Департамент образования Курганской области

Государственное бюджетное профессиональное образовательное учреждение

«Курганский технологический колледж

имени Героя Советского Союза Н. Я. Анфиногенова»

Защищен с оценкой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Отделение Автоматизация и вычислительная техника

КУРСОВОЙ ПРОЕКТ

по МДК.01.01 «Разработка программных модулей»

на тему: «Разработка приложения для тестирования студентов по школьным предметам»

КТК.09.02.07.2063.14885.КП

Студент: Новиков Данил Вячеславович

Преподаватель: Афанасьев Алексей Васильевич

Курган, 2022

Задание

на курсовой проект

по МДК.01.01 «Разработка программных модулей»

на тему: «Разработка приложения для тестирования студентов по школьным предметам»

Пояснительная записка

Введение

1 Аналитическая часть

* характеристика программных средств для создания приложения дать
* функции и параметры программных средств, используемых для решения задач
* характеристика предметной области

2 Проектная часть

* структура базы данных, роли и пользователи
* структура информационной системы

3 Руководство пользователя

* стартовая страница
* роль преподавателя
* роль студента
* роль администратора

Заключение

Литература

Получил студент гр. 2063 \_\_\_\_\_\_\_\_\_\_\_ Д.В. Новиков

Преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Афанасьев

Дата \_\_\_\_\_\_\_\_\_\_2022

Содержание

[Введение 2](#_Toc107246216)

[1 Аналитическая часть 3](#_Toc107246217)

[1.1 Характеристика программных средств для создания приложения 5](#_Toc107246218)

[1.2 Функции и параметры программных средств, используемых для решения задач 5](#_Toc107246219)

[1.3 Характеристика предметной области 9](#_Toc107246220)

[2 Проектная часть 11](#_Toc107246221)

[2.1 Структура базы данных, роли и пользователи 11](#_Toc107246222)

[2.2 Структура информационной системы 14](#_Toc107246223)

[3 Руководство пользователя 16](#_Toc107246224)

[3.1 Стартовая страница 16](#_Toc107246225)

[3.2 Роль преподавателя 18](#_Toc107246226)

[3.3 Роль студента 10](#_Toc107246227)

[3.4 Роль администратора 20](#_Toc107246228)

[Заключение 23](#_Toc107246229)

[Литература 24](#_Toc107246230)

Введение

Суть проекта состоит в создании приложения, в котором студенты первых курсов имеют возможность проходить тестирования по школьным предметам. Учитель имеет возможность добавить тестирование по определенному предмету. Администратор может изменять любую информацию, имеющуюся в базе данных.

Данное приложение автоматизирует систему тестирования. Преподавателю нет необходимости распечатывать тест, затем проверять его. Всю информацию преподаватель может внести в приложение, после прохождения теста, приложение выведет верные ответы и общее их количество.

Цель курсового проекта: «Разработка приложения для тестирования студентов по школьным предметам»

Для удовлетворения поставленной цели, необходимо выполнить ряд задач:

* определить роли и пользователей в приложение;
* создать базу данных для хранения информации;
* построить макет приложения;
* создать логику программы, для выполнения функций.

1 Аналитическая часть

1.1 Характеристика программных средств для создания приложения

Для реализации цели были использованы следующие программы:

- Microsoft Visual Studio - это панель запуска для написания, отладки и создания кода для приложений, а затем их публикации. В дополнение к стандартному редактору и отладчику, которые есть в большинстве IDE, Visual Studio включает компиляторы, надстройки кода, графические дизайнеры и многие другие функции, расширяющие возможности разработки.

Visual Studio включает редактор исходного кода, который поддерживает технологию IntelliSense и позволяет легко перерабатывать код. Встроенный отладчик может действовать как отладчик уровня исходного кода или уровня машины. Другие встраиваемые инструменты включают редактор форм, упрощающий создание графического пользовательского интерфейса приложения, веб-редактор, конструктор классов и конструктор схемы базы данных.

Visual Studio позволяет создавать и подключать сторонние надстройки (расширения) для расширения функциональности практически на каждом уровне, включая поддержку систем контроля версий исходного кода (таких как Subversion и Visual SourceSafe), добавление новых наборов инструментов (таких как domain- определенные языки программирования), редактирование кода и визуальный дизайн) или другие аспекты процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server)

- Material Design In XAML - дополнение (плагин), разработанное компанией Microsoft, для удобного создания графического интерфейса в приложение WPF

- Database Designer - онлайн-инструмент проектирования и моделирования баз данных, графическое представление таблиц и связей. Помимо создания SQL-скрипта для всех основных движков баз данных, также можно экспортировать свою базу данных в формате image (PNG) или pdf. SQL скрипт создается автоматически после визуального построения модели и поддерживается следющие движки баз данных: MySQL, PostgreSQL, MS SQL, Oracle и SQLite.

- Microsoft SQL Server - система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, созданный совместно Microsoft и Sybase. Transact-SQL — это реализация стандартного языка структурированных запросов ANSI/ISO с расширениями. Он используется для работы с базами данных, начиная от персональных и заканчивая крупномасштабными базами данных

- Microsoft Excel - программа для работы с электронными таблицами, созданная корпорацией Microsoft для Microsoft Windows, Windows NT и Mac OS, а также Android, iOS и Windows Phone. Она предоставляет возможности экономико-статистических расчетов, графические инструменты и, за исключением Excel 2008 под Mac OS X, язык макропрограммирования VBA (Visual Basic for Application). Microsoft Excel входит в состав Microsoft Office.

1.2 Функции и параметры программных средств, используемых для решения задач

Для реализации кода в Microsoft Visual Studio был использован язык программирования С#. Для реализации дизайна, язык разметки XAML.

C# — это объектно - и компонентно-ориентированный язык программирования. C# предоставляет языковые конструкции прямая поддержка такой концепции работы. Это делает C# подходящим для создания и использования программных компонентов.

С момента своего появления C# был обогащен функциями, поддерживающими новые рабочие нагрузки и современные передовые методы разработки программного обеспечения. Программы C# выполняются в .NET, виртуальной системе выполнения, которая вызывает CLR и библиотеки классов. Common Language Runtime (CLR) — это международная стандартная реализация Microsoft Common Language Infrastructure (CLI). CLI является основой для создания сред выполнения и разработки, в которых языки и библиотеки прозрачно взаимодействуют друг с другом.

Исходный код, написанный на C#, переводится на промежуточный язык (IL), соответствующий спецификациям CLI. Код и ресурсы IL, включая растровые изображения и строки, хранятся в одной сборке, обычно с расширением .dll. Компиляция содержит манифест, содержащий информацию о типах, версии и культуре компиляции.

Когда программа C# выполняется, сборка загружается в среду CLR. CLR JIT преобразует код IL в инструкции машинного языка. Среда CLR также выполняет другие операции, такие как автоматическая сборка мусора, обработка исключений и управление ресурсами. Код, выполняемый в общеязыковой среде выполнения, иногда называют «управляемым кодом», чтобы отличить этот подход от «неуправляемого кода» который немедленно переводится на машинный язык для данной платформы.

Обеспечение взаимодействия между языками — ключевая особенность .NET. IL-код, сгенерированный компилятором C#, соответствует стандарту Common Type Specification (CTS). Код IL, сгенерированный из кода C#, может работать с кодом, сгенерированным из языковых версий .NET F#, Visual Basic, C++ и более чем 20 CTS-совместимых языков. Сборка может содержать несколько модулей, написанных на разных языках. NET, и все типы могут ссылаться друг на друга, как если бы они были написаны на одном языке.

В дополнение к службам среды выполнения .NET включает расширенные библиотеки. Эти библиотеки поддерживают широкий спектр рабочих нагрузок. Они организованы в пространства имен, которые предоставляют полезные опции от операций ввода и вывода файлов до манипуляций со строками и анализа XML, от фреймворков веб-приложений до элементов управления Windows Forms. Приложения C# обычно широко используют библиотеку классов .NET для решения общих задач.

Работа над C# началась в декабре 1998 года. Проект получил кодовое название COOL (объектно-ориентированный язык в стиле C).

Первая бета-версия C# 1.0 была выпущена летом 2000 года, а в феврале 2002 года это была финальная версия языка в дополнение к Microsoft Visual Studio. Поскольку C# сочетает в себе лучшие черты популярных языков программирования, которые предшествовали ему в C, Java и C++, программисты могут легко перейти на C# на основе своих знаний любого из этих языков.

Основным отличием от предшественников была возможность прописывать ингредиенты. C # представил такие функции, как события, методы и свойства. Финальный релиз C# 2.0 вышел в 2005 году. Это укрепило ее позиции на рынке.

Новые функции, такие как анонимные методы, частичные и параметризованные типы, значительно расширили возможности использования C#.

Во второй версии добавлены 64-битные вычисления, что открыло возможность увеличения адресного пространства. Триггеры, хранимые процедуры и типы данных также были созданы на языках .NET.

Версия 3.0, выпущенная в 2008 году, позволила C# занять лидирующие позиции в «гонке вооружений» языков. Новым в C# 3.0 является интегрированный язык запросов (LINQ); неявно введенные переменные и методы расширения; лямбда-выражения, которые назначают реализацию кода делегатам с использованием нового, более простого синтаксиса.

В частности, лямбда-выражения можно «отличить» вместе с языком интегрированных запросов.

Переменные анонимного типа позволили избежать неудобств и прямого неудобства описания переменных, позволяя объявлять новый тип непосредственно при создании. Также новым в C# 3.0 являются так называемые «ленивые вычисления», которые выполняют необходимые вычисления только в том случае, если запрашиваются соответствующие релевантные данные.

C# 4.0 был выпущен в 2010 году. Основное дополнение к предыдущим версиям с именем и необязательными аргументами. Первый позволяет связать аргумент и параметр по имени, а второй позволяет указать аргумент по умолчанию для каждого параметра. Не менее важным нововведением является динамический тип. Позволяет проверять соответствие типов объектов не на этапе компиляции, а непосредственно во время выполнения программы.

В то же время в .NET Framework 4.0 были внедрены нововведения — параллельная версия библиотеки распараллеливания задач (TPL) и интегрированный язык запросов (PLNQ). Их поддержка позволяет выполнять код параллельно на компьютерах с многоядерными или несколькими одноядерными процессорами.

XAML - основанный на XML язык разметки для декларативного программирования приложений, разработанный Microsoft. XAML широко используется в .NET Framework 3.0, особенно в Windows Presentation Foundation (WPF), Xamarin.Forms, Windows Workflow Foundation (WWF) и Silverlight. В WPF XAML — это язык разметки пользовательского интерфейса для определения элементов пользовательского интерфейса, привязки данных, поддержки событий и других свойств. В WWF XAML можно использовать для определения рабочих процессов.

Файлы XAML можно создавать и редактировать с помощью инструментов визуального дизайна, таких как Microsoft Expression Blend, Microsoft Visual Studio, WPF Visual Designer. Их также можно создавать с помощью стандартного текстового редактора, редактора кода, такого как XAMLPad, или графического редактора, такого как Vectropy.

Все, что создано или реализовано на XAML, может быть выражено на традиционных языках .NET, таких как C# или Visual Basic.NET. Однако ключевым аспектом технологии является снижение сложности инструментов, используемых для обработки XAML, поскольку XAML основан на XML. В результате существует множество продуктов, которые создают приложения на основе XAML. Поскольку XAML основан на XML, разработчики и дизайнеры могут работать с содержимым одновременно без необходимости перевода.

WPF (Windows Presentation Foundation) - это .NET платформа для создания клиентских приложений для настольных систем. Данный видео курс предназначен для тех, кто владеет языком C# на базовом уровне и хочет развиваться в сфере разработки настольного программного обеспечения

1.3 Характеристика предметной области

Каждый студент, зарегистрированный в приложение, характеризуется следующими параметрами:

* номер студенческого билета
* фамилия, имя отчество
* пол
* группа
* пароль

Каждый преподаватель, зарегистрированный в приложение, характеризуется следующими параметрами.

* табельный номер
* фамилия, имя, отчество
* пол
* пароль
* должность

Каждая созданная тема по предмету, характеризуется по следующим параметрам:

* название темы
* предмет
* вопрос
* ответ

Если не брать во внимание таблицы с информацией о студентах и преподавателях, то все таблицы связанны между собой (рис. 1.1). Тщательно обдумав все нюансы и образ работы приложения, можно приступить к созданию базы данных для хранения, ввода и вывода информации в приложение.

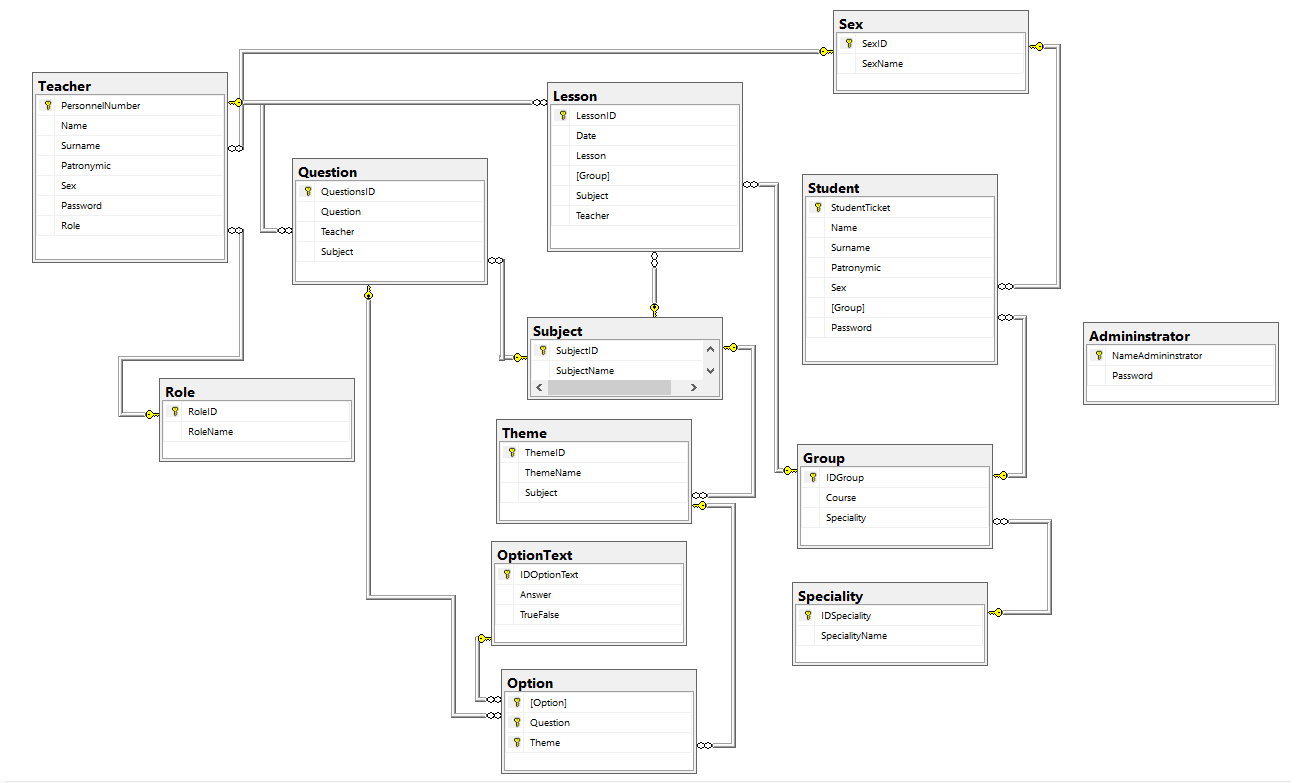


Рисунок 1.1 – Структура базы данных

В реляционных базах данных схема определяет таблицы, поля в каждой таблице (обычно с указанием их названия, типа, обязательности), и ограничения целостности (первичный, потенциальные и внешние ключи, другие ограничения).

После создания визуальной структуры базы данных, было принято решения перейти к следующему этапу, создание базы данных путем написания скрипта.

2 Проектная часть

2.1 Структура базы данных, роли и пользователи

Таблица «Студент» содержит список всех пользователей, которым открыт доступ для прохождения тестирования в приложение (рис. 2.1)

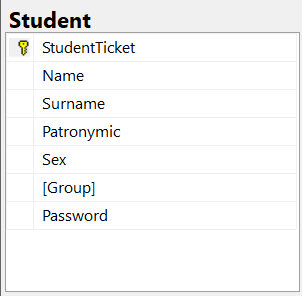


Рисунок 2.1.1 – Таблица для хранения данных студента

Таблица «Пол» и «Группа» являются внешним ключом.

Таблица «Преподаватель» содержит список всех преподавателей, которым открыт доступ для создания теста по определённому предмету (Рисунок 2.2).

Аналогично таблице «Студент», в таблице «Преподаватель» внешними ключами является поле «Пол» и поле «Должность».

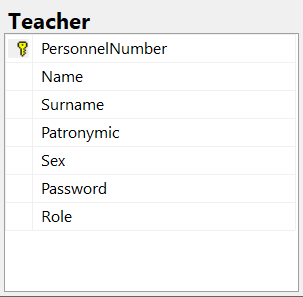


Рисунок 2.1.2 – Таблица для хранения данных преподавателя

Таблица «Вопрос» хранит в себе информацию о вопросе, предмете и учителе, который добавил вопрос.

Таблица «Тема», хранит в себе информацию о теме вопроса, и предмет, по которому она создана.

Таблица «Предмет», хранит в себе информацию о предмете.

Таблица «Вопрос текст» хранит в себе все варианты ответа.

Таблица «Вопрос» является связывающей таблицей для таблиц «Вопрос», «Ответ» и «Тема» (рис. 2.3)

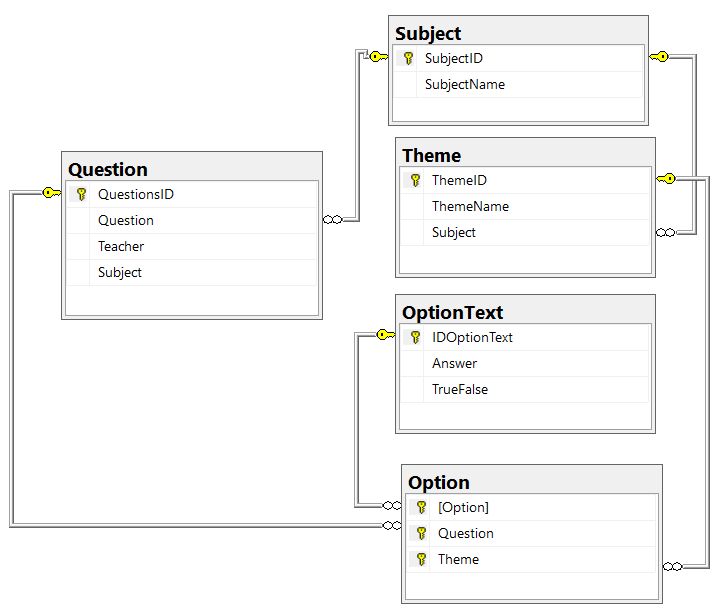


Рисунок 2.1.3 – Таблица для хранение данных о вопросе, темы и ответах

После авторизации преподаватель имеет возможность для добавления теста, студент имеет возможность прохождения теста, администратор имеет возможность редактирования данных, добавления студентов, преподавателей

2.2 Структура информационной системы

Структура информационных систем (рис. 2.2.1) может быть выбрана как совокупность прикладных систем независимо от области применения. В данном случае речь идет о структурном признаке исключения, и они присваиваются подсистемами. Таким образом, структуру любой информационной системы можно представить набором продуктивных подсистем.



Рисунок 2.2.1 – Структура информационных систем.

Проект находится в одной из этих подсистем, а именно Программное обеспечении.

Программное обеспечение включает общесистемные и специальные программные продукты, а также техническую документацию.

Программное обеспечение системного уровня включает ориентированные на пользователя программные системы, которые выполняют типичные задачи обработки информации. Они используются для расширения функциональных возможностей компьютеров, а также для контроля и управления процессами обработки данных.

Специальное программное обеспечение – это набор программ, разработанных при создании той или иной информационной системы. В него входят пакеты прикладных программ (ПП), реализующие модели с разной степенью соответствия, отражающие работу реального объекта.

К специальному ПО относятся такие программы, как 1С-Бухгалтерия, 3D-max, Консультант+ и другие. Обычно специальное программное обеспечение предназначено для поддержки принятия решения должностным лицом. А именно выбор оптимального (методического) режима решения из разработанной базы данных, на основе разработанного и апробированного алгоритма.

Системное программное обеспечение — это набор программ, которые рассматривают компоненты компьютера, такие как процессор, оперативная память, устройства ввода/вывода, сетевое оборудование, как «прослойку», аппаратное обеспечение с одной стороны и аппаратное обеспечение с другой. другие пользовательские приложения. В отличие от прикладного ПО, системное ПО не решает конкретных практических задач, а лишь обеспечивает работу других программ, наделяя их служебными функциями, абстрагирующими детали аппаратно-программной реализации вычислительной системы, и управляет аппаратными ресурсами вычислительной системы. система.

3 Руководство пользователя

3.1 Стартовая страница

В приложение существуют роль преподавателя, студента и администратора.

После запуска приложение открывается окно авторизации, пользователю необходимо авторизоваться (рис. 3.1.1) Учетную запись создает только администратор, варианта зарегистрироваться самостоятельно нет. При первом запуске приложения, учетной записи администратора нет, это предусмотрено, поэтому пользователь увидит диалоговое окно, которое предлагает создать администратора (рис. 3.1.2)

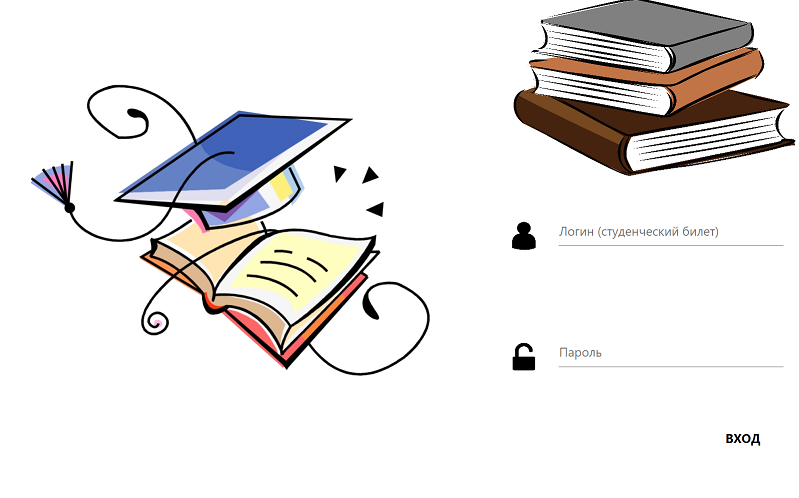


Рисунок 3.1.1 – Стартовая страница в приложение

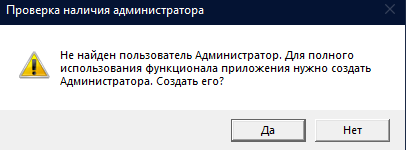


Рисунок 3.1.2 – Проверка наличия администратора

После появления диалогового окна (рис. 3.2), пользователю необходимо нажать на кнопку «Да», тогда он перейдет на регистрацию администратора (рис 3.3). Роль администратора в приложение только одна.

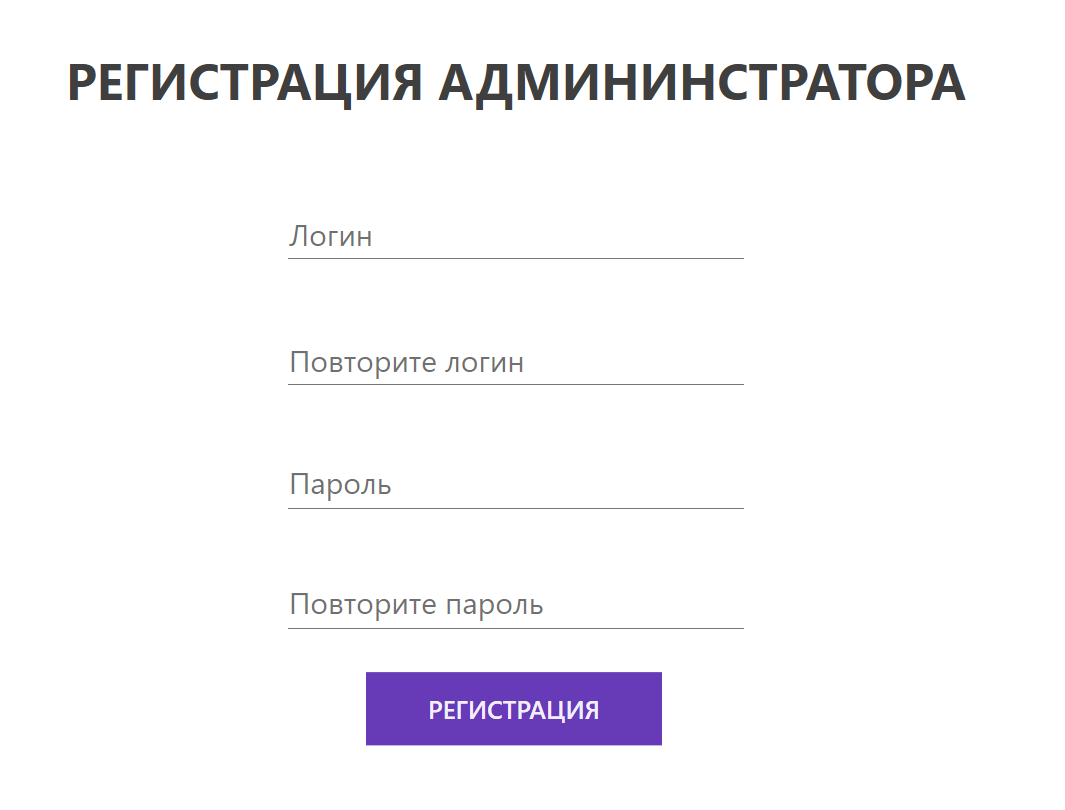


Рисунок 3.1.3 – Страница регистрации администратора

3.2 Роль преподавателя

После авторизации, у преподавателя есть возможность добавления тестирования для студентов. Для этого ему необходимо выбрать предмет, который добавляет администратор, выбрать или создать тему для вопросов, заполнить необходимые поля и выбрать правильный ответ. У вопроса всегда три ответа и один из них может быть верным (рис. 3.2.1).

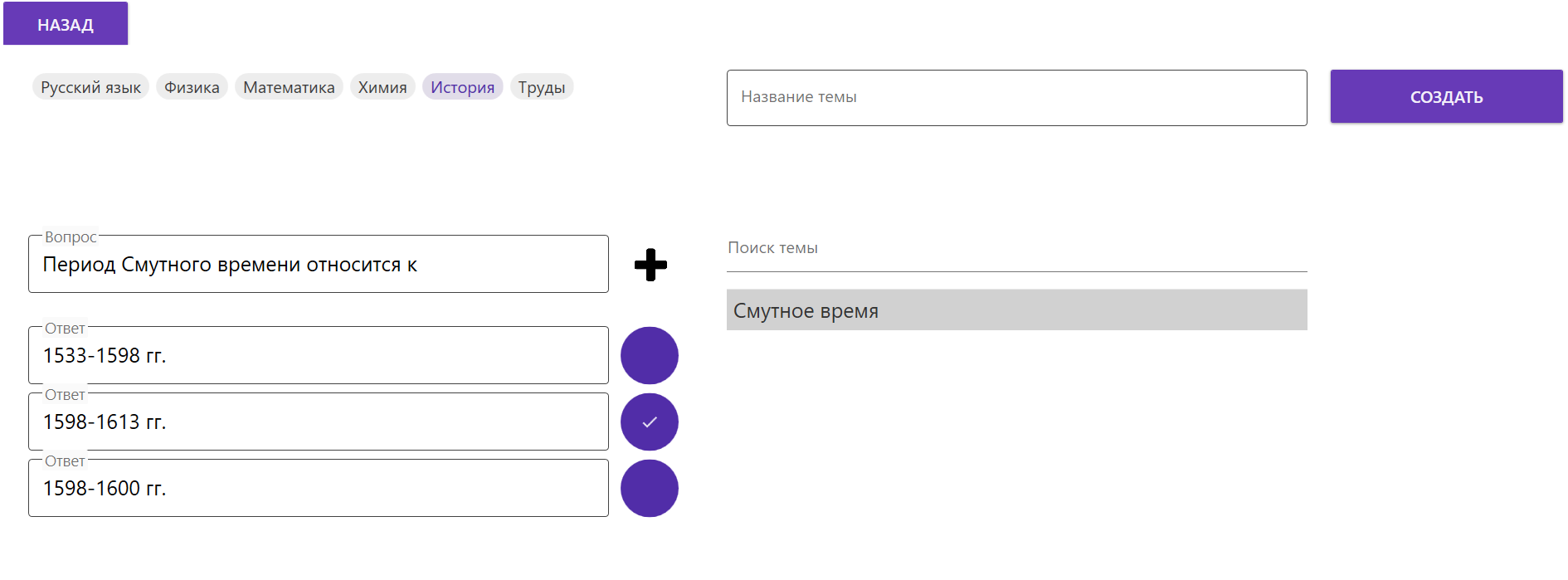


Рисунок 3.2.1 – Страница добавление темы и вопроса

После ввода всех данных, пользователю необходимо нажать на кнопку «+» для добавления вопроса в тему (рис. 3.2.1). Количество вопросов в теме не ограничено, пользователь может добавить любое количество. Для удобства поиска тем, в приложение реализован поиск, если у предмета большое количество тем. Пользователю необходимо вписать название темы в строку «Поиск тем».

3.3 Роль студента

После авторизации студента, появляется возможность прохождения теста. Пользователю необходимо выбрать предмет, затем выбрать тестирование, которое создал преподаватель ранее (рис. 3.3.1).

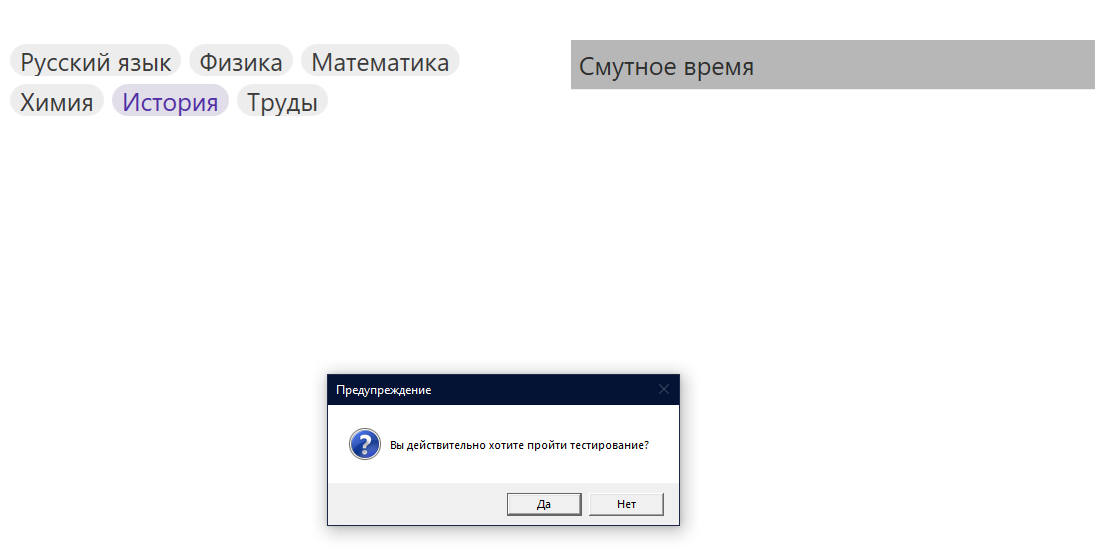


Рисунок 3.3.1 – Страница выбора тестирования

После подтверждения перехода на страницу с тестированием, пользователь видит вопрос и три ответа, ему необходимо выбрать один ответ и нажать кнопку далее, появится следующий вопрос (рис 3.3.2).

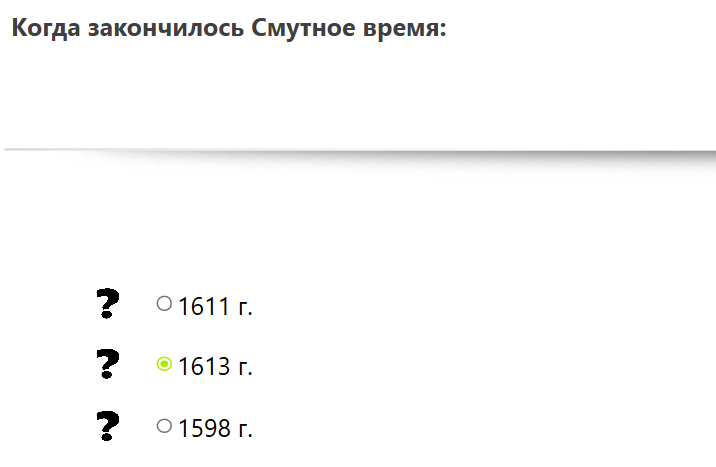


Рисунок 3.3.2 – Страница прохождения тестирования

Если вопросы в тестирование закончились, пользователь видит свой результат (рис. 3.3.3)

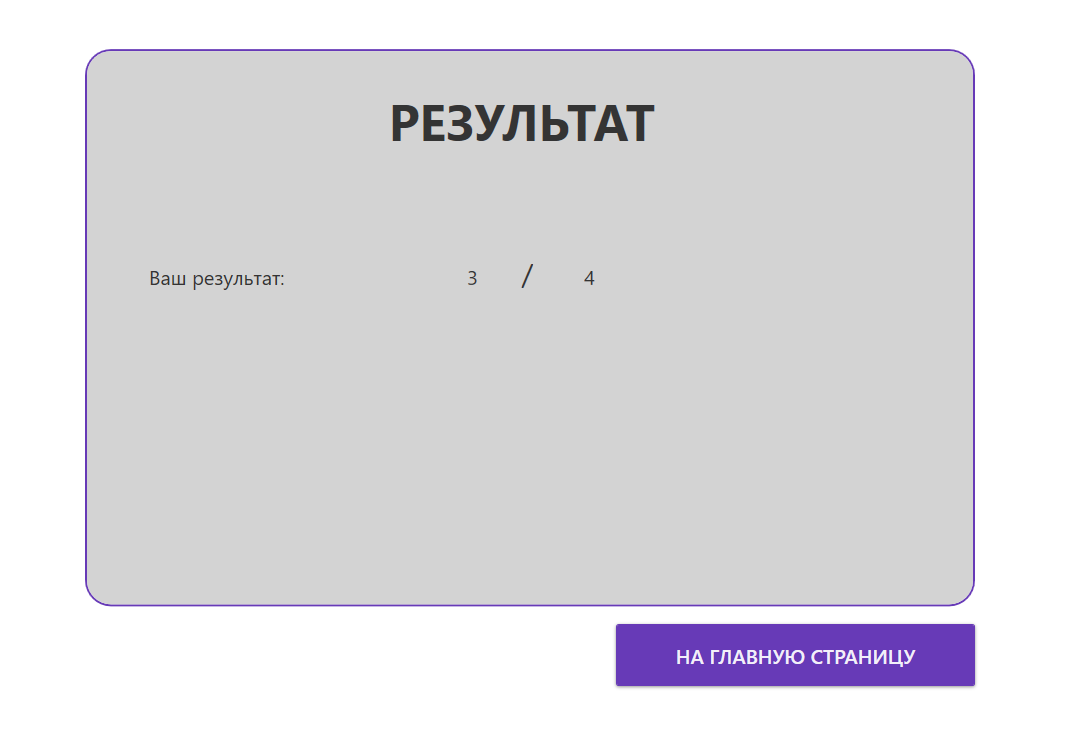


Рисунок 3.3.3 – Страница результата тестирования

3.4 Роль администратора

Администратор играет не мало важную роль в данном приложение. После регистрации администратора, ему необходимо ввести данные:

* студента
* преподавателя
* групп
* предметов
* ролей

Для добавления данных о студентах или преподавателях, администратору необходимо нажать на вкладку «Регистрация студента» или «Регистрация преподавателя» (рис.3.4.1)

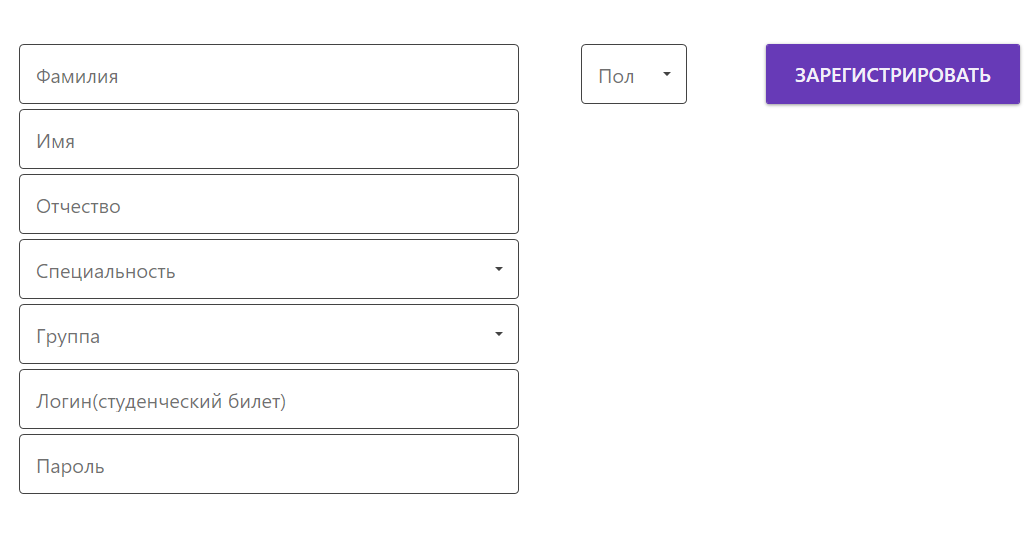


Рисунок 3.4.1 – Страница регистрации студента

После нажатия на вкладку, администратору необходимо заполнить все поля для сохранения данных и нажать на кнопку «Зарегистрировать», после этого информация о пользователе сохранится в базе данных (рис. 3.4.1). Добавления данных о студентах, идентично добавлению данных о преподавателях.

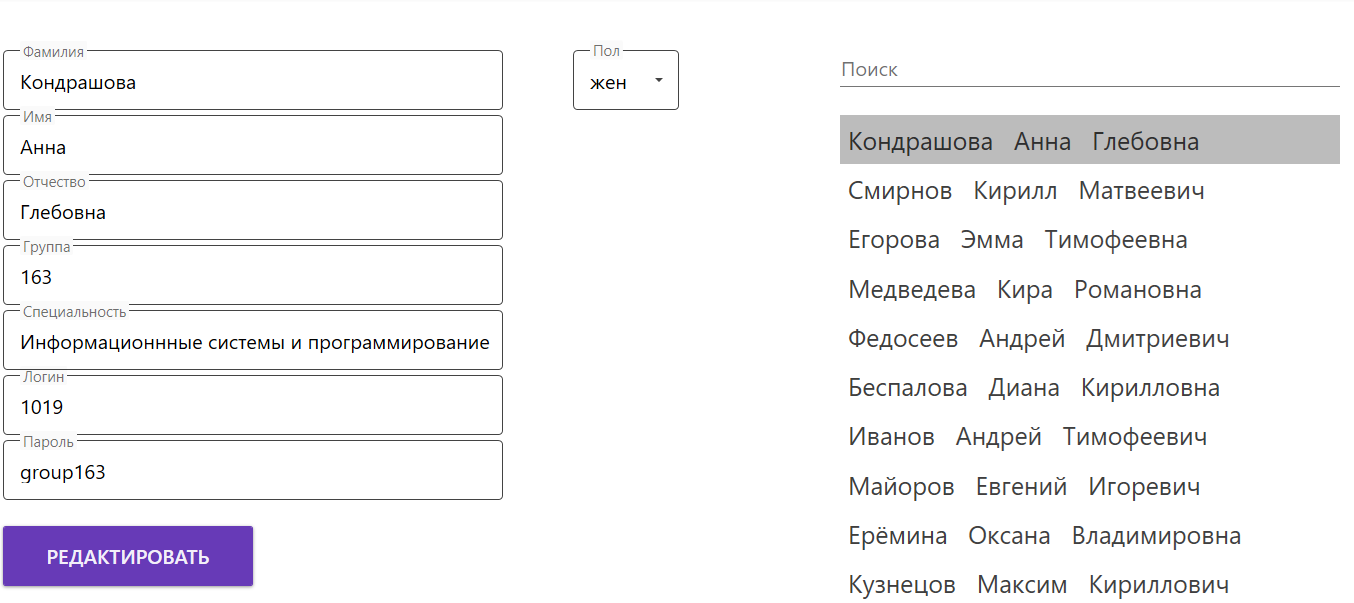


Рисунок 3.4.2 – Страница редактирования данных студента

На странице редактирования данных, администратор может изменить информацию о студенте. Для изменения информацию, администратору необходимо выбрать нужного студента из списка, затем в поле изменить информацию и нажать на кнопку «Редактировать». Логин изменить нет возможности, он является первичным ключом (рис. 3.4.2). Идентичная страница редактирования данных о преподавателях.

На вкладке «Регистрация студента» можно создать специальность (рис. 3.4.3). На вкладке «Регистрация преподавателя» можно создать должность преподавателя и предмет (рис. 3.4.4)

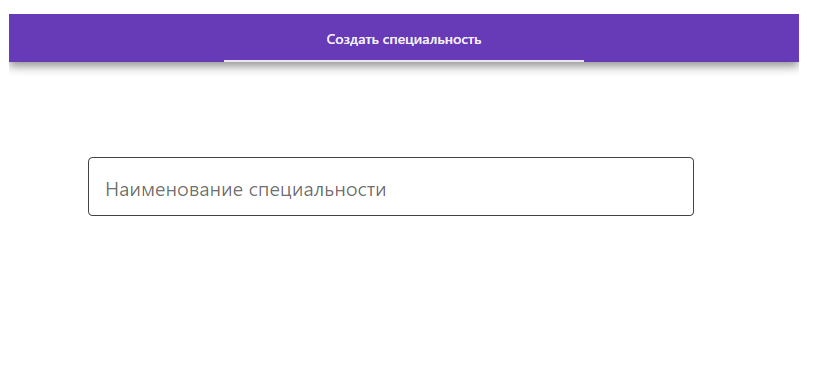


Рисунок 3.4.3 – Страница для создания специальности.

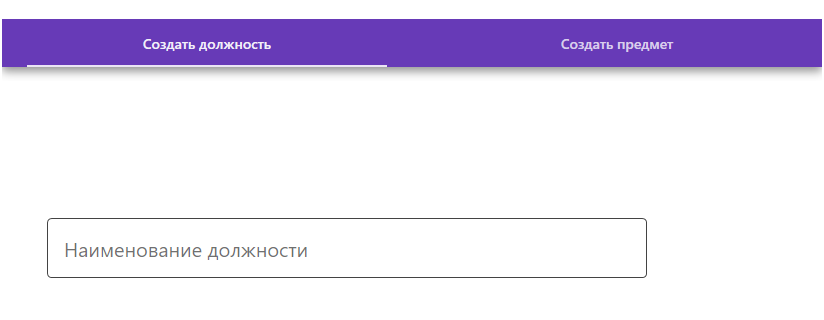


Рисунок 3.4.4 – Страница для создания должности и предмета

Заключение

Данная курсовая работа состоит из вводной, теоретической и практической частей. В курсовой работе были рассмотрены вопросы, касающиеся возможности прохождения тестирования для студента и возможности добавления тестирования для учителя. Темы должны состоять из тем школьных предметов.

Было разработано приложение, которое автоматизирует работу преподавателя, дает возможность добавления тестирования. Можно наглядно увидеть, как происходит и функционирует это приложение.

В ходе написания проекта была освоена и закреплена работа со средой разработки Microsoft Visual Studio и языком программирования C#. Цель и задачи были достигнуты, а именно «Разработка приложения для тестирования студентов по школьным предметам», путем создания приложения, представляющего собой программную реализацию.

Литература

1. Рихтер CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C# / Рихтер, Джефри. - М.: Питер, 2020. - 656 c.
2. Нейгел, К. C# 2005 для профессионалов / К. Нейгел. - М.: Вильямс, 2020. – 927 c.
3. Хорев П. Б. Объектно-ориентированное программирование с примерами на С# [Текст]: учебное пособие / П. Б. Хорев — 3-е изд. — Москва: Форум, Инфра-М, 2016. — 200 c.
4. Милов, А. В. Основы программирования в задачах и примерах / А.В. Милов. - М.: Фолио, 2018. - 400 c.
5. Грин Дж. Изучаем C# / Дж. Грин, Э. Стиллмен. – СПб.: «Питер», 2012.
6. Основы языка программирования C#: [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/C\_Sharp
7. Microsoft Visual Studio [Электронный ресурс]. URL <https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio>
8. Microsoft SQL Server [Электронный ресурс] URL <https://ru.wikipedia.org/wiki/Microsoft_SQL_Server>
9. XAML [Электронный ресурс] URL https://ru.wikipedia.org/wiki/XAML
10. Основы работы с платформой WPF: [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf (дата 20.06.2022).
11. Windows Presentation Foundation: [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Windows\_Presentation\_Foundatio (дата 20.05.2022).