавателя;
* пароль и и имя учащегося;
* параметры настройки, перечисленные в разделе «Функциональные требования»;
* данные, вводимые пользователем;
* вопросы и ответы, введенные преподавателем;
* ответы, введенные учеником.

Выходными данными разрабатываемого ПП являются:

* информация, отображаемая на экран по мере написания вопроса;
* оценка ученика.

Постоянными данными являются:

* шаблоны документов,
* БД.

**Разрабатываемый ПП должен создавать и модифицировать перечисленные ниже БД:**

БД тестов, содержащая две и более таблиц:

* таблицы тестов, созданных вручную. Названиями таблиц - являются темы.

БД авторизации, содержащая две таблицы:

* таблица авторизации учителей и учеников;

БД вопросов, содержащая две и более таблиц:

* таблица вопросов и их рейтинг.

Текущий уровень автоматизации

На данный момент Заказчик не располагает каким-либо ПП, обеспечивающим полноценную автоматизацию всей его деятельности. Заказчик располагает оборудованием:

* 5 компьютеров Р-III-500 (256RAM RIMM), Unix Free BSD, Star office; – 4 компьютера П-133 (16RAM 4VRAM), Windows 98, Microsoft office

97.

**Информационная модель:**

Информационная модель представлена на рисунке 3.1.



Рисунок 3.1 – Информационная модель проекта

**Структура меню:**

Справка

**Функциональные требования:**

Разрабатываемое ПО должно реализовать следующие функции

Ведение базы данных по тестам.

Функция предназначена для:

* Сохранения тестов;

Ведение базы данных по авторизации.

Функция предназначена для:

* Предоставления правильного уровня допуска;
* Сохранения вопросов и их рейтинга;
* Учет сдачи тестов;

Авторизация предназначена для контроля допуска Требования к информационному обеспечению

Разрабатываемый ПП должен:

* сохранять данные, вводимые пользователем;
* содержать необходимые таблицы;
* обеспечивать обмен данными с другими ПП по локальной сети.

**Требования к пользовательскому интерфейсу**:

Разрабатываемый ПП должен быть обеспечен упрощенной формой интерфейса:

* глубина вложенности окон должна быть не более трех;
* содержать краткие описания элементов меню;
* должен быть понятным и простым в использовании.

Панель интерфейса должна содержать кнопку, предназначенную для сохранения необходимой информации

**Требования к алгоритмам:**

Разрабатываемые алгоритмы должны легко модифицироваться. В течение сроков, установленных в договоре между Заказчиком и Разработчиком в ПП, могут быть внесены изменения, не требующие глобальной переработки алгоритмов.

**Прочие требования:**

Разрабатываемый ПП должен функционировать в ОС Win 10, NT, 2K.

Другие ОС поддерживают формат создаваемых файлов в процессе работы с ПП.

Хранение данных пользователя должно производиться в файлах с встроенной защитой, не подлежащих удалению без пароля Администратора.

ПП должен использовать нижние регистры памяти, что обеспечивает защиту от сбоев на 95 %.

ПП должен обеспечивать сетевой обмен информацией с другими БД. При создании БД происходит резервное сохранение информации, вводимой пользователем в файлы с расширением «mdb». Сетевой обмен происходит за счет передачи/ получения файлов с таким же расширением от других систем.

**Экспертиза:**

Не требуется разработка макета на стадии «Технический проект». Требуется проведение экспертизы при участии представителя Заказчика.

**Тестирование:**

Тестирование должно проводиться Разработчиком в соответствии с установленными правилами. Присутствие представителя Заказчика необязательно.

Тестирование проводится Разработчиком по предложенной схеме Заказчика:

1. создается БД небольшого объема по всем таблицам и пунктам меню;
2. в течение срока, указанного в договоре, Разработчик проводит тестирование ПП и передает Заказчику готовый программный продукт, который будет сопровождаться Разработчиком в течении сроков, предусмотренных договором.

**Опытная эксплуатация**:

Опытная эксплуатация проводится Заказчиком, при контроле Разработчика, в соответствии с договором.

**Требования к справочной системе:**

Справочная система должна содержать описание всех команд, использующихся в ПП. Справочная система должна быть встроенной в интерфейс и располагаться непосредственно в окнах.

**Требования к документации пользователя**:

Разрабатываемый ПП должен сопровождаться полным пакетом конструкторской и эксплуатационной документации, перечисленной в договоре.

Разработанный ПП сопровождается полным пакетом документов, предусмотренным ГОСТ 7685 – 2000, а именно: лицензионное право использования, разработанного ПП; документация по эксплуатации; талон на последующее обслуживание ПП; документы, подтверждающие монопольное использование разработанного ПП.

**Проектирование программного обеспечения**

**Обоснование выбора языка программирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| Ключевые критерии | С# | Html | Object Pascal |
| Знания языка | + | - | +- |
| Опыт | + | - | - |
| Работа с БД | + | - | + |
| Объектно-ориентированный | + | - | + |
| Высокоуровневый | + | - | + |
| Скорость разработки программы | + | - | - |
| Скорость работы | + | - | - |
| Кроссплатформенность | + | + | - |
| Строгая типизация | + | - | + |
| Работа с памятью | + | - | - |
| Скорость тестирования | + | - | - |

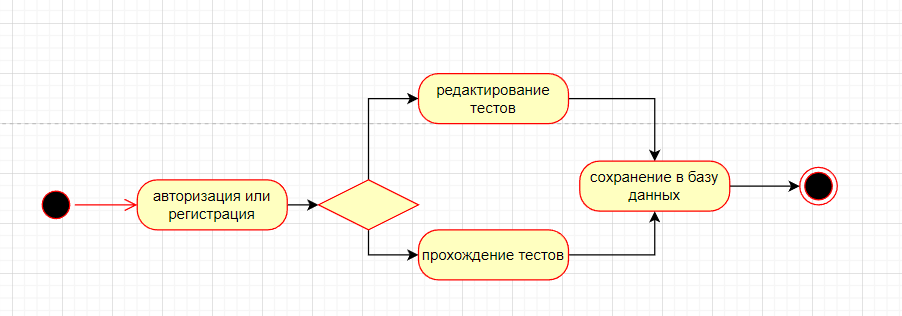
**Обоснование выбора среды разработки:**

|  |  |  |  |
| --- | --- | --- | --- |
| Ключевые критерии | Visual Studio | Delphi | Notepad ++ |
| Поддержка С# | + | - | + |
| Поддержка форм | + | + | - |
| Работа с БД | + | + | - |
| Встроенный терминал | + | + | - |
| Удобность интерфейса | + | - | - |
| Помощь при кодировании | + | + | - |
| Поддержка | + | - | - |

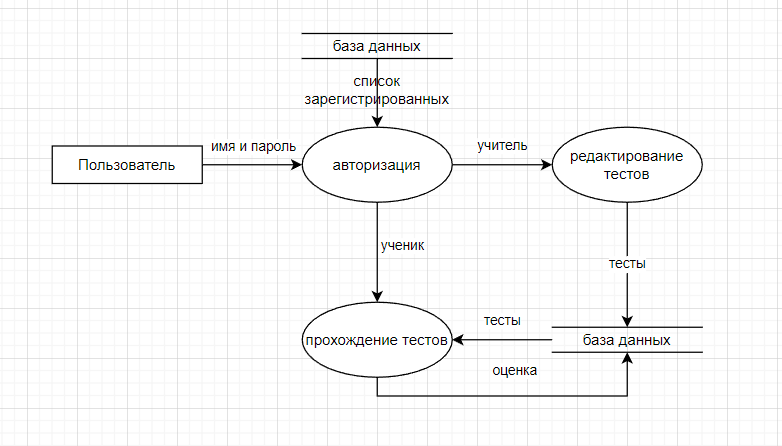
**Обоснование выбора базы данных:**

|  |  |  |  |
| --- | --- | --- | --- |
| Ключевые критерии | Ms SQL | My SQL | Ms Access |
| Производительность | + | + | - |
| Надежность | + | + | - |
| Легкость установки | + | - | + |
| Удобность использования | + | + | - |
| Удобность интерфейса | + | + | - |
| Работа с сервером | + | + | - |
| Поддержка | + | - | - |

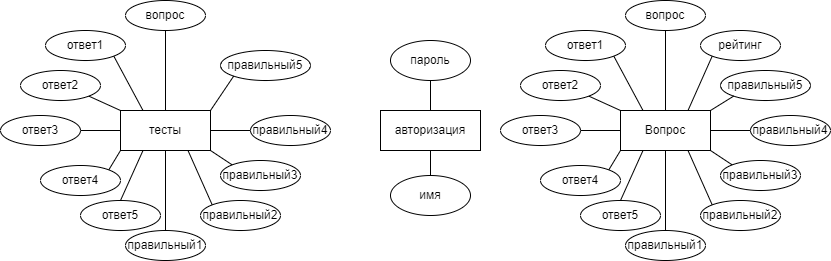
**Модель бизнес-процесса**



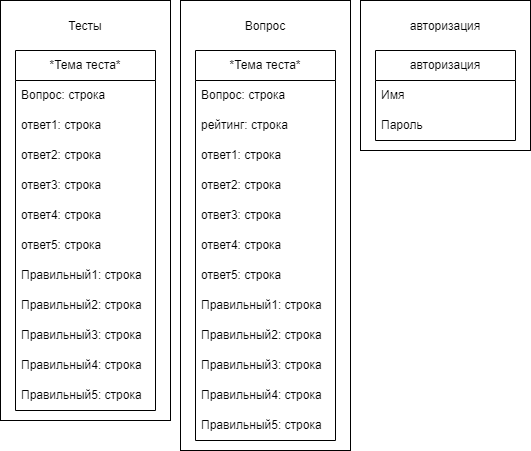
**Функциональная модель**



**Модель базы данных**

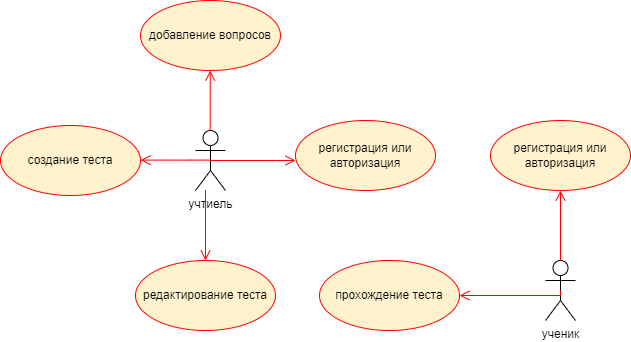


**Модель базы данных (Нотация Чена)**

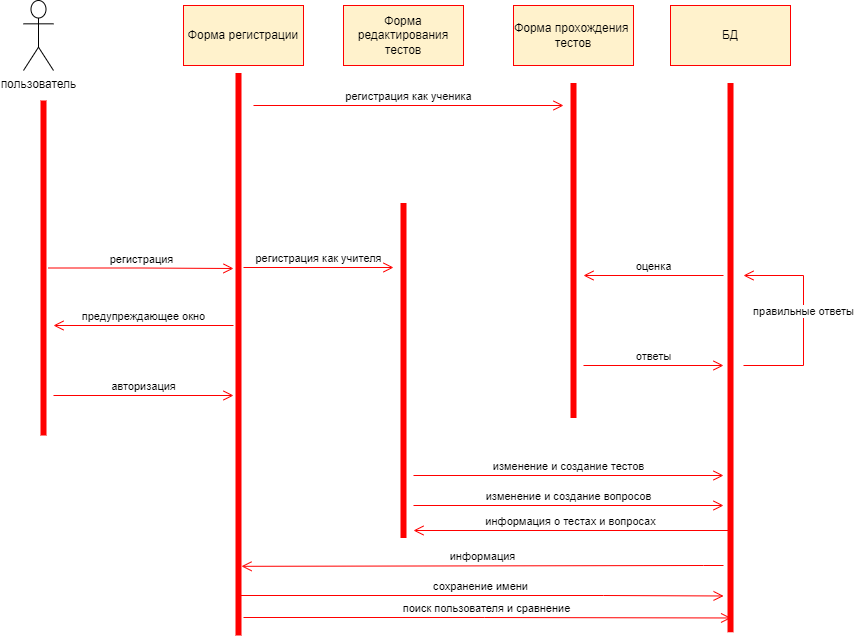


**Концептуальная модель UML**

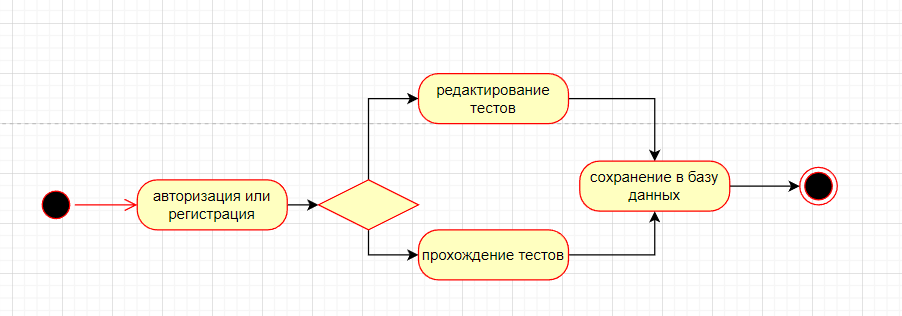
вариантов использования



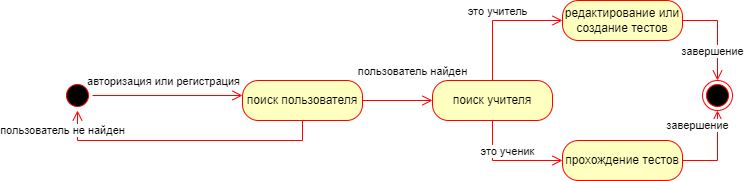
последовательности

****

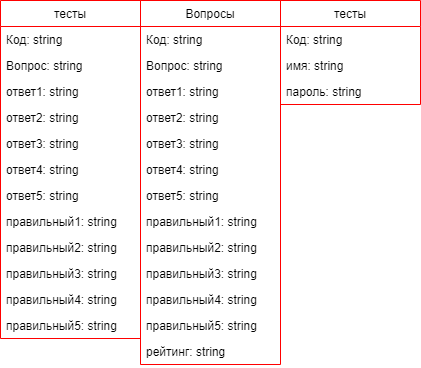
деятельности



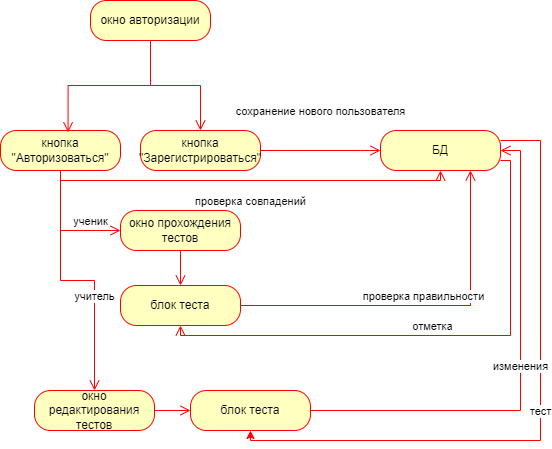
состояния



классов



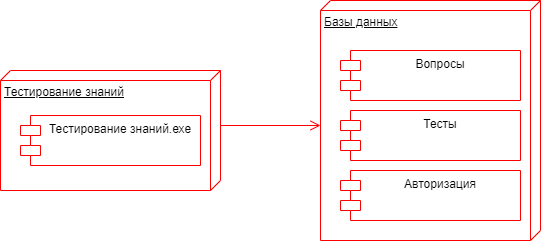
объектов



компонентов

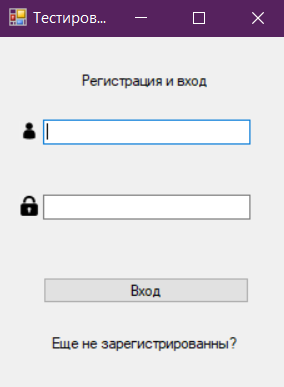


развертывания

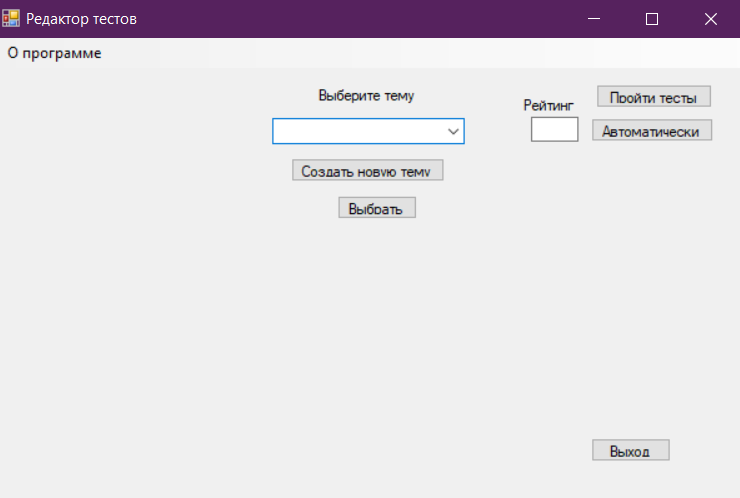


**Проектирование интерфейса:**

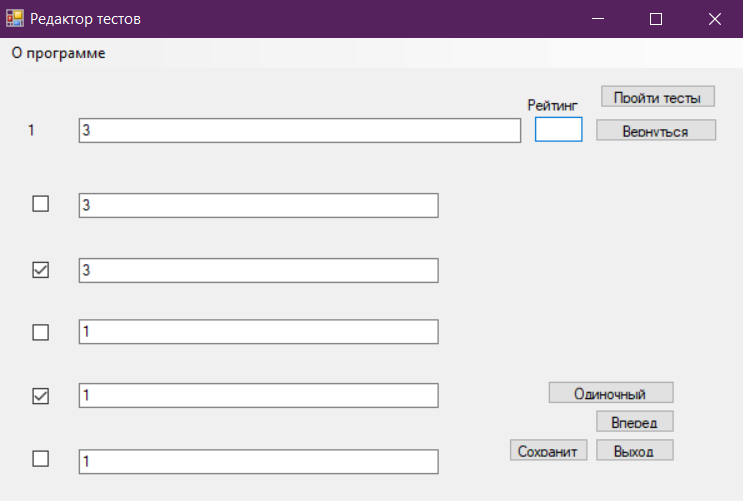
Окно авторизации



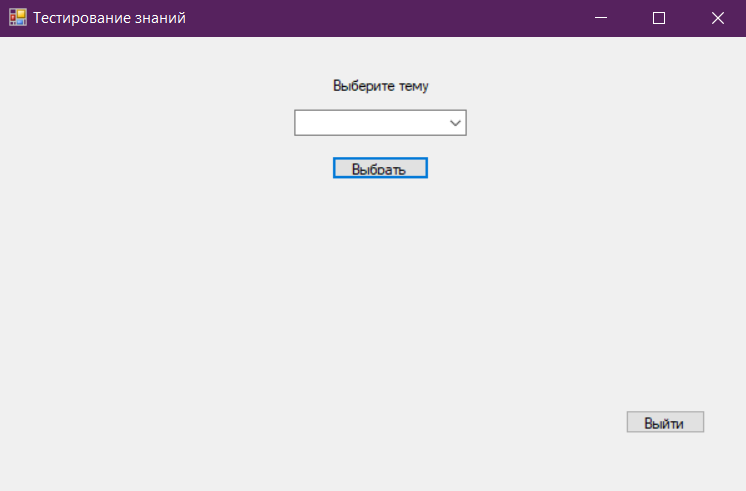
Окно редактирования тестов (Выбор темы теста или создание теста)



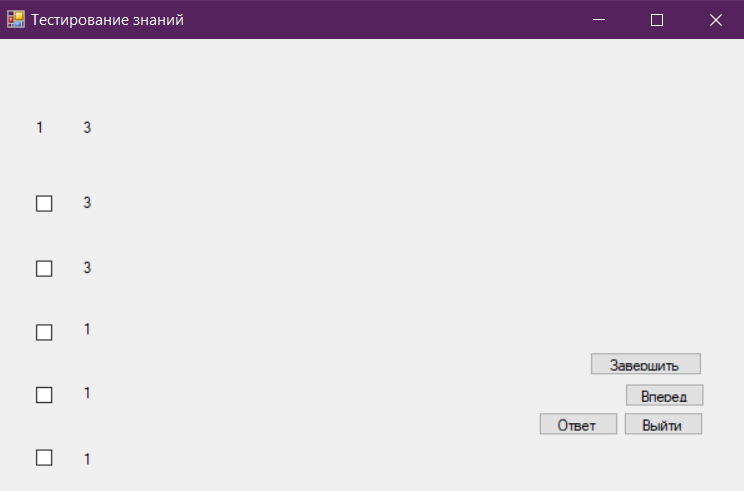
Окно редактирования тестов (Редактирование теста или создание теста)



Окно тестирования знаний (Выбор темы)



Окно тестирования знаний (Прохождение теста)



**Кодирование модулей системы**

**1 Окно(Форма): Авторизация**

using System;

using System.Configuration;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Тестирование\_знаний

{

public partial class Form1 : Form

{

public string text = "";

public bool inp = true;

public SqlConnection sqlConnection = new SqlConnection(ConfigurationManager.ConnectionStrings["Autorisation"].ConnectionString);

//Поиск айди по Имени

public int Search(string text)

{

string query = $"SELECT Id FROM Авторизация WHERE Пользователи = N'{text}'";

try

{

SqlCommand sqlCommand = new SqlCommand(query,sqlConnection);

return Convert.ToInt32(sqlCommand.ExecuteScalar().ToString());

}

catch

{

return -1;

}

}

//Поиск пароля по индексу

public string Searchpass(int i)

{

string query = $"SELECT Пароль FROM Авторизация WHERE Id = {i}";

SqlCommand command = new SqlCommand(query, sqlConnection);

try

{

return command.ExecuteScalar().ToString();

}

catch

{

return "";

}

}

//админ ли это?

public bool Admin(int i)

{

string query = $"SELECT Администратор FROM Авторизация WHERE Id = {i}";

try

{

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

return 1 == Convert.ToInt16(sqlCommand.ExecuteScalar().ToString());

}

catch

{

return false;

}

}

public Form1()

{

InitializeComponent();

}

//Изменение Регистрации и Входа

private void label2\_Click(object sender, EventArgs e)

{

if (inp)

{

button1.Text = "Регистрация";

label2.Text = "Уже зарегистрированны?";

inp = false;

}

else

{

button1.Text = "Вход";

label2.Text = "Еще не зарегистрированны?";

inp = true;

}

}

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

{

text = textBox1.Text.Replace(' ', '\_');

//Если написано правильно

if (textBox1.Text != "" && textBox2.Text != "")

{

// Вход

if (inp)

{

//Если такой есть

if (textBox2.Text.Replace(' ','\_') == Searchpass(Search(textBox1.Text.Replace(' ', '\_'))))

{

//Если Учитель

if (Admin(Search(textBox1.Text.Replace(' ', '\_'))))

{

Form2 form2 = new Form2();

form2.text = this.Text;

form2.Show();

this.Hide();

}

//Если Ученик

else

{

Form3 form3 = new Form3();

form3.text = text;

form3.Show();

this.Hide();

}

}

//Если нет пользователя

else

MessageBox.Show("Такого пользователя нет", "Error");

}

//Если регистрация

else

{

//Если нет такого имени

if (Search(textBox1.Text.Replace(' ', '\_')) < 0)

{

string query = $"INSERT INTO Авторизация (Пользователи, Пароль) VALUES (N'{textBox1.Text}', N'{textBox2.Text}')";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlCommand.ExecuteScalar();

}

//Если уже есть

else

MessageBox.Show("Такой пользователь уже существует", "Error");

}

}

//Если неправильно ввели

else

{

if (textBox1.Text == "")

textBox1.BackColor = Color.RosyBrown;

if (textBox2.Text == "")

textBox2.BackColor = Color.RosyBrown;

MessageBox.Show("Ваполните все строки", "Error");

}

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

private void Form1\_Load(object sender, EventArgs e)

{

sqlConnection.Open();

}

//Изменение цвета при вприсывании

private void textBox1\_TextChanged(object sender, EventArgs e)

{

textBox1.BackColor = SystemColors.Window;

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

textBox2.BackColor = SystemColors.Window;

}

}

}

**2 Окно(Форма): Редактор тестов**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Тестирование\_знаний

{

public partial class Form2 : Form

{

public string text = "";

public int[,] strings = new int[1, 2];

public int numerator = 0;

public SqlConnection sqlConnectiont = new SqlConnection(ConfigurationManager.ConnectionStrings["Test"].ConnectionString);

public SqlConnection sqlConnectionq = null;

public int Search()

{

string query = $"SELECT Id FROM {textBox7.Text} WHERE' Вопрос = N'{textBox1.Text}', Ответ1 = N'{textBox2.Text}', Ответ2 = N'{textBox3.Text}', Ответ3 = N'{textBox4.Text}', Ответ4 = N'{textBox5.Text}', Ответ5 = N'{textBox6.Text}'";

try

{

SqlCommand sqlCommand = new SqlCommand(query, sqlConnectionq);

return Convert.ToInt32(sqlCommand.ExecuteScalar().ToString());

}

catch

{

return -1;

}

}

public void Navigation()

{

label2.Text = numerator.ToString();

if (checkBox1.Size == new Size(0,0))

button9.Text = "Множественный";

else

button9.Text = "Одиночный";

int numans = 0;

TextBox[] textbox= new TextBox[] {textBox1, textBox2,textBox3, textBox4, textBox5, textBox6};

RadioButton[] radiobuttons = new RadioButton[] { radioButton1, radioButton2, radioButton3, radioButton4, radioButton5 };

CheckBox[] checkboxes = new CheckBox[] {checkBox1,checkBox2,checkBox3,checkBox4,checkBox5 };

string[] texts = new string[] { "Вопрос", "Ответ1", "Ответ2", "Ответ3", "Ответ4", "Ответ5" };

string[] answers = new string[] {"Правильный1", "Правильный2", "Правильный3", "Правильный4", "Правильный5" };

for (int i = 0; i < 6; i++)

{

string query = $"SELECT {texts[i]} FROM {textBox7.Text} WHERE Id = {numerator}";

textbox[i].Text = new SqlCommand(query, sqlConnectiont).ExecuteScalar().ToString();

}

for (int i = 0; i < 5; i++)

{

string query = $"SELECT {answers[i]} FROM {textBox7.Text} WHERE Id = {numerator}";

try

{

radiobuttons[i].Checked = Convert.ToBoolean(Convert.ToInt32(new SqlCommand(query, sqlConnectiont).ExecuteScalar().ToString()));

checkboxes[i].Checked = Convert.ToBoolean(Convert.ToInt32(new SqlCommand(query, sqlConnectiont).ExecuteScalar().ToString()));

}

catch

{

radiobuttons[i].Checked = false;

checkboxes[i].Checked = false;

}

if (checkboxes[i].Checked)

numans++;

}

if (numans > 1)

{

for (int i = 0; i <5; i++)

checkboxes[i].Size = new Size(17, 17);

}

else

{

for (int i = 0; i < 5; i++)

checkboxes[i].Size = new Size(0, 0);

}

if (checkBox1.Size == new Size(0,0))

button9.Text = "Множественный";

else

button9.Text = "Одиночный";

}

public void Next()

{

numerator++;

if (numerator == 10)

button5.Size = new Size(0, 0);

button6.Size = new Size(64, 19); ;

}

public void Prew()

{

numerator--;

if (numerator == 1)

button6.Size = new Size(0, 0);

button5.Size = new Size(64, 19);

}

//Переход

public void First()

{

label1.Visible = !label1.Visible;

label2.Visible = !label2.Visible;

checkBox1.Visible = !checkBox1.Visible;

checkBox2.Visible = !checkBox2.Visible;

checkBox3.Visible = !checkBox3.Visible;

checkBox4.Visible = !checkBox4.Visible;

checkBox5.Visible = !checkBox5.Visible;

radioButton1.Visible = !radioButton1.Visible;

radioButton2.Visible = !radioButton2.Visible;

radioButton3.Visible = !radioButton3.Visible;

radioButton4.Visible = !radioButton4.Visible;

radioButton5.Visible = !radioButton5.Visible;

button1.Visible = !button1.Visible;

if (checkBox1.Visible)

button2.Text = "Вернуться";

else

button2.Text = "Автоматически";

button5.Visible = !button5.Visible;

button6.Visible = !button6.Visible;

button7.Visible = !button7.Visible;

button8.Visible = !button8.Visible;

button9.Visible = !button9.Visible;

comboBox1.Visible = !comboBox1.Visible;

textBox1.Visible = !textBox1.Visible;

textBox2.Visible = !textBox2.Visible;

textBox3.Visible = !textBox3.Visible;

textBox4.Visible = !textBox4.Visible;

textBox5.Visible = !textBox5.Visible;

textBox6.Visible = !textBox6.Visible;

textBox7.Visible = !textBox7.Visible;

}

//сохранить вопрос

public void Add()

{

if (textBox1.Text.Length <= 50 && textBox2.Text.Length <= 50 && textBox3.Text.Length <= 50 && textBox4.Text.Length <= 50 && textBox5.Text.Length <= 50 && textBox6.Text.Length <= 50)

{

if (!checkBox1.Checked && !checkBox2.Checked && !checkBox3.Checked && !checkBox4.Checked && !checkBox5.Checked)

MessageBox.Show("Должен быть хотя бы 1 ответ");

else

{

if (textBox1.Text != "" && textBox2.Text != "" && textBox3.Text != "")

{

if ((textBox4.Text == "" && checkBox3.Checked) || (textBox5.Text == "" && checkBox4.Checked) || (textBox6.Text == "" && checkBox5.Checked))

MessageBox.Show("Нельзя ставить ответ на пустые вопросы");

else

{

if ((textBox4.Text == "" && textBox5.Text != "" && textBox6.Text != "") || (textBox5.Text == "" && textBox6.Text != ""))

MessageBox.Show("Нельзся, что бы ответы шли не по порядку");

else

{

int x = 0;

while (!int.TryParse(textBox8.Text, out x) && x >= 1 && x <= 20)

MessageBox.Show("Введите число от 1 до 20");

string queryt = $"UPDATE {textBox7.Text} SET Вопрос = N'{textBox1.Text}', Ответ1 = N'{textBox2.Text}', Ответ2 = N'{textBox3.Text}', Ответ3 = N'{textBox4.Text}', Ответ4 = N'{textBox5.Text}', Ответ5 = N'{textBox6.Text}', Правильный1 = N'{Convert.ToInt32(checkBox1.Checked)}', Правильный2 = N'{Convert.ToInt32(checkBox2.Checked)}', Правильный3 = N'{Convert.ToInt32(checkBox3.Checked)}', Правильный4 = N'{Convert.ToInt32(checkBox4.Checked)}', Правильный5 = N'{Convert.ToInt32(checkBox5.Checked)}' WHERE Id = {label2.Text}";

SqlCommand commandt = new SqlCommand(queryt, sqlConnectiont);

commandt.ExecuteScalar();

}

}

}

else

MessageBox.Show("Строки Вопроса, 1 Ответа, 2 Ответа обязательны");

}

}

else

MessageBox.Show("Длина строки не больше 50");

}

//добавить в Вопрос

public void Insert()

{

if (textBox1.Text.Length <= 50 && textBox2.Text.Length <= 50 && textBox3.Text.Length <= 50 && textBox4.Text.Length <= 50 && textBox5.Text.Length <= 50 && textBox6.Text.Length <= 50)

{

if (!checkBox1.Checked && !checkBox2.Checked && !checkBox3.Checked && !checkBox4.Checked && !checkBox5.Checked)

MessageBox.Show("Должен быть хотя бы 1 ответ");

else

{

if (textBox1.Text != "" && textBox2.Text != "" && textBox3.Text != "")

{

if ((textBox4.Text == "" && checkBox3.Checked) || (textBox5.Text == "" && checkBox4.Checked) || (textBox6.Text == "" && checkBox5.Checked))

MessageBox.Show("Нельзя ставить ответ на пустые вопросы");

else

{

if ((textBox4.Text == "" && textBox5.Text != "" && textBox6.Text != "") || (textBox5.Text == "" && textBox6.Text != ""))

MessageBox.Show("Нельзся, что бы ответы шли не по порядку");

else

{

int x = 0;

while (!int.TryParse(textBox8.Text, out x) && x >= 1 && x <= 20)

MessageBox.Show("Введите число от 1 до 20");

string queryq = $"INSERT INTO {textBox7.Text} (Вопрос, Ответ1, Ответ2, Ответ3, Ответ4, Ответ5, Правильный1, Правильный2, Правильный3, Правильный4, Правильный5, Рейтинг) VALUES (N'{textBox1.Text}', N'{textBox2.Text}', N'{textBox3.Text}', N'{textBox4.Text}', N'{textBox5.Text}', N'{textBox6.Text}', N'{Convert.ToInt32(checkBox1.Checked)}', N'{Convert.ToInt32(checkBox2.Checked)}', N'{Convert.ToInt32(checkBox3.Checked)}', N'{Convert.ToInt32(checkBox4.Checked)}', N'{Convert.ToInt32(checkBox5.Checked)}', N'{x}')";

SqlCommand commandq = new SqlCommand(queryq, sqlConnectionq);

commandq.ExecuteScalar();

}

}

}

else

MessageBox.Show("Строки Вопроса, 1 Ответа, 2 Ответа обязательны");

}

}

else

MessageBox.Show("Длина строки не больше 50");

}

//новая таблица

public void New()

{

string queryq = $"CREATE TABLE [dbo].[{textBox7.Text}]([Id] INT NOT NULL PRIMARY KEY IDENTITY, [Вопрос] NVARCHAR(50) NULL, [Ответ1] NVARCHAR(50) NULL, [Ответ2] NVARCHAR(50) NULL, [Ответ3] NVARCHAR(50) NULL, [Ответ4] NVARCHAR(50) NULL, [Ответ5] NVARCHAR(50) NULL, [Правильный1] NVARCHAR(50) NULL, [Правильный2] NVARCHAR(50) NULL, [Правильный3] NVARCHAR(50) NULL, [Правильный4] NVARCHAR(50) NULL, [Правильный5] NVARCHAR(50) NULL), [Рейтинг] NVARCHAR(50) NULL)";

string queryt = $"CREATE TABLE [dbo].[{textBox7.Text}]([Id] INT NOT NULL PRIMARY KEY IDENTITY, [Вопрос] NVARCHAR(50) NULL, [Ответ1] NVARCHAR(50) NULL, [Ответ2] NVARCHAR(50) NULL, [Ответ3] NVARCHAR(50) NULL, [Ответ4] NVARCHAR(50) NULL, [Ответ5] NVARCHAR(50) NULL, [Правильный1] NVARCHAR(50) NULL, [Правильный2] NVARCHAR(50) NULL, [Правильный3] NVARCHAR(50) NULL, [Правильный4] NVARCHAR(50) NULL, [Правильный5] NVARCHAR(50) NULL)";

SqlCommand commandq = new SqlCommand(queryq, sqlConnectionq);

SqlCommand commandt = new SqlCommand(queryt, sqlConnectiont);

commandq.ExecuteScalar();

commandt.ExecuteScalar();

}

public Form2()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

sqlConnectionq = new SqlConnection(ConfigurationManager.ConnectionStrings["Question"].ConnectionString);

sqlConnectiont.Open();

sqlConnectionq.Open();

DataTable data= sqlConnectiont.GetSchema("Tables");

foreach (DataRow row in data.Rows)

comboBox1.Items.Add(row["TABLE\_NAME"].ToString());

textBox7.Size = new Size(0, 0);

}

private void Form2\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

private void button2\_Click(object sender, EventArgs e)

{

if (button2.Text == "Автоматически")

{

numerator = 1;

int x = 0;

bool plus = true;

bool one = false;

bool step = false;

if (comboBox1.Text != "")

{

if (int.TryParse(textBox8.Text, out x))

{

if (!(x >= 1 && x <= 20))

MessageBox.Show("Введите число от 1 до 20");

else

{

First();

SqlDataReader dataReader = null;

for (int i = 0; i < 20; i++)

{

try

{

string queryq = $"SELECT Вопрос, Ответ1, Ответ2, Ответ3, Ответ4, Ответ5, Правильный1, Правильный2, Правильный3, Правильный4, Правильный5 FROM {textBox7.Text} WHERE Рейтинг = {x}";

SqlCommand commandq = new SqlCommand(queryq, sqlConnectionq);

dataReader = commandq.ExecuteReader();

while (dataReader.Read())

{

textBox1.Text = Convert.ToString(dataReader["Вопрос"]);

textBox2.Text = Convert.ToString(dataReader["Ответ1"]);

textBox3.Text = Convert.ToString(dataReader["Ответ2"]);

textBox4.Text = Convert.ToString(dataReader["Ответ3"]);

textBox5.Text = Convert.ToString(dataReader["Ответ4"]);

textBox6.Text = Convert.ToString(dataReader["Ответ5"]);

try

{

checkBox1.Checked = Convert.ToBoolean(Convert.ToInt32(Convert.ToString(dataReader["Правильный1"])));

radioButton1.Checked = Convert.ToBoolean(Convert.ToInt32(Convert.ToString(dataReader["Правильный1"])));

}

catch

{

checkBox1.Checked = false;

radioButton1.Checked = false;

}

try

{

checkBox2.Checked = Convert.ToBoolean(Convert.ToInt32(Convert.ToString(dataReader["Правильный1"])));

radioButton2.Checked = Convert.ToBoolean(Convert.ToInt32(Convert.ToString(dataReader["Правильный1"])));

}

catch

{

checkBox2.Checked = false;

radioButton2.Checked = false;

}

try

{

checkBox3.Checked = Convert.ToBoolean(Convert.ToInt32(Convert.ToString(dataReader["Правильный1"])));

radioButton3.Checked = Convert.ToBoolean(Convert.ToInt32(Convert.ToString(dataReader["Правильный1"])));

}

catch

{

checkBox3.Checked = false;

radioButton3.Checked = false;

}

try

{

checkBox4.Checked = Convert.ToBoolean(Convert.ToInt32(Convert.ToString(dataReader["Правильный1"])));

radioButton4.Checked = Convert.ToBoolean(Convert.ToInt32(Convert.ToString(dataReader["Правильный1"])));

}

catch

{

checkBox4.Checked = false;

radioButton4.Checked = false;

}

try

{

checkBox5.Checked = Convert.ToBoolean(Convert.ToInt32(Convert.ToString(dataReader["Правильный1"])));

radioButton5.Checked = Convert.ToBoolean(Convert.ToInt32(Convert.ToString(dataReader["Правильный1"])));

}

catch

{

checkBox5.Checked = false;

radioButton5.Checked = false;

}

label2.Text = numerator.ToString();

Add();

numerator++;

if (numerator == 11)

break;

}

if (numerator == 11)

break;

}

catch

{

}

finally

{

if (dataReader != null && !dataReader.IsClosed)

dataReader.Close();

}

// THIS\*

plus = !plus;

if (plus)

{

if (one)

x++;

else

x += i + 1;

}

else

{

if (one)

x--;

else

x -= i + 1;

}

if (step)

one = true;

if (one)

plus = !plus;

if (x == 1 || x == 20)

step = true;

//\*

if (numerator == 11)

break;

}

numerator = 2;

Prew();

Navigation();

}

}

else

MessageBox.Show("Введите число от 1 до 20");

}

else

MessageBox.Show("Выберите тему");

}

else

{

Form2 form2 = new Form2();

form2.Text = "Редактор тестов:";

First();

}

}

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

{

Application.Exit();

}

private void button5\_Click(object sender, EventArgs e)

{

Add();

Next();

Navigation();

}

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

{

if (textBox7.Size == new Size(0, 0))

{

label1.Text = "Напишите название темы";

button1.Text = "Вернуться к выбору темы";

textBox7.Text = "";

button2.Visible = false;

textBox7.Size = new Size(154, 21);

}

else

{

label1.Text = "Выберите тему";

button1.Text = "Создать новую тему";

button2.Visible = true;

textBox7.Size = new Size(0, 0);

}

}

private void button3\_Click(object sender, EventArgs e)

{

Form3 form3 = new Form3();

form3.admin= true;

form3.text = this.text;

form3.Show();

this.Hide();

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

textBox7.Text = comboBox1.Text;

}

private void button8\_Click(object sender, EventArgs e)

{

radioButton1.Size = new Size(17,17);

checkBox1.Size = new Size(0, 0);

}

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

{

Add();

Prew();

Navigation();

}

private void button8\_Click\_1(object sender, EventArgs e)

{

if (comboBox1.Text == "")

MessageBox.Show("Выберите тему");

else

{

Form2 form2 = new Form2();

form2.Text = "Редактор тестов: " + comboBox1.Text;

numerator = 1;

if (textBox7.Size == new Size(0, 0))

Navigation();

else

{

New();

Navigation();

}

First();

button6.Size = new Size(0, 0);

}

}

private void button9\_Click(object sender, EventArgs e)

{

if (checkBox1.Size == new Size(0, 0))

{

button9.Text = "Одиночный";

checkBox1.Size = new Size(17,17);

checkBox2.Size = new Size(17, 17);

checkBox3.Size = new Size(17, 17);

checkBox4.Size = new Size(17, 17);

checkBox5.Size = new Size(17, 17);

}

else

{

button9.Text = "Множественный";

checkBox1.Size = new Size(0, 0);

checkBox2.Size = new Size(0, 0);

checkBox3.Size = new Size(0, 0);

checkBox4.Size = new Size(0, 0);

checkBox5.Size = new Size(0, 0);

}

}

private void radioButton1\_Click(object sender, EventArgs e)

{

checkBox1.Checked = radioButton1.Checked;

checkBox2.Checked = false;

checkBox3.Checked = false;

checkBox4.Checked = false;

checkBox5.Checked = false;

}

private void radioButton2\_Click(object sender, EventArgs e)

{

checkBox1.Checked = false;

checkBox2.Checked = radioButton2.Checked;

checkBox3.Checked = false;

checkBox4.Checked = false;

checkBox5.Checked = false;

}

private void radioButton3\_Click(object sender, EventArgs e)

{

checkBox1.Checked = false;

checkBox2.Checked = false;

checkBox3.Checked = radioButton3.Checked;

checkBox4.Checked = false;

checkBox5.Checked = false;

}

private void radioButton4\_Click(object sender, EventArgs e)

{

checkBox1.Checked = false;

checkBox2.Checked = false;

checkBox3.Checked = false;

checkBox4.Checked = radioButton4.Checked;

checkBox5.Checked = false;

}

private void radioButton5\_Click(object sender, EventArgs e)

{

checkBox1.Checked = false;

checkBox2.Checked = false;

checkBox3.Checked = false;

checkBox4.Checked = false;

checkBox5.Checked = radioButton5.Checked;

}

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

{

Add();

if (Search() < 0)

Insert();

}

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

{

Help.ShowHelp(this, "C:\\Users\\37533\\Desktop\\C#\\Тестирование знаний\\Тестирование знаний\\Admin.chm");

}

}

}

**2 Окно(Форма): Редактор тестов**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

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

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

namespace Тестирование\_знаний

{

public partial class Form3 : Form

{

public bool admin = false;

public string text = "";

SqlConnection sqlConnectionq = new SqlConnection(ConfigurationManager.ConnectionStrings["Question"].ConnectionString);

SqlConnection connection = new SqlConnection(ConfigurationManager.ConnectionStrings["Autorisation"].ConnectionString);

public int numerator;

public int[][] ans = new int[10][] { new int[5], new int[5], new int[5], new int[5], new int[5], new int[5], new int[5], new int[5], new int[5], new int[5] };

public int[][] rig = new int[10][] { new int[5], new int[5], new int[5], new int[5], new int[5], new int[5], new int[5], new int[5], new int[5], new int[5] };

public int Search(string text)

{

string query = $"SELECT Id FROM Авторизация WHERE Пользователи = N'{text}'";

try

{

SqlCommand sqlCommand = new SqlCommand(query, connection);

return Convert.ToInt32(sqlCommand.ExecuteScalar().ToString());

}

catch

{

return -1;

}

}

public int Search()

{

string query = $"SELECT Id FROM {comboBox1.Text} WHERE Вопрос = N'{label3.Text}' AND Ответ1 = N'{label4.Text}' AND Ответ2 = N'{label5.Text}' AND Ответ3 = N'{label6.Text}' AND Ответ4 = N'{label7.Text}' AND Ответ5 = N'{label8.Text}'";

try

{

SqlCommand sqlCommand = new SqlCommand(query,sqlConnectionq);

return Convert.ToInt32(sqlCommand.ExecuteScalar().ToString());

}

catch

{

return -1;

}

}

public int Rating()

{

return Convert.ToInt32(new SqlCommand($"SELECT Рейтинг FROM {comboBox1.Text} WHERE Id = {Search()}", sqlConnectionq).ExecuteScalar().ToString());

}

public void Navigation()

{

Form2 form2 = new Form2();

form2.sqlConnectiont.Open();

label2.Text = numerator.ToString();

int numans = 0;

Label[] textbox = new Label[] {label3, label4, label5, label6, label7, label8 };

RadioButton[] radiobuttons = new RadioButton[] { radioButton1, radioButton2, radioButton3, radioButton4, radioButton5 };

CheckBox[] checkboxes = new CheckBox[] { checkBox1, checkBox2, checkBox3, checkBox4, checkBox5 };

string[] texts = new string[] { "Вопрос", "Ответ1", "Ответ2", "Ответ3", "Ответ4", "Ответ5" };

string[] answers = new string[] { "Правильный1", "Правильный2", "Правильный3", "Правильный4", "Правильный5" };

for (int i = 0; i < 6; i++)

{

string query = $"SELECT {texts[i]} FROM {comboBox1.Text} WHERE Id = {numerator}";

textbox[i].Text = new SqlCommand(query, form2.sqlConnectiont).ExecuteScalar().ToString();

}

for (int i = 0; i < 5; i++)

{

string query = $"SELECT {answers[i]} FROM {comboBox1.Text} WHERE Id = {numerator}";

try

{

rig[numerator - 1][i] = Convert.ToInt32(new SqlCommand(query, form2.sqlConnectiont).ExecuteScalar().ToString());

}

catch

{

rig[numerator - 1][i] = 0;

}

if (rig[numerator - 1][i] == 1)

numans++;

}

if (numans > 1)

{

for (int i = 0; i < 5; i++)

checkboxes[i].Size = new Size(17, 17);

}

else

{

for (int i = 0; i < 5; i++)

checkboxes[i].Size = new Size(0, 0);

}

}

public void Next()

{

numerator++;

if (numerator == 10)

button2.Size = new Size(0, 0);

button3.Size = new Size(64, 19); ;

}

public void Prew()

{

numerator--;

if (numerator == 1)

button3.Size = new Size(0, 0);

button2.Size = new Size(64, 19);

}

public void First()

{

label1.Visible = !label1.Visible;

label2.Visible = !label2.Visible;

label3.Visible = !label3.Visible;

label4.Visible = !label4.Visible;

label5.Visible = !label5.Visible;

label6.Visible = !label6.Visible;

label7.Visible = !label7.Visible;

label8.Visible = !label8.Visible;

checkBox1.Visible = !checkBox1.Visible;

checkBox2.Visible = !checkBox2.Visible;

checkBox3.Visible = !checkBox3.Visible;

checkBox4.Visible = !checkBox4.Visible;

checkBox5.Visible = !checkBox5.Visible;

comboBox1.Visible = !comboBox1.Visible;

radioButton1.Visible = !radioButton1.Visible;

radioButton2.Visible = !radioButton2.Visible;

radioButton3.Visible = !radioButton3.Visible;

radioButton4.Visible = !radioButton4.Visible;

radioButton5.Visible = !radioButton5.Visible;

button1.Visible = !button1.Visible;

button2.Visible = !button2.Visible;

button3.Visible = !button3.Visible;

button5.Visible = !button5.Visible;

button7.Visible = !button7.Visible;

}

public Form3()

{

InitializeComponent();

}

private void Form3\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

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

{

if (comboBox1.Text == "")

MessageBox.Show("Выберите тему");

else

{

First();

numerator = 2;

Prew();

Navigation();

}

}

private void Form3\_Load(object sender, EventArgs e)

{

Form2 form2 = new Form2();

form2.sqlConnectiont.Open();

if (admin)

button6.Show();

DataTable data = form2.sqlConnectiont.GetSchema("Tables");

foreach (DataRow row in data.Rows)

comboBox1.Items.Add(row["TABLE\_NAME"].ToString());

sqlConnectionq.Open();

connection.Open();

}

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

{

Form2 form2 = new Form2();

form2.Show();

this.Close();

}

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

{

Application.Exit();

}

private void button2\_Click(object sender, EventArgs e)

{

Next();

Navigation();

}

private void button3\_Click(object sender, EventArgs e)

{

Prew();

Navigation();

}

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

{

if (checkBox1.Size == new Size(0,0))

{

ans[numerator - 1][0] = Convert.ToInt32(radioButton1.Checked);

ans[numerator - 1][1] = Convert.ToInt32(radioButton2.Checked);

ans[numerator - 1][2] = Convert.ToInt32(radioButton3.Checked);

ans[numerator - 1][3] = Convert.ToInt32(radioButton4.Checked);

ans[numerator - 1][4] = Convert.ToInt32(radioButton5.Checked);

}

else

{

ans[numerator - 1][0] = Convert.ToInt32(checkBox1.Checked);

ans[numerator - 1][1] = Convert.ToInt32(checkBox2.Checked);

ans[numerator - 1][2] = Convert.ToInt32(checkBox3.Checked);

ans[numerator - 1][3] = Convert.ToInt32(checkBox4.Checked);

ans[numerator - 1][4] = Convert.ToInt32(checkBox5.Checked);

}

}

private void button5\_Click(object sender, EventArgs e)

{

Form1 form1 = new Form1();

string miss = "";

bool pass = false;

int mark = 0;

for (int i = 0; i < 10; i++)

{

if (Enumerable.SequenceEqual(ans[i], new int[5] { 0, 0, 0, 0, 0 }))

{

pass = true;

miss += (i + 1) + ", ";

}

}

if (pass)

MessageBox.Show("Вы не ответили на " + miss + "вопросы, не забывайте нажимать на ответ");

else

{

miss = "";

for (int i = 0; i < 10 ; i++)

{

if (Enumerable.SequenceEqual(ans[i], rig[i]))

{

numerator= i + 1;

Navigation();

if (Search() >=0 && Rating() > 1)

{

SqlCommand c = new SqlCommand($"UPDATE {comboBox1.Text} SET Рейтинг = '{Rating() -1}' Where Id = {Search()}",sqlConnectionq);

c.ExecuteScalar();

}

miss+= i + ", ";

mark++;

}

else

{

numerator = i + 1;

Navigation();

if (Search() >= 0 && Rating() < 20)

{

SqlCommand c1 = new SqlCommand($"UPDATE {comboBox1.Text} SET Рейтинг = '{Rating() + 1}' Where Id = {Search()}",sqlConnectionq);

c1.ExecuteScalar();

}

}

}

string qy = $"SELECT {comboBox1.Text} FROM Авторизация WHERE Id = {Search(text)}";

SqlCommand c2 = new SqlCommand(qy, connection);

if (c2.ExecuteScalar().ToString() == "")

{

c2 = new SqlCommand($"UPDATE Авторизация SET {comboBox1.Text} = N'{mark}' WHERE Id = {Search(text)}", connection);

c2.ExecuteScalar();

}

MessageBox.Show("Оценка " + mark + ". Правильно отвенено на " + miss + "вопросы","Оценка:" +mark);

}

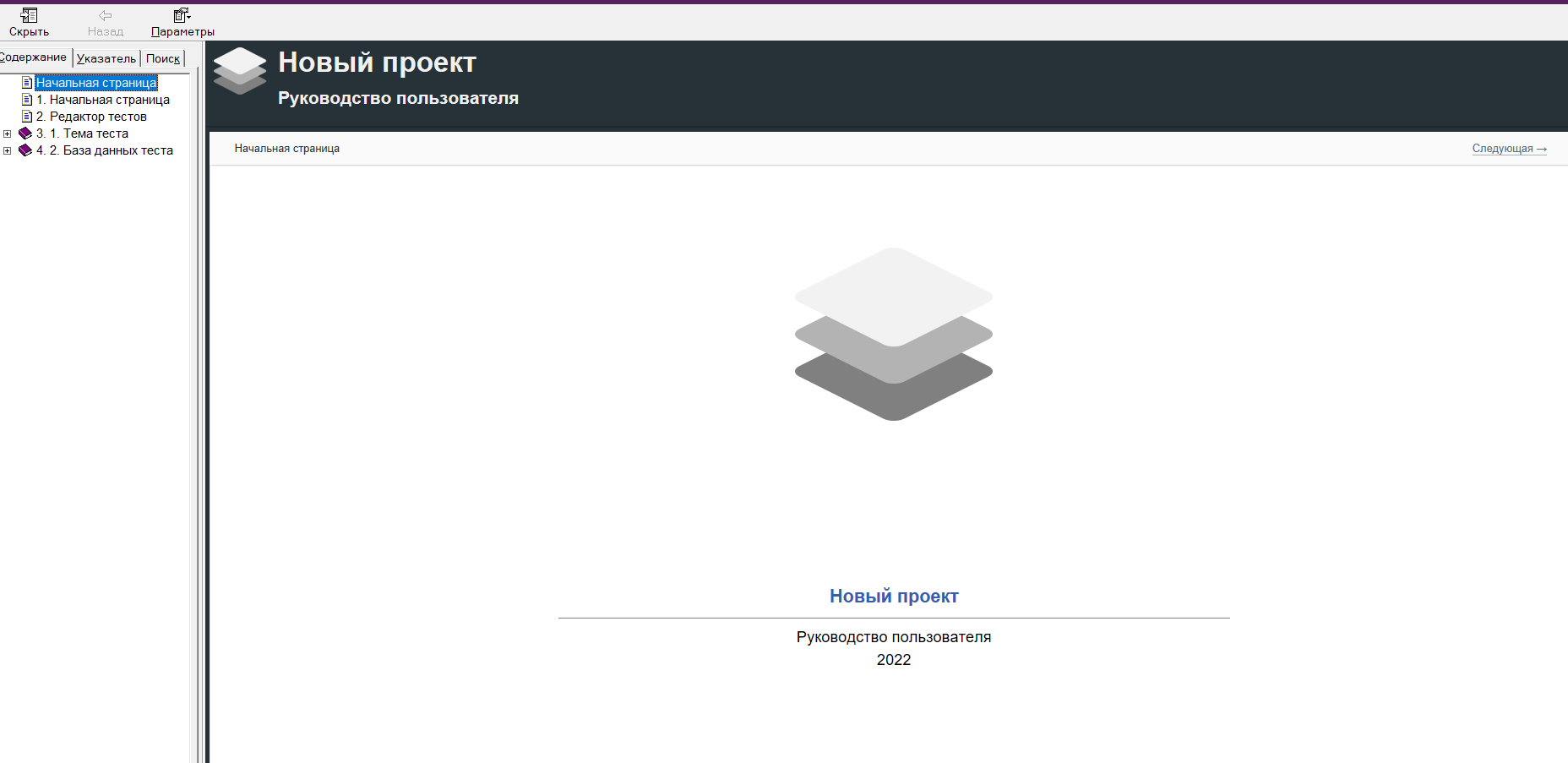
}

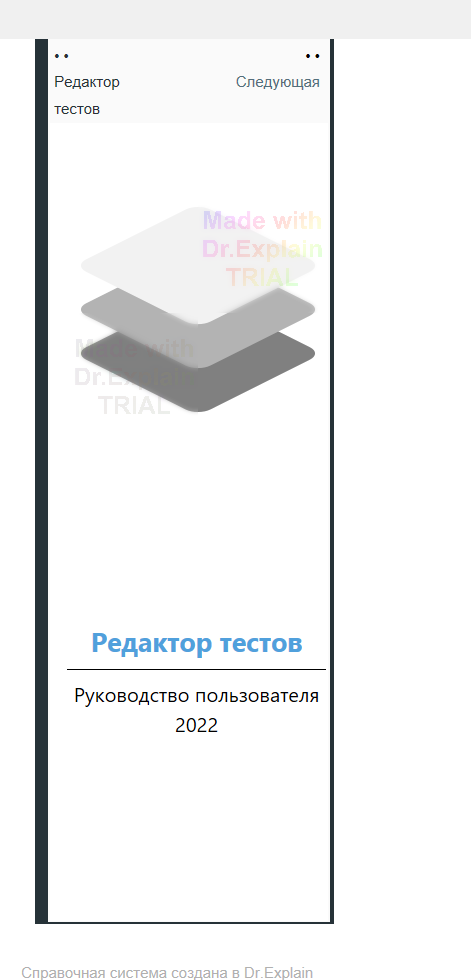
}

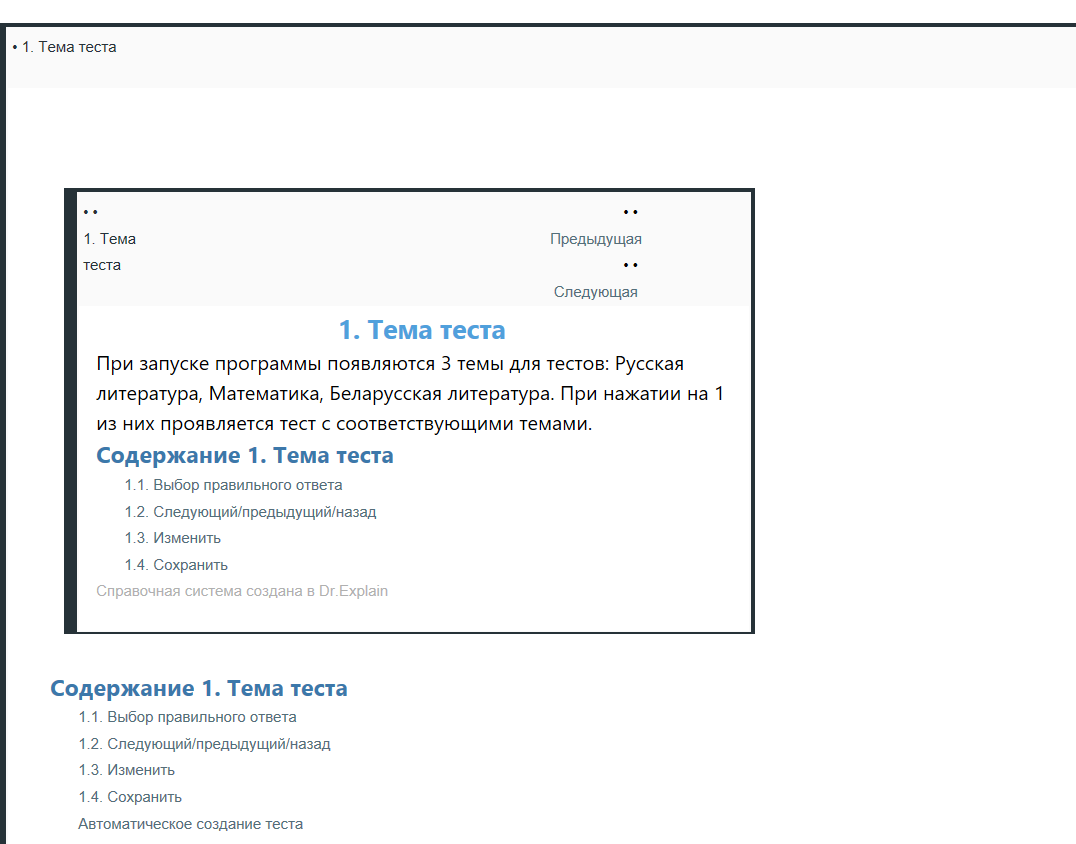
}

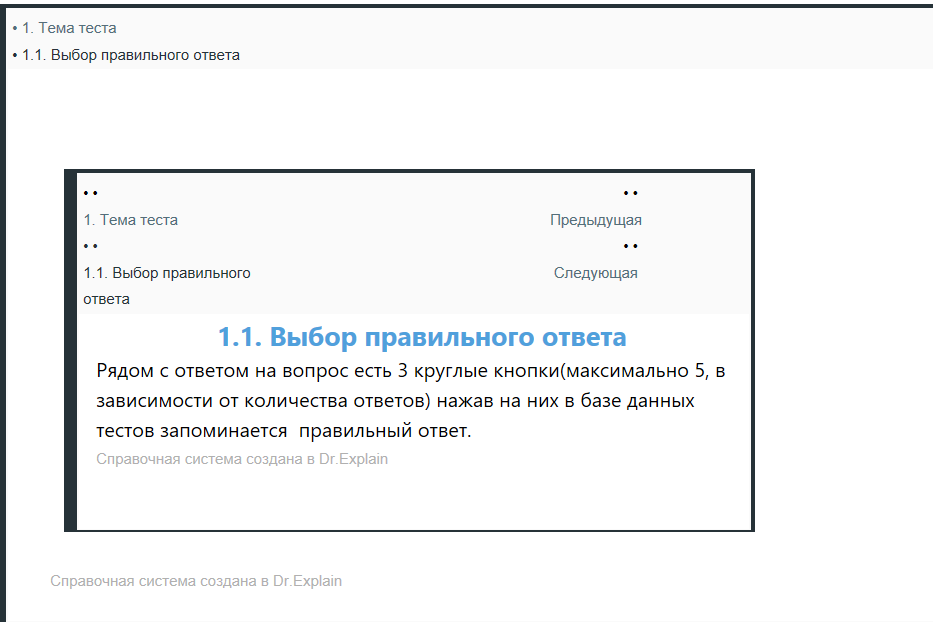
**Реализация**

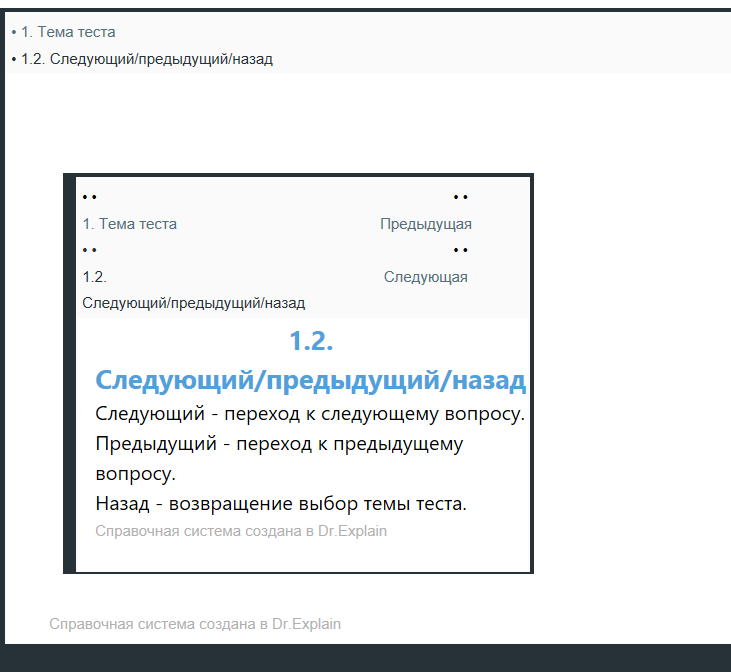
**Руководство Администратора**

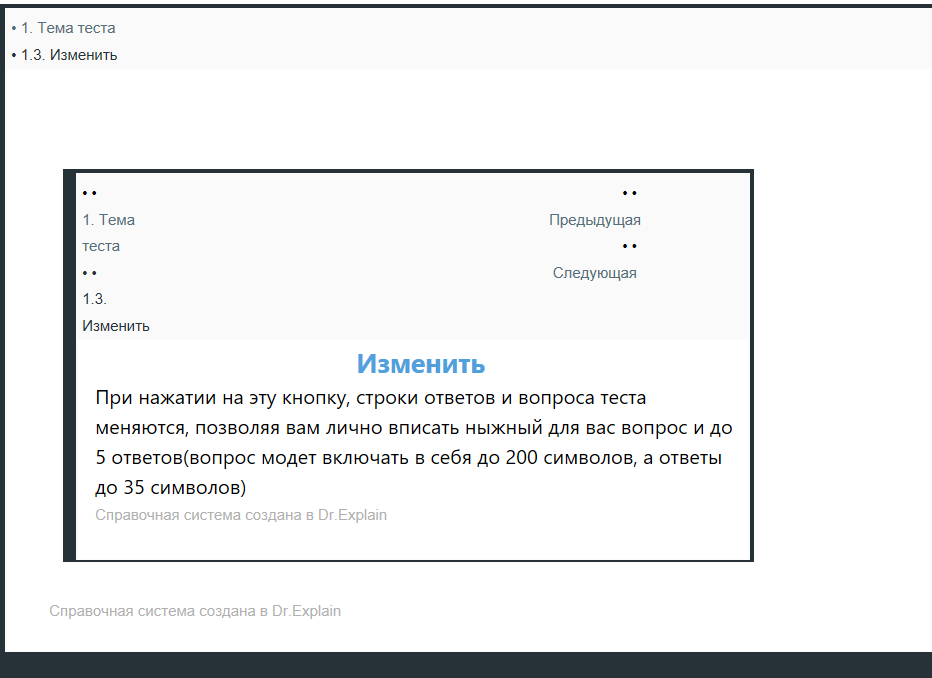


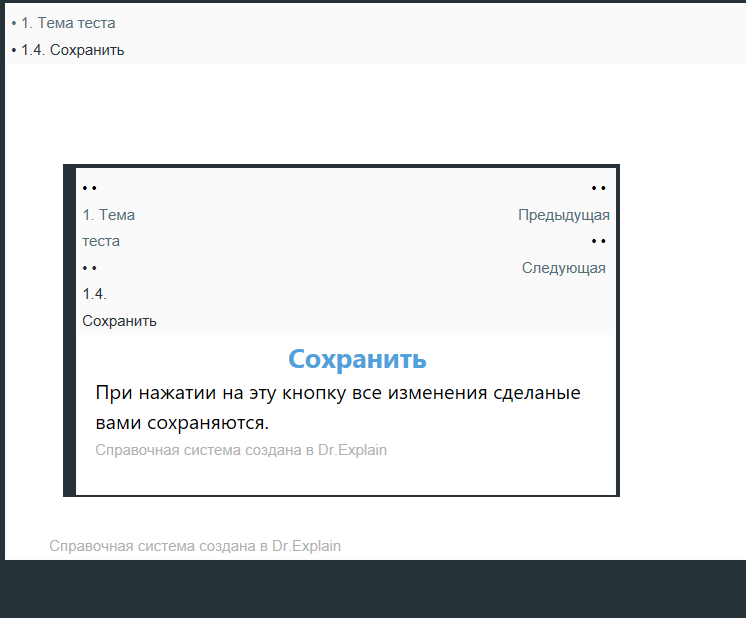


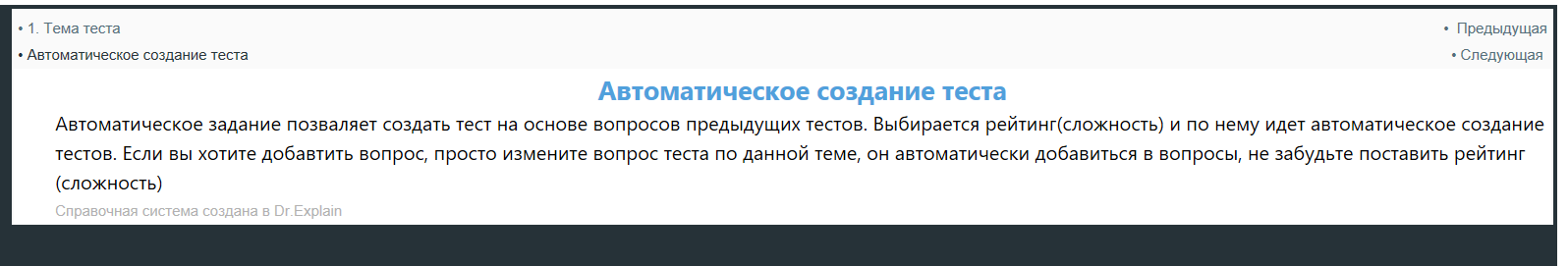


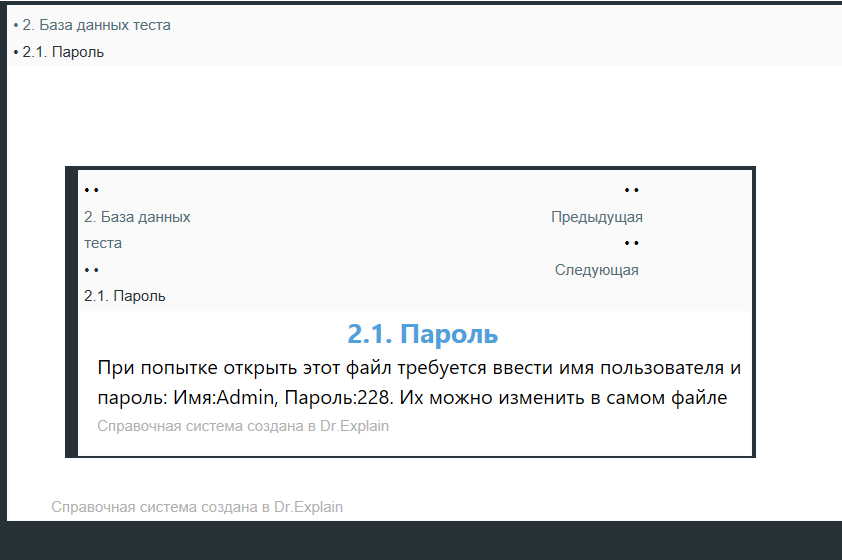


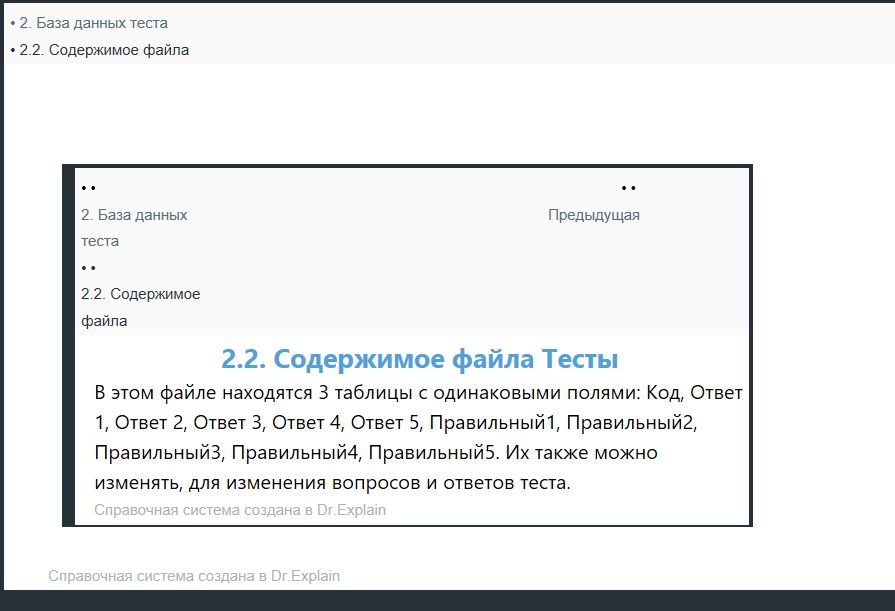


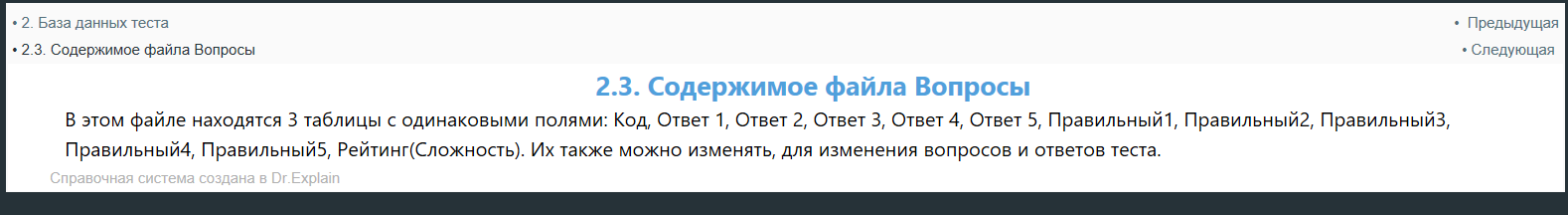


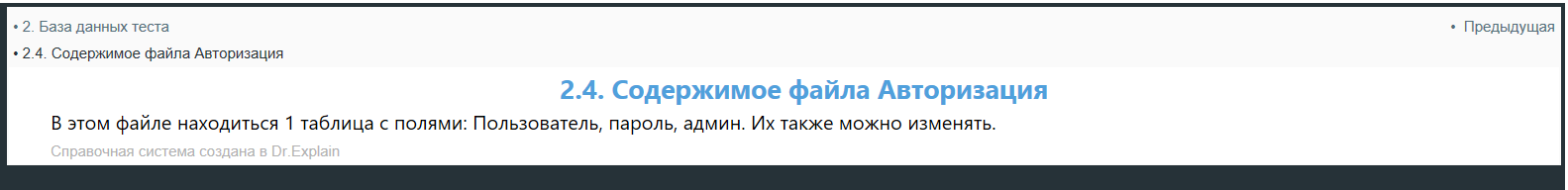












**Руководство пользователя**

Выбрать тему – выбираете тему в падающем меню, после чего откроется тест.

Назад\Вперед – кнопки навигации.

Ответ – выбрать ответ, по-иному оно не защитает его.

Выход – Завершение программы.

Вернуться – вернуться к выбору тестов.

**Тестирование**

**Тест кейс**

**Форма 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер | Действие | Ожидание | Результат | Примечание |
| 1 | Неправильно написать имя  И нажать «Вход» | Окно с сообщением | Окно с сообщением | - |
| 2 | Неправильно написать пароль  И нажать «Вход» | Окно с сообщением | Окно с сообщением | - |
| 3 | Не писать имя  И нажать «Вход» | Окно с сообщением | Окно с сообщением | - |
| 4 | Не писать пароль  И нажать «Вход» | Окно с сообщением | Окно с сообщением | - |
| 5 | Правильно ввести имя и пароль  И нажать «Вход» | Переход в другое окно | Переход в другое окно | - |
| 6 | Неправильно написать имя  И нажать «Регистрация» | Сохранение пользователя в БД | Сохранение пользователя в БД | - |
| 7 | Неправильно написать пароль  И нажать «Регистрация» | Сохранение пользователя в БД | Сохранение пользователя в БД | - |
| 8 | Не писать имя  И нажать «Регистрация» | Окно с сообщением | Окно с сообщением | - |
| 9 | Не писать пароль  И нажать «Регистрация» | Окно с сообщением | Окно с сообщением | - |

**Форма 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер | Действие | Ожидание | Результат | Примечание |
| 1 | Выбрать тест или создать новый тест | Появление теста | Появление теста | - |
| 2 | Нажать «Сохранить» | Добавиться сохранение теста | Добавиться сохранение теста | - |
| 3 | Нажать «Сохранить» без ответа | Окно с сообщением | Окно с сообщением | - |
| 4 | Нажать «Сохранить» оставляя ответ через пустой ответ | Окно с сообщением | Окно с сообщением | - |
| 5 | Нажать «Сохранить» оставляя правильный ответ на пустой ответ | Окно с сообщением | Окно с сообщением | - |
| 6 | Нажать «Сохранить» без вопроса | Окно с сообщением | Сохранение теста в БД | Исправить добавив условие |
| 7 | Нажать «Сохранить» без вопроса | Окно с сообщением | Окно с сообщением | - |
| 8 | Нажать «Сохранить» | Сохранение теста в БД | Сохранение теста в БД | - |
| 9 | Нажать «Вперед» | Переход на следующий вопрос | Переход на следующий вопрос | - |
| 10 | Нажать «Назад» | Переход на предыдущий вопрос | Переход на предыдущий вопрос | - |
| 11 | Нажать «Одиночный/Множественный ответ» | Изменение кол-ва ответов | Изменение кол-ва ответов | - |
| 12 | Нажать «Пройти тест» | Переход на 3 форму | Переход на 3 форму | - |
| 13 | Нажать «Автоматически», не вписав в рейтинг ничего | Окно с сообщением | Окно с сообщением | - |
| 14 | Нажать «Автоматически», не вписав в рейтинг ничего | Появление теста | Появление теста | - |
| 15 | Нажать «Выйти» | Завершение программы | Завершение программы | - |
| 16 | Нажать «Сохранить» не вписав в рейтинг ничего | Окно с сообщением | Окно с сообщением | - |

**Форма 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер | Действие | Ожидание | Результат | Примечание |
| 1 | Выбрать тест | Появление теста | Появление теста | - |
| 2 | Нажать «Сохранить» | Добавиться сохранение теста | Добавиться сохранение теста | - |
| 3 | Нажать «Сохранить» без ответа | Окно с сообщением | Окно с сообщением | - |
| 4 | Нажать «Вперед» | Переход на следующий вопрос | Переход на следующий вопрос | - |
| 5 | Нажать «Назад» | Переход на предыдущий вопрос | Переход на предыдущий вопрос | - |
| 6 | Нажать «Завершить тест» | Вывод оценки | Вывод оценки | - |
| 7 | Нажать «Завершить тест» не отвечая на ответы. | Окно с сообщением | Окно с сообщением | - |
| 8 | Нажать «Выйти» | Завершение программы | Завершение программы | - |

**Заключение**

В работе были успешно выполнены все поставленные задачи.

1. Создана инфологическая модель данных, показывающая взаимосвязи между сущностями данных.

2. Создана логическая модель данных, показывающая конкретные связи между таблицами.

3. Оптимальной средой разработки приложения является Visual Studio, оптимальный способ организации данных – СУБД MS SQL.

4. Создано приложение с интуитивно понятным интерфейсом, не требующее сложной установки.

При перемещении программы Теста с одного компьютера на другой также не возникает осложнений – приложение создано так, что единственным условием является целостность всех файлов программы, и более ничего. Программа – единственный скомпилированный файл. Внимательнее стоит быть с папкой, так как пропажа или порча одного файла из папки с программой, прекращает корректную работу приложения.

В приложении удобным интерфейсом расположены все необходимые функции.

Конечно, представленная программа не является идеалом автоматического тестирования, но нельзя сказать, что она и не удовлетворяет минимальному набору требований.

Программа написана с учетом офлайн проведения тестов, и рассчитана на сетевой доступ к базам данных.

Поставленная цель автоматизации тестирования достигнута, посредством поиска по данным, автоматизации создания тестов.