МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Специализация Программирование интернет-приложений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированное программирование»

Тема Программное средство составления и прохождения тестов

Исполнитель

Студент (ка) 2 курса группы 4 Савчук Алёна Михайловна

(Ф.И.О.)

Руководитель Радиванович Дмитрий Александрович

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Комиссия Радиванович Дмитрий Александрович

(подпись)

Минск 2018

**Содержание**

[**Введение** 3](#_Toc515300275)

[**1. Постановка задачи** 4](#_Toc515300276)

[**1.1 Актуальность решаемой задачи** 4](#_Toc515300277)

[**1.2 Обзор прототипов** 4](#_Toc515300278)

[**2. Разработка архитектуры проекта** 8](#_Toc515300279)

[**2.1 Выбор используемых технологий** 8](#_Toc515300280)

[**2.2 Диаграмма вариантов использования** 8](#_Toc515300281)

[**2.2 Диаграмма классов** 10](#_Toc515300282)

[**3. Руководство пользователя** 12](#_Toc515300283)

[**4. Тестирование** 15](#_Toc515300284)

[**Список использованных источников** 18](#_Toc515300285)

[**Приложение А** 19](#_Toc515300286)

**Введение**

Не секрет, что сила в знании, и так было во все времена. Без стремления к познанию человечество перестало бы развиваться и осталось бы на первобытном уровне. Все мы хотим стать образованными людьми, окончить школу, затем ВУЗ, получить хорошую работу, где можно применить свои знания на практике. Вопросы, связанные с получением знаний и их применением, будут актуальны всегда.

Все большую популярность набирают тесты, как средство проверки знаний и выявления личностных характеристик человека. Тестирование – это исследовательский метод, который позволяет выявить уровень знаний, умений и навыков, способностей и других качеств личности, а также их соответствие определенным нормам путем анализа способов выполнения испытуемым ряда специальных заданий. Такие задания принято называть тестами. Тест – задания, дающие возможность быстро выявить и оценить степень развития определенных психологических качеств, а также уровень знаний, умений и навыков.

Целью курсового проекта является разработка программного средства для создания и прохождения тестов, которое позволяет создавать свои тесты и тестировать учеников, студентов или коллег на знания по той или иной теме, а также позволяет отслеживать результаты пройденных тестов.

Тест может содержать неограниченное количество вопросов и ответов. Есть возможность редактирования теста, добавление картинки к вопросу, добавление вопросов из базы данных.

Программа имеет весьма простой интерфейс. Отсутствуют отвлекающие детали, все интуитивно понятно. Для его разработки использовалась технология WPF – обширный API-интерфейс для создания настольных графических программ, имеющих насыщенный дизайн и интерактивность. Для работы с WPF использовался объектно-ориентированный язык программирования С#, разработанный для создания приложений на платформе Microsoft .NET Framework и XAML – язык декларативного описания интерфейса, основанный на XML.

Для хранения тестов и результатов их прохождения используется база данных SQL Server.

**1. Постановка задачи**

**1.1 Актуальность решаемой задачи**

Тема моего курсового проекта «Программное средство составления и прохождения тестов» выбрана не случайно. Актуальность и выбор темы состоит в том, что проблема контроля за уровнем знаний, умений и навыков учащихся – вечная проблема образования, и в разные времена предлагалось решать её по-разному, преобладали то одни формы контроля, то другие, создавались контролирующие системы. Тест может служить методом педагогического измерения на любом этапе обучения. Тестовые задания эффективны и для самопроверки знаний. Тема является ещё актуальнойи потому, что итоговой формой контроля знаний учащихся за курс основной школы является централизованное тестирование(ЦТ), основу которого составляют тестовые задания.

Тесты как система оценки успеваемости, имеют целый ряд положительных характеристик, позволяющих:

* проверить качество усвоения учащимися теоретического и практического материала;
* оживить процесс обучения, вводя не только новую для учащихся форму контроля, но и различные виды тестов;
* сэкономить учебное время, затраченное на опрос, и личное время учителя, идущее на проверку результатов выполненной работы;
* обеспечить оперативность проверки выполненной работы.

Однако метод тестирования обладает рядом недостатков:

* большая вероятность выбора ответа наугад;
* затруднение со стороны учителя, а чаще невозможность проследить логику рассуждений учащихся;
* категоричность оценки выполнения задания – задание выполнено правильно и полностью и задание не выполнено.

**1.2 Обзор прототипов**

Программное средство *«Конструктор тестов»*

Конструктор тестов - программа для создания и проведения тестов. Приложение позволяет:

* создавать тесты из пяти различных типов вопросов;
* форматировать текст в вопросах;
* выводить тесты и результаты тестирования на печать;
* проверять орфографию;
* настраивать интерфейс приложения;
* использовать неограниченное количество тем и вопросов;
* задавать каждому вопросу индивидуальную оценку в баллах;
* искать вопросы в базе;
* включать в вопросы мультимедийные файлы (картинки, звук, видео).

Программа содержит три модуля: один предназначен для составления тестов и заполнения базы вопросов (рисунок 1.1); второй для проведения тестирования и третий - для оценки результатов тестирования.

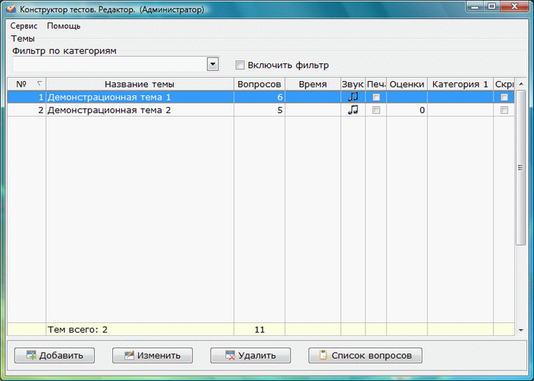


Рисунок 1.1 – Интерфейс для составления теста и заполнения базы вопросов

Тестируемый имеет возможность пропускать сложные вопросы. Преподаватель может приостанавливать тестирование, ограничивать его временными рамками, а также централизованно получать данные о прохождении тестов. Поддерживается разграничение прав доступа, резервное копирование и синхронизация базы. С сайта разработчиков программы можно загрузить множество вариантов готовых тестов на самую разнообразную тематику: ПДД, ЕГЭ, русский, французский и английский языки, экология, информатика, математика и др.

Программное средство «Универсальный тест»

Универсальный тест - приложение для создания тестов и проведения тестирования. Программа содержит в себе два модуля. Модуль под названием Tester позволяет проверять знания учащихся, студентов и персонала при помощи готовых тестов на самую разнообразную тематику - по русскому и английскому языку, информатике, ОБЖ, геометрии, физике, медицине, социологии, религиоведению, бухгалтерскому учету и т.д. База имеющихся тестов состоит из трех групп - тестов для школьников, тестов для ВУЗов, ПТУ и техникумов и тестов для проверки профессиональных знаний. Эти тесты составлялись пользователями предыдущих версий программы. Во время тестирования перед пользователем выводится серия вопросов с вариантами ответов (рисунок 1.2). Вопросы выбираются из базы в случайном порядке.

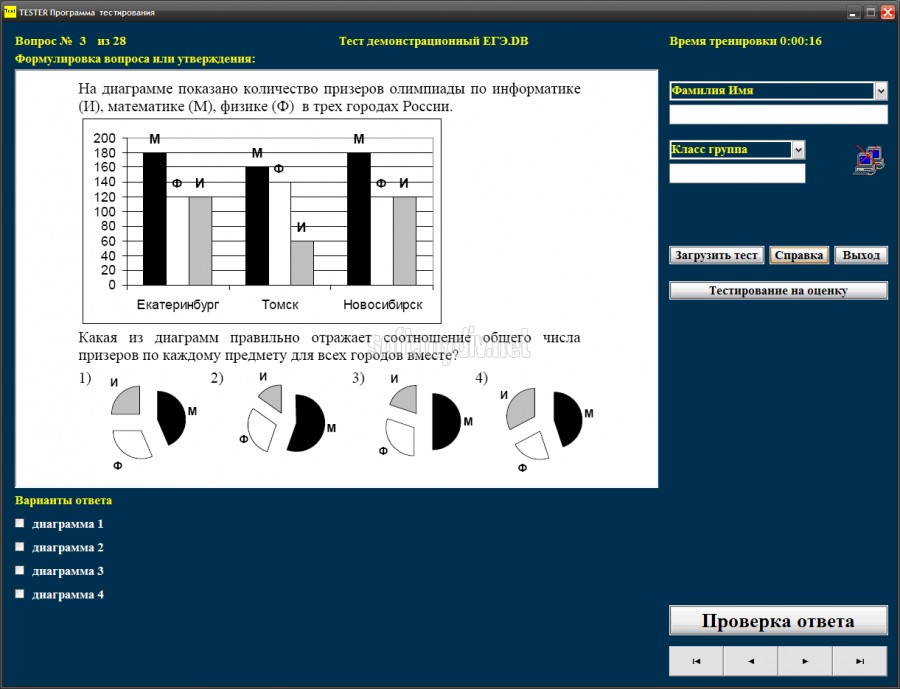


Рисунок 1.2 – Интерфейс для тестирования

При помощи модуля Test maker можно составлять и собственные тесты. В них можно использовать графику и тексты с форматированием. Доступ к модулю Test maker предоставляется либо на платной основе, либо в обмен на составленный пользователем собственный тест. Имеется возможность тестирования по локальной сети и сбора полученных результатов при помощи встроенного сервера.

Программное средство *«AnsTester»*

AnsTester - приложение для создания разнообразных тестов, основанных на системе начисления баллов. AnsTester изначально предназначался для проведения компьютерного тестирования в средней школе. Помимо тестирования, программа позволяет проводить различные опросы. В пакет входят несколько модулей - один предназначен для проведения тестирования (рисунок 1.3), второй - для просмотра результатов тестов и третий - для создания тестов.

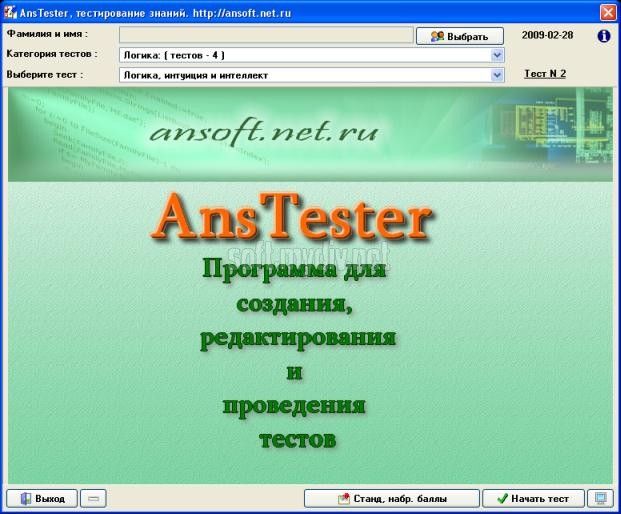


Рисунок 1.3 – Интерфейс для проведения тестирования

# **2. Разработка архитектуры проекта**

## **2.1 Выбор используемых технологий**

При разработке приложения были использованы:

* интегрированная среда разработки Microsoft Visual Studio 2017;
* программная платформа .NET Framework 4.6.1;
* язык программирования C#;
* расширяемый язык разметки XAML;
* технология WPF;
* Entity Framework;
* Material Design;
* Microsoft SQL Server;

*Entity Framework* представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работает с объектами.

*Material Design* – дизайн программного обеспечения и приложений операционных систем Android и Chrome OS от компании Google. Material Design зиждется на четырех основных принципах: тактильные поверхности, полиграфический дизайн, осмысленная анимация, адаптивный дизайн.

*Microsoft SQL Server* – система управления реляционными базами данных, разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL.

## **2.2 Диаграмма вариантов использования**

На диаграммах вариантов использования отображается взаимодействие между вариантами использования, представляющими функции системы, и действующими лицами, представляющими людей или системы, получающие или передающие информацию в данную систему. Из диаграмм вариантов использования можно получить довольно много информации о системе. Этот тип диаграмм описывает общую функциональность системы. Варианты использования показаны на диаграмме (рисунок 2.1).

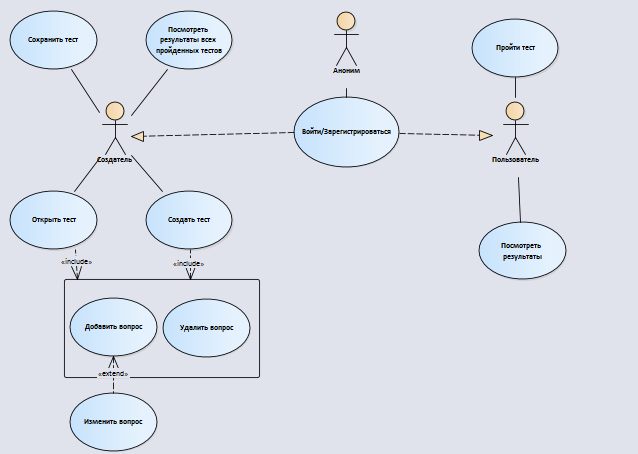


Рисунок 2.1 – Диаграмма вариантов использования

Прецедент «Войти/Зарегистрироваться» заключается в авторизации (т.е. выбора роли создателя/пользователя). В случае успешного авторизации вход в учетную запись.

Если мы зашли как «Создатель»:

Прецедент «Сохранить тест» заключается в сохранении теста в базу данных.

Прецедент «Посмотреть результаты всех пройденных тестов» заключается в просмотре результатов пройденных тестов.

Прецедент «Открыть тест» в открытии из базы данных ранее созданного теста.

Прецедент «Создать тест» заключается в создании нового теста.

Прецедент «Добавить вопрос» заключается в добавлении нового вопроса к текущему тесту.

Прецедент «Удалить вопрос» заключается в удалении вопроса из теста.

Прецедент «Изменить вопрос» заключается в изменении выбранного вопроса.

## **2.2 Диаграмма классов**

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. На таких диаграммах отображается множество классов, интерфейсов, коопераций и отношений между ними.

Диаграмма классов может отражать различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений.

Диаграмма классов приложения представлена в **приложении А**.

Также в проекте есть 10 классов окон и 1 класс пользовательского элемента управления. Классы окон представлены на рисунке 2.2.

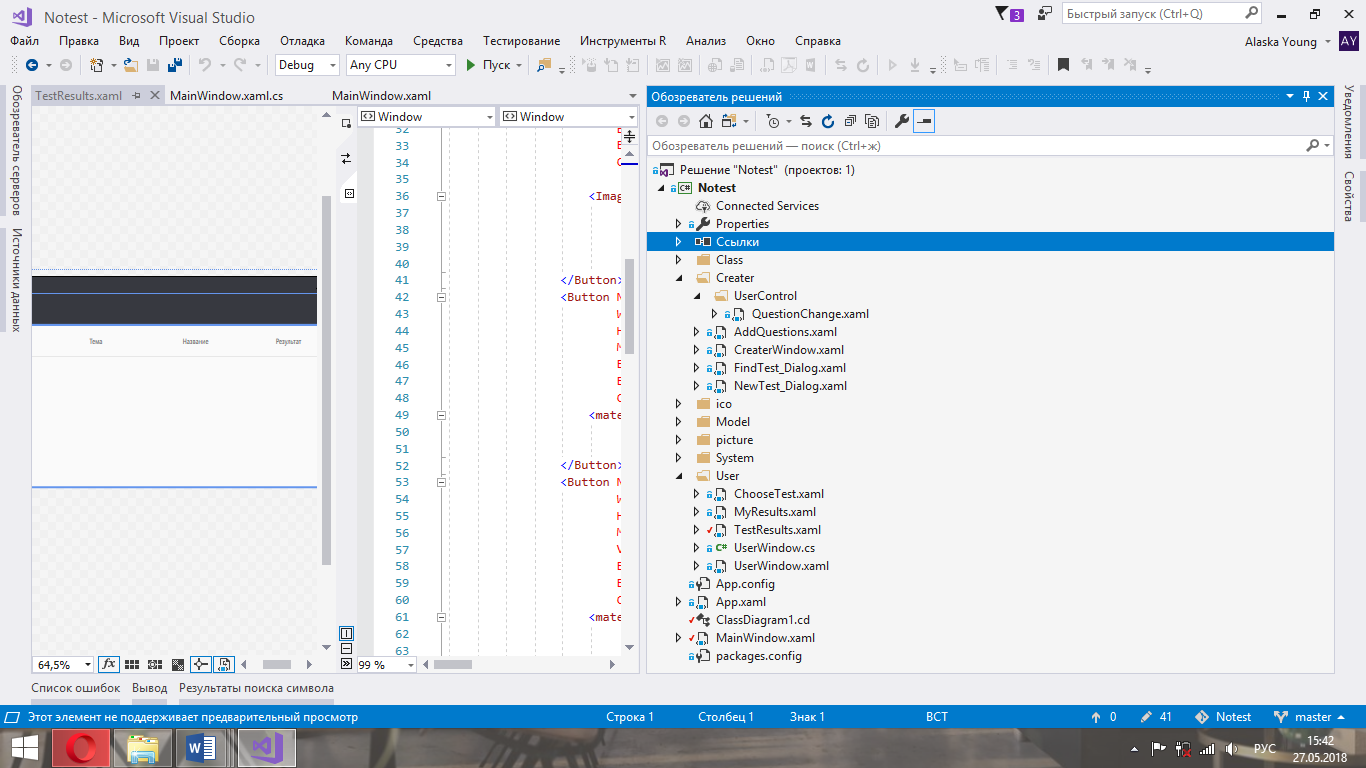


Рисунок. 2.2 – Классы окон

Окно MainWindow является первым окном, которое видит пользователь запустив приложение. Оно позволяет зарегистрироваться или войти в приложение под определённой ролью: «пользователь» или «создатель».

Окно CreaterWindow служат для создания теста. В нём располагается пользовательский элемент управления QuestionChange, который содержит в себе необходимые компоненты для работы с вопросом.

Окно AddQuestions служит для добавления готовых вопросов из базы данных по теме, соответствующей теме текущего теста.

Окно FindTest\_Dialog предназначено для открытия теста из базы данных.

Окно NewTest\_Dialog служит для создания нового теста.

**2.3 Проектирование базы данных**

Для реализации поставленной задачи была создана база данных Database. Для её создания использовалась система управления реляционными базами данных Microsoft SQL Server, технология Entity Framework и подход Code First.

База данных состоит из 5 таблиц, перечисленных ниже (рисунок 2.3).

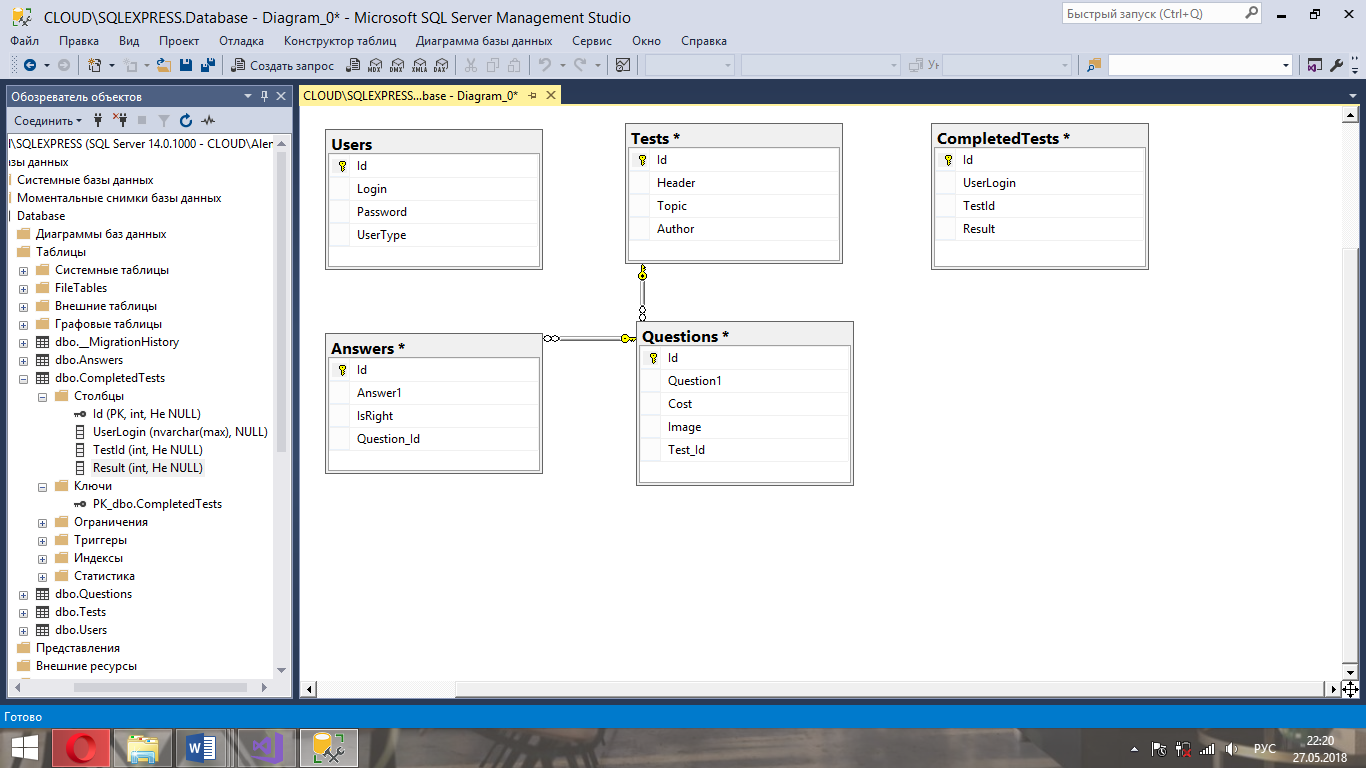


Рисунок 2.3 – Диаграмма базы данных

В таблице Users хранятся все пользователи, зарегистрированные в программе, Tests – хранит все созданные тесты, Questions – хранит все созданные вопросы, Answers – хранит ответы к вопросам, а также таблица CompletedTests, которая хранит пройденные тесты.

Таблица Users состоит из четырёх столбцов:

* Id – хранит идентификатор пользователя;
* Login – хранит псевдоним пользователя;
* Password – хранит пароль пользователя;
* UserType – хранит тип пользователя.

Таблица Tests состоит из четырёх столбцов:

* Id – хранит идентификатор теста;
* Header – хранит название теста;
* Topic – хранит тему теста;
* Author – хранит автора теста.

Таблица Questions состоит из пяти столбцов:

* Id – хранит идентификатор вопроса;
* Question1– хранит сам вопрос;
* Cost – хранит ценность вопроса;
* Image – хранит изображение, прикреплённое к вопросу, если такое имеется;
* Test\_Id – хранит идентификатор теста, к которому относится вопрос.

Таблица Answers состоит из четырёх столбцов:

* Id – хранит идентификатор ответа;
* Answer1 – хранит сам ответ;
* IsRight – хранит булевское значение, определяющее является ли ответ верным;
* Question\_Id – хранит идентификатор, к которому относится ответ.

Таблица ComletedTests состоит из четырёх столбцов:

* Id – идентификатор результата;
* UserLogin – псевдоним пользователя, выполнившего тест;
* TestId – идентификатор завершённого теста;
* Result – результат выполнения теста.
* Date – дата выполнения.

# **3. Руководство пользователя**

После запуска приложения пользователь увидит главное окно. Оно представлено на рисунке 3.1. Окно предоставляет пользователю возможность зарегистрироваться или зайти под ранее зарегистрированным псевдонимом.

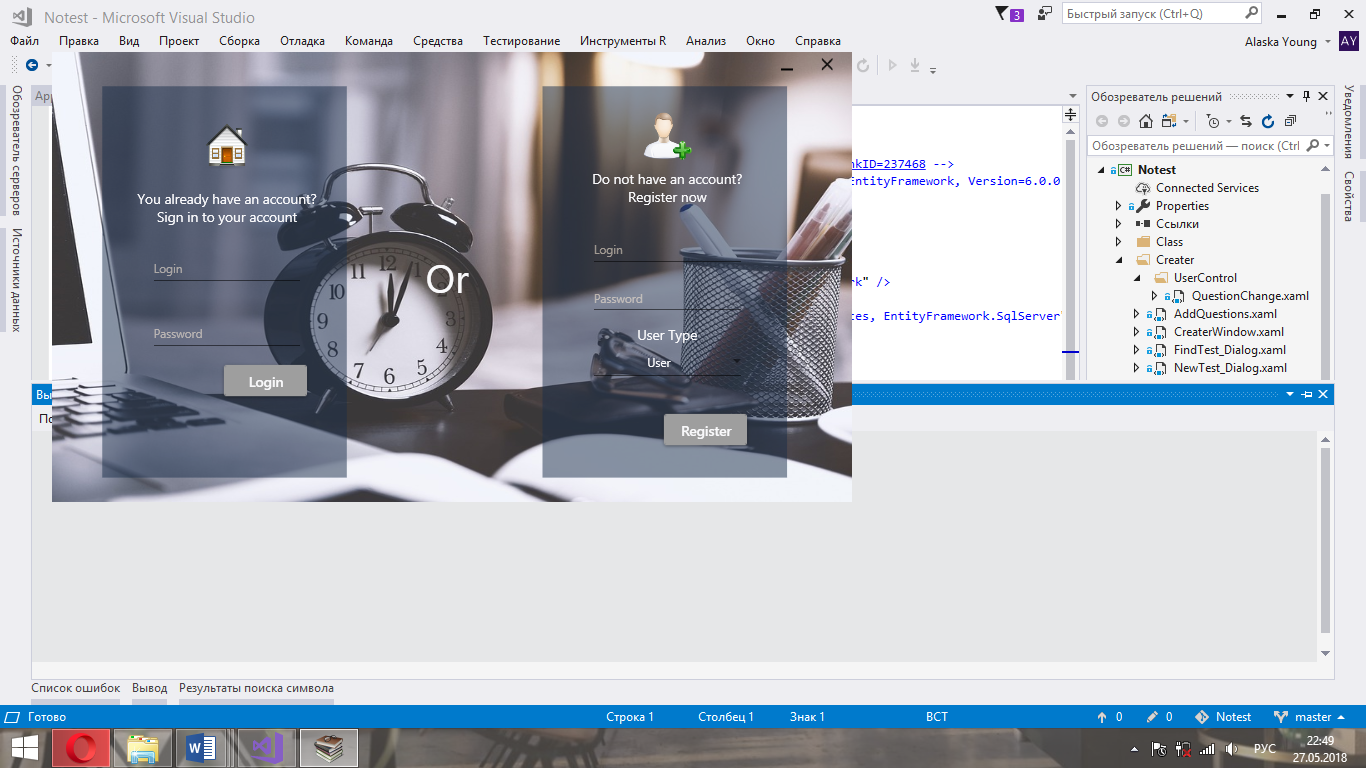


Рисунок 3.1 – Главное окно

Если ввести данные и нажать на кнопку «Login» (или «Register»), то пользователь переместится в окно, соответствующее выбранной роли. Если такой ролью был «Creator», то пользователь увидит окно, представленное на рисунке 3.2.

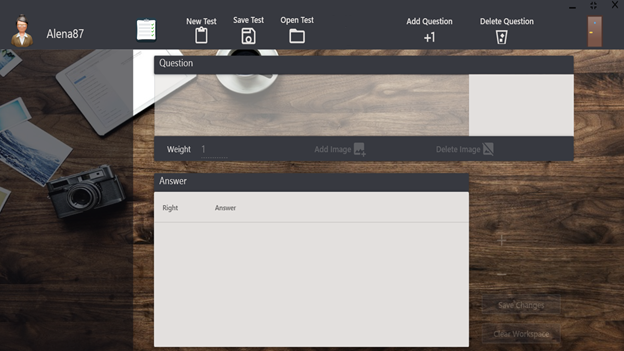


Рисунок 3.2 – Окно создателя

При нажатии на кнопку «New Test» появится диалоговое окно, представленное на рисунке 3.3, в которое нужно будет ввести тему и название теста.

При нажатии на кнопку «Save Test» пользователь увидит уведомление о том, что тест сохранён или же уведомление об ошибке, в случае её возникновения.

При нажатии на кнопку «Print Test» пользователь увидит уведомление о том, что тест готов к печати (сохранён в текстовый файл) или же уведомление об ошибке, в случае её возникновения.

Дведь в правом верхнем углу представляет собой кнопку выхода в главное окно.

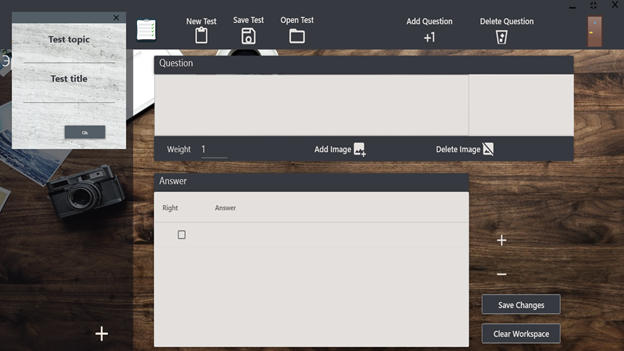


Рисунок 3.3 – Главное окно пользователя

Процесс редактирование теста имеет интерфейс, представленный на рисунке 3.4.

При работе с вопросом пользователь может добавить к вопросу изображение, указать его стоимость. Вопрос может иметь неограниченное количество ответов. Правильных ответов может быть много, один или вовсе ни одного. Любой вопрос из списка слева можно изменять. Для этого нужно его выбрать, изменить и нажать кнопку «Save Changes». Также есть кнопка «Clear Workspace» для очистки области работы с вопросом.

Для удаления вопроса необходимо выбрать вопрос в списке слева, нажать на кнопку «Delete Question».

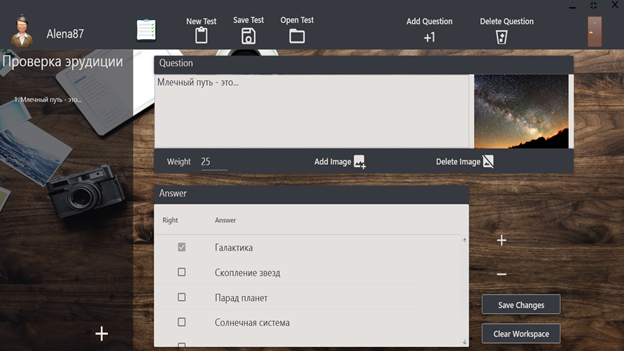


Рисунок 3.4 – Интерфейс для создания вопроса

Также имеется возможность добавления существующего вопроса из базы данных. Для этого необходимо нажать на «плюс», который находится внизу списка вопросов текущего теста. После нажатия на эту кнопку пользователь увидит окно (рисунок 3.5), в котором будут все вопросы из базы данных, относящиеся к теме текущего теста.

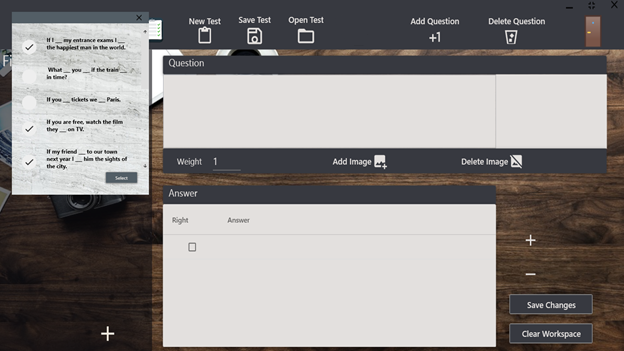


Рисунок 3.5 – Интерфейс добавления вопроса из базы данных

Посмотреть результаты всех пользователей можно нажав на кнопку в виде списка (рисунок 3.6), которая располагается на верхней панели. Там видна информация о том, кто и во сколько проходил какой-либо тест, а также результат прохождения теста.



Рисунок 3.6 – Мониторинг успеваемости

# **4. Тестирование**

При тестировании данного приложения были проиграны сценарии, которые могли бы привести к ошибке.

В момент регистрации возможна ситуация, когда пользователь вводит данные, включающие в себя запрещённые символы. Обработка данного исключения продемонстрирована на рисунке 4.1.

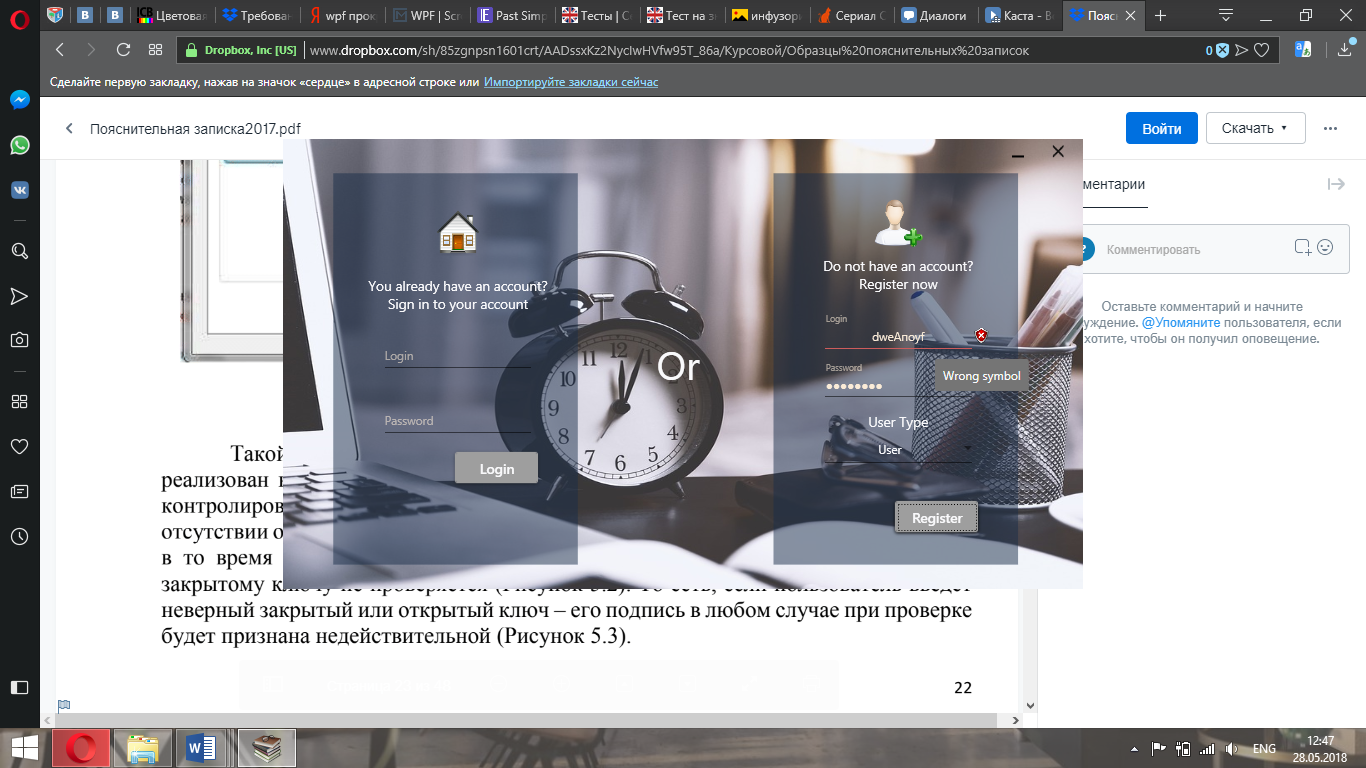


Рисунок 4.1 – Тестирование регистрации

В момент регистрации возможна также ситуация, когда пользователь вводит псевдоним, который уже используется другим. Обработка данного исключения продемонстрирована на рисунке 4.2.

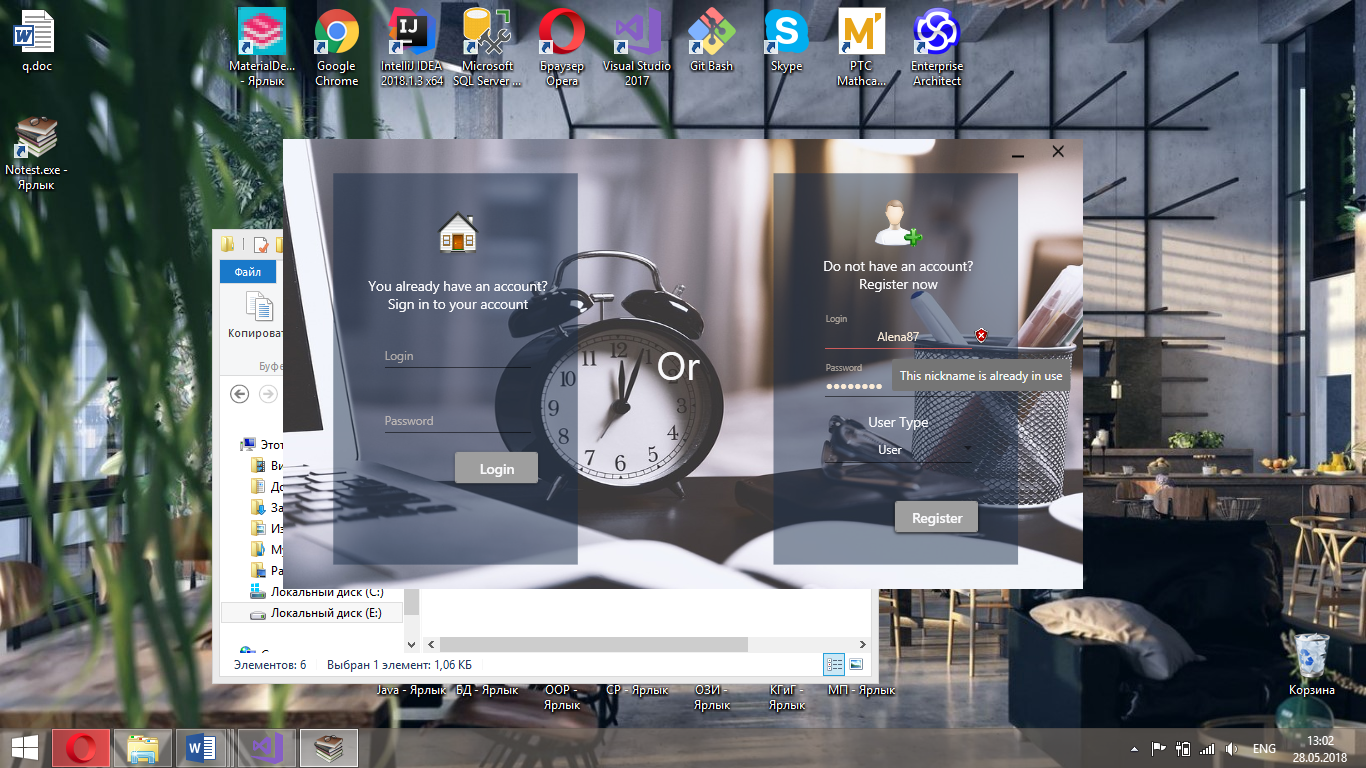


Рисунок 4.2– Тестирование регистрации

Во время работы с вопросом, если пользователь попытается ввести что-либо кроме цифр в поле «Weight», он увидит предупреждение о некорректных введённых данных. Обработка данного исключения продемонстрирована на рисунке 4.3.

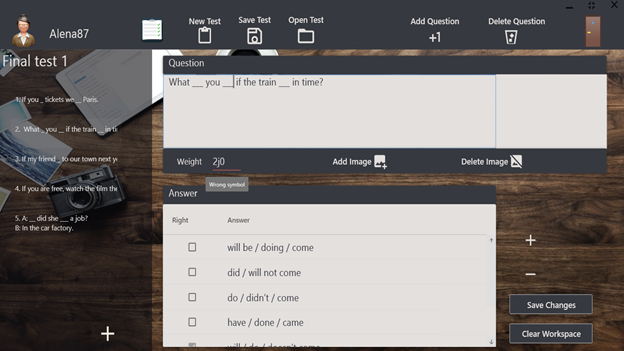


Рисунок 4.3 – Тестирование введённых данных

Если пользователь попытается сохранить тест, в котором нет вопросов, то увидит уведомление об ошибке (рисунок 4.4).

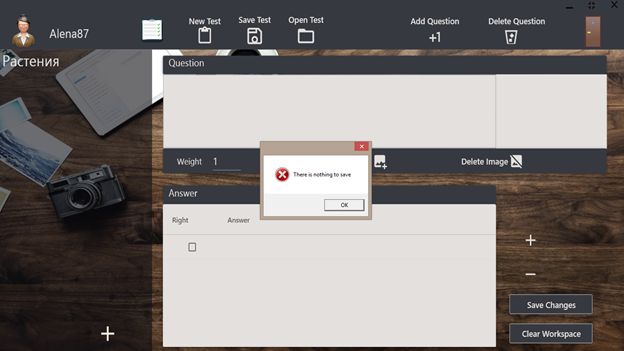


Рисунок 4.3 – Тестирование сохранения теста

**Заключение**

В данном курсовом проекте было разработано программное средство «Notest» при помощи языка программирования C#, API-интерфейса Windows Presentation Foundation, технологии Entity Framework и базы данных Microsoft SQL Server. При выполнении курсового проекта использовались принципы и приемы ООП.

Для хранения исходного кода и удобства контроля версий проекта использовался крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки – Github.

В ходе выполнения курсового проекта было создано настольное приложение (desktop), которое выполняет следующие функции:

* авторизация пользователя;
* регистрация пользователя;
* создание нового теста;
* открытие существующего теста;
* добавление созданного вопроса в тест;
* добавление готового вопроса в тест;
* добавление картинки к вопросу;
* удаление вопроса;
* сохранение теста;
* сохранение теста в текстовый документ;
* мониторинг результатов прохождения теста.

Данное программное средство имеет удобный и понятный интерфейс, реагирует на ошибочный ввод данных.

Данное приложение соответствует поставленной задаче и отвечает всем требованиям, необходимым для пользования данным программным средством.

**Список использованных источников**

1. Пацей, Н.В. Курс лекций по языку программирования С# / Н.В. Пацей. – Минск: БГТУ, 2016. – 175 с.
2. Пацей, Н.В. Технология разработки программного обеспечения / Н.В. Пацей. – Минск: БГТУ, 2016. – 129 с.
3. MSDN сеть разработчиков в Microsoft [Электронный ресурс] / Режим доступа: <http://msdn.microsoft.com/library/rus/>
4. METANIT.COM Сайт о программировании [Электронный ресурс] / Режим доступа: <https://metanit.com>
5. ProfessorWeb .NET & Web Programming [Электронный ресурс] / Режим доступа: <https://professorweb.ru>
6. Stack Overflow [Электронный ресурс] / Режим доступа: <https://stackoverflow.com>
7. Хабр [Электронный ресурс] / Режим доступа: <https://habr.com>

**Приложение А**

