

На тему: **«Система тестирования»**

Студента группы W-ПВ122

Запорожца Евгения Игоревича

г. Кривой Рог 2022 г. 

Содержание

[ВВЕДЕНИЕ 3](#_heading=h.30j0zll)

[ПОСТАНОВКА ЗАДАЧИ 3](#_heading=h.1fob9te)

[Требования к функциональным характеристикам системы 3](#_heading=h.3znysh7)

[ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ ХАРАКТЕРИСТИКАМ 3](#_heading=h.2et92p0)

[ТРЕБОВАНИЯ К ПРОГРАММНЫМ ХАРАКТЕРИСТИКАМ 4](#_heading=h.tyjcwt)

[ПРОЕКТИРОВАНИЕ СИСТЕМЫ 4](#_heading=h.3dy6vkm)

[Проектирование интерфейса 4](#_heading=h.1t3h5sf)

[Проектирование алгоритмов 4](#_heading=h.4d34og8)

[Проектирование диаграммы классов 5](#_heading=h.2s8eyo1)

[Проектирование базы данных](#_heading=h.17dp8vu) 8

[РУКОВОДСТВО ПРОГРАММИСТА](#_heading=h.3rdcrjn) 13

[РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 1](#_heading=h.26in1rg)6

[РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ: УЧИТЕЛЬ](#_heading=h.x19h4fwdp9ij) [18](#_heading=h.1t3h5sf)

[РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ: СТУДЕНТ](#_heading=h.kpzj2qirnk6t)28

[СПИСОК ЛИТЕРАТУРЫ](#_heading=h.lnxbz9) 36

# ВВЕДЕНИЕ

В наше время, когда образование стаёт всё более цифровизированным, а обучение всё чаще происходит в онлайн-формате, востребованным является предоставить эффективные, удобные и интуитивно понятные программы, предназначенные для создания тестов и проведения тестирования студентов. И именно для этого создана эта программа.

# ПОСТАНОВКА ЗАДАЧИ

База данных содержит категории и тесты в них. Программа просматривает список категорий и тестов, и в зависимости от учётной записи пользователя позволяет студентам проходить тесты и просматривать свои результаты, а учителям создавать, редактировать и удалять тесты и категории.

## Требования к функциональным характеристикам системы

* Тест может иметь произвольное количество вопросов, каждый вопрос может иметь произвольное количество вариантов ответа.
* База данных содержит два типа пользователей (teacher, student).
* Учителям доступны окна для добавления, редактирования и удаления категорий, тестов, вопросов, вариантов ответа.
* Студентам доступны окна для сдачи теста.
* Программа должна выводить список всех доступных категорий и тестов в них.
* При прохождении теста все вопросы выводятся на панель (номер, вопрос, варианты ответа), оставшееся время на ответ (если ограничено), оставшееся время на тест (если ограничено), и кнопку подтверждения ответа.
* После прохождения теста система выдаёт оценку и подробную информацию о результате, такую как время прохождения, количество правильных ответов и др.
* Добавление и редактирование категорий и тестов выполняется в отдельных окнах.

# ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ ХАРАКТЕРИСТИКАМ

Процессор 1 ГГц

ОЗУ 512 Мб

Дисковое пространство (минимум)

32-разрядная версия 4,5 ГБ

64-разрядная версия 4,5 ГБ

# ТРЕБОВАНИЯ К ПРОГРАММНЫМ ХАРАКТЕРИСТИКАМ

Компьютер на базе Windows 7+, с установленным .net 6.0+.

# ПРОЕКТИРОВАНИЕ СИСТЕМЫ

## Проектирование интерфейса

При проектировании интерфейса программы использовался Material Design In Xaml Toolkit, который изменяет внешний вид многих стандартных компонентов WPF и предоставляет большое количество стилей для них, основанных на системе дизайна Google's Material Design. С его помощью интерфейс удалось сделать более эргономичным и удобным для пользователя.

## Проектирование алгоритмов

Есть два действующих лица: teacher и student. И teacher и student имеют доступ к категориям и тестам, но только teacher может ими управлять, и только student может их проходить. Оба могут просматривать личную статистику, а teacher также и общую статистику студентов и тестов (Рис. 1)



Рис. 1

## Проектирование диаграммы классов

Всем представлениям соответствуют свои модели представления. Для реализации моделей представления используется самописная библиотека MvvmBaseViewModels. Все модели представления наследуются от одного из базовых абстрактных классов - ViewModelBase, ValidatableViewModelBase или NavigationViewModelBase. Последние, помимо основного функционала, предоставляют также поддержку валидации свойств или навигации соответственно. Каждый базовый класс реализует интерфейсы ICloseable и IDisposable. Реализация ICloseable предоставляет возможность модели представления вызывать закрытие окна, а IDisposable предоставляет механизм освобождения неуправляемых ресурсов, если таковые используются в модели представления. Помимо этого все базовые классы наследуются от абстрактного класса ObservableObject из NuGet-пакета Community.Toolkit.Mvvm, который реализует интерфейс INotifyPropertyChanged, используемый для уведомлений о том, что значение какого-то свойства изменилось. Помимо этого модели представления хранят в качестве свойств экземпляры классов

RelayCommand и AsyncRelayCommand, которые реализуют интерфейс ICommand. Также все базовые классы моделей представления предоставляют события возникновения информационных сообщений, предупреждений, ошибок и критических ошибок, а также стандартные обработчики для них (Рис.2)

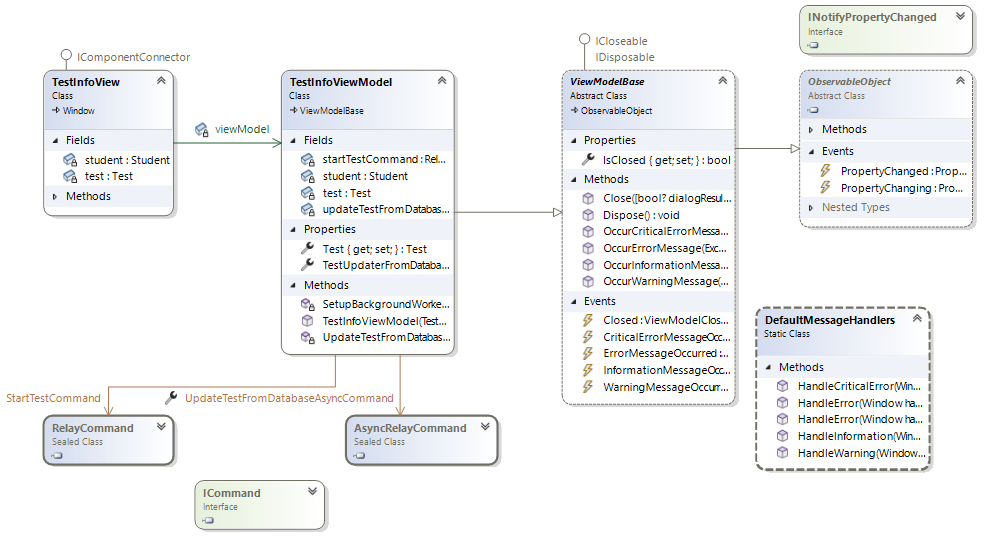


Рис. 2

В данной программе используется паттерн MVVM (Model-View-ViewModel)

(Рис. 3)

Каждое окно содержит лишь визуальное представление, а логику и данные содержит соответствующая модель представления.

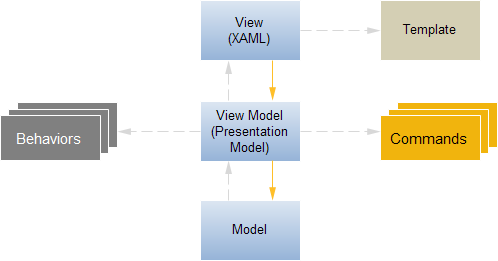


Рис.3

Например, представлению TestInfoView соответствует модель представления TestInfoViewModel, которая наследуется от базового класса ViewModelBase и содержит команды StartTestCommand (команда, которая запускает прохождение теста и открывает соответствующее модальное окно) и UpdateTestFromDatabaseAsyncCommand (команда, которая обновляет данные из базы данных). Данная модель представления работает с моделью dbo.Tests (таблица в базе данных MS SQL). (Рис. 4)

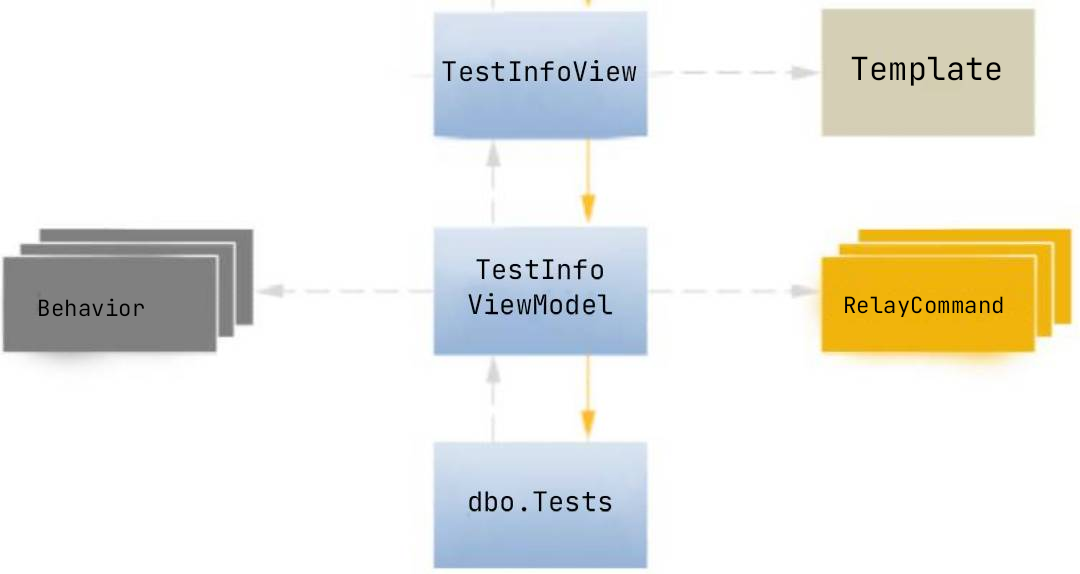


Рис.4

## Проектирование базы данных

Таблица **AnswerOptions**

| № | Название | Тип | Длина | Назначение |
| --- | --- | --- | --- | --- |
| 1 | Id | int |  | Первичный ключ |
| 2 | SerialNumberInQuestion | int |  | Порядковый номер варианта ответа в вопросе |
| 3 | QuestionId | int |  | Внешний ключ на Вопрос |
| 4 | Content | nvarchar | от 1 до 255 | Содержимое вопроса |
| 5 | IsCorrect | bit |  | Определяет является ли вариант ответа правильным/одним из правильных |

Таблица **Questions**

| № | Навзание | Тип | Длина | Назначение |
| --- | --- | --- | --- | --- |
| 1 | Id | int |  | Первичный ключ |
| 2 | TestId | int |  | Внешний ключ на Тест |
| 3 | SerialNumberInTest | int |  | Порядковый номер вопроса в тесте |
| 4 | Content | nvarchar | от 1 до 512 | Содержимое вопроса |
| 5 | PointsCost | float |  | Стоимость вопроса в баллах |
| 6 | NumberOfSecondsToAnswer | int? |  | Количество секунд на ответ |

Таблица **Tests**

| № | Название | Тип | Длина | Назначение |
| --- | --- | --- | --- | --- |
| 1 | Id | int |  | Первичный ключ |
| 2 | Name | nvarchar | от 1 до 255 | Название теста |
| 3 | MaximumPoints | int |  | Максимальное количество баллов за тест |
| 4 | NumberOfSecondsToAnswerEachQuestion | int? |  | Количество секунд на ответ для каждого вопроса |
| 5 | NumberOfSecondsToComplete | int? |  | Количество секунд на прохождение теста |
| 6 | IsAccountingForIncompleteAnswersEnabled | bit |  | Определяет нужно ли частично засчитывать баллы, если ответ неполный |
| 7 | CategoryId | int |  | Внешний ключ на Категорию |

Таблица **Categories**

| № | Название | Тип | Длина | Назначение |
| --- | --- | --- | --- | --- |
| 1 | Categories | int |  | Первичный ключ |
| 2 | Name | nvarchar | от 1 до 128 | Название категории |

Таблица **Teachers**

| № | Название | Тип | Длина | Назначение |
| --- | --- | --- | --- | --- |
| 1 | Id | int |  | Первичный ключ |
| 2 | Name | nvarchar | от 5 до 20 | Никнейм |
| 3 | HashedPassword | varchar | 128 | Хешированный пароль |
| 4 | FullName | nvarchar | от 1 до 128 | ФИО |

Таблица **Students**

| № | Название | Тип | Длина | Назначение |
| --- | --- | --- | --- | --- |
| 1 | Id | int |  | Первичный ключ |
| 2 | Name | nvarchar | от 5 до 20 | Никнейм |
| 3 | HashedPassword | varchar | 128 | Хешированный пароль |
| 4 | FullName | nvarchar | от 1 до 128 | ФИО |

Таблица **TestsTeacherOwners**

| № | Название | Тип | Длина | Назначение |
| --- | --- | --- | --- | --- |
| 1 | OwnedTestsId | int |  | Часть составного первичного ключа |
| 2 | OwnerTeachersId | int |  | Часть составного первичного ключа |

Таблица **TestResults**

| № | Название | Тип | Длина | Назначение |
| --- | --- | --- | --- | --- |
| 1 | Id | int |  | Первичный ключ |
| 2 | TestId | int |  | Внешний ключ на Тест |
| 3 | StudentId | int |  | Внешний ключ на Студента |
| 4 | Score | int |  | Оценка за тест |
| 5 | NumberOfCorrectAnswers | int |  | Количество правильных ответов |
| 6 | NumberOfIncorrectAnswers | int |  | Количество неправильных ответов |
| 7 | TestCompletionTime | time |  | Время, за сколько был пройден тест |
| 8 | AverageAnswerTime | time |  | Среднее время ответа на вопрос |
| 9 | CompletionDate | datetime2 |  | Дата и время, когда был пройден тест |

**Диаграмма структуры базы данных**

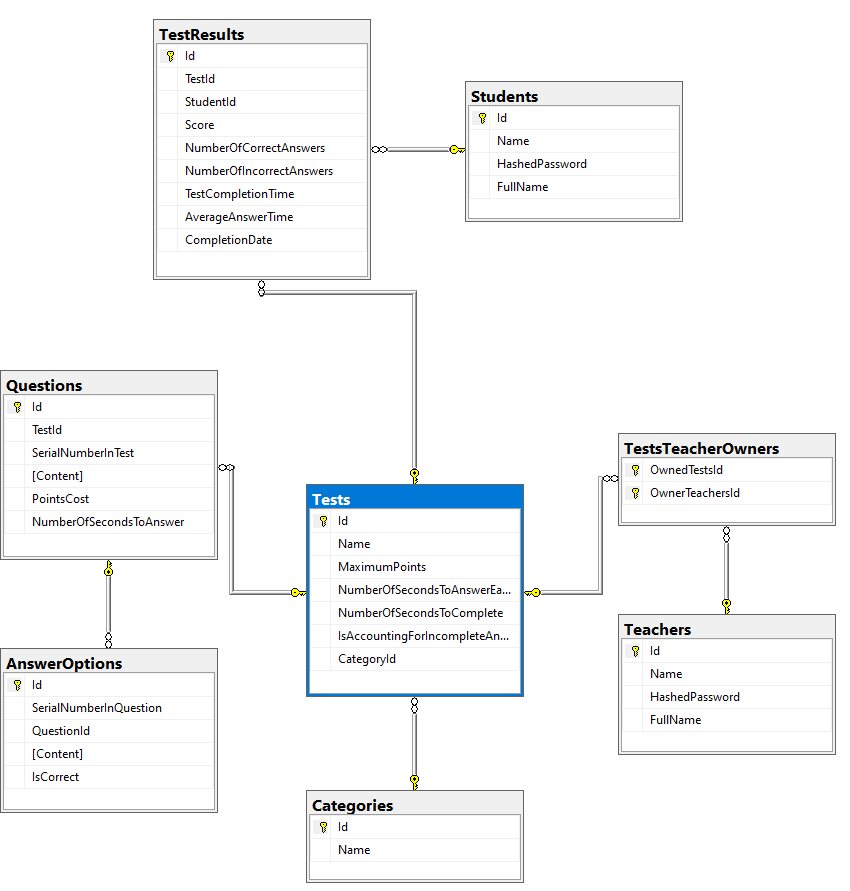


Рис. 5

# РУКОВОДСТВО ПРОГРАММИСТА

Чтобы корректно использовать программу, нужно открыть TestingSystem.sln в Visual Studio. Далее можно либо использовать базу данных по умолчанию если она доступна, либо свою. Для того чтобы использовать свою, в файле App.config нужно указать все 4 параметра, относящиеся к базе данных (databaseDataSource, databaseInitialCatalog, databaseAdminUserId, databasePassword). (Рис. 6)

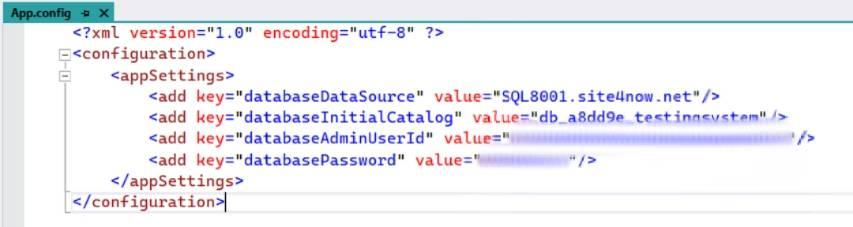


Рис.6

После этого нужно удалить папку Migrations, а затем заново создать миграцию базы данных и обновить БД. Для этого сначала нужно перейти во вкладку Tools -> NuGet Package Manager -> Package Manager Console (Рис. 7.1)

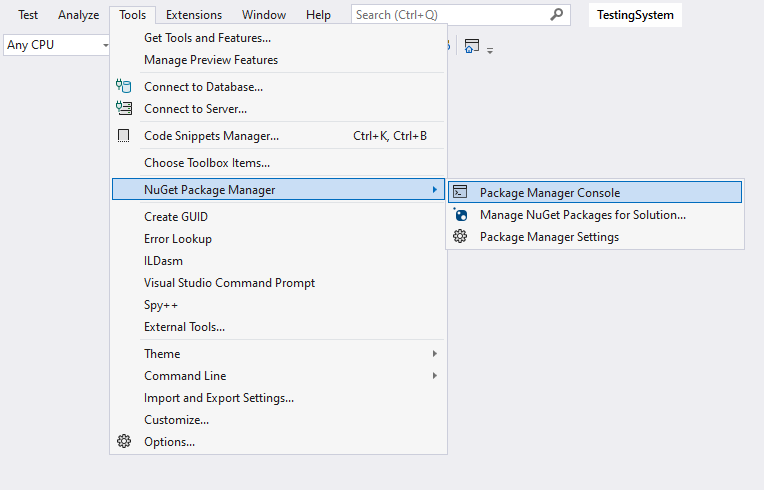


Рис. 7.1

И в открывшемся окне, предварительно убедившись что в комбобоксе Default Project выбран TestingSystem, прописать команду

*Add-Migration InitialCreate -Context TestingSystemDatabaseBuilderContext -OutputDir Migrations*

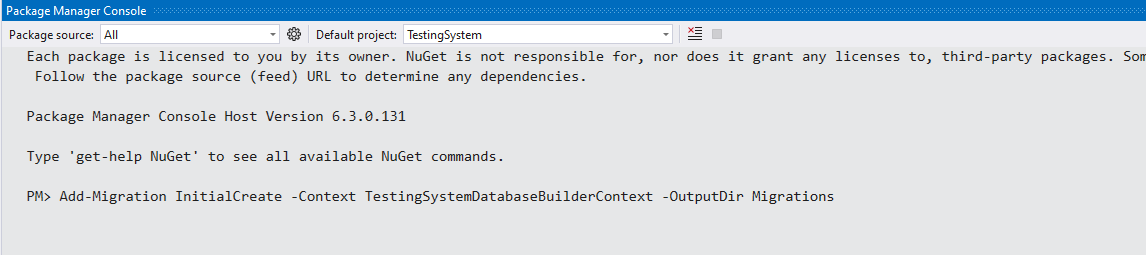


Рис. 7.2

И после того как добавление миграции завершится, нужно прописать команду

*Update-Database -Context TestingSystemDatabaseBuilderContext*

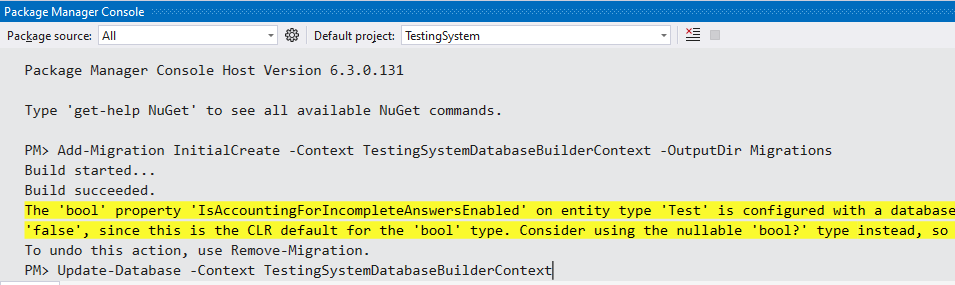


Рис. 7.3

Для правильной работы программы должны быть реализованы классы

RelayCommand и AsyncRelayCommand, которые реализуют интерфейс ICommand. В данной программе по умолчанию для этого используются NuGet пакет Community.Toolkit.Mvvm и расширение для него HappyStudio.Mvvm.Input.Wpf, которое содержит те же классы, но адаптированные под WPF. Кроме этого в сборке должны присутствовать зависимости на следующие dll-библиотеки (Рис. 8.1)

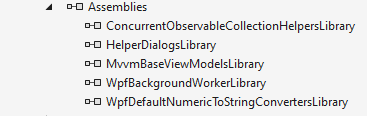


Рис. 8.1

Все эти библиотеки самописные, по умолчанию их можно найти в папке Libraries проекта.

Помимо них также должны присутствовать зависимости на следующие NuGet пакеты (Рис. 8.2)

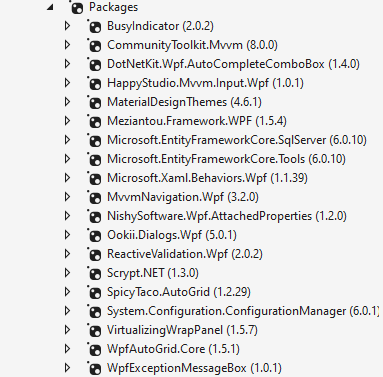
****

Рис. 8.2

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Перед стартом программы нужно убедиться в наличии файла App.config, и в том что все 4 параметра указаны. (Рис. 9)

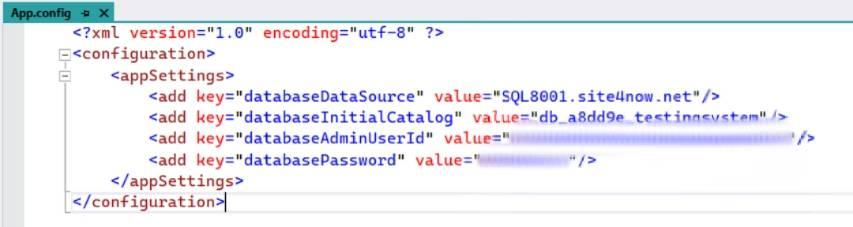


Рис. 9

При запуске программы открывается окно, где нужно ввести авторизационные данные пользователя teacher или student, либо зарегистрироваться (Рис. 10.1, 10.2)

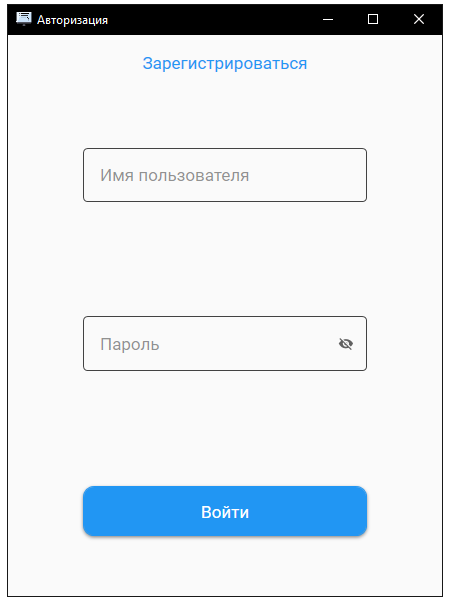


Рис. 10.1

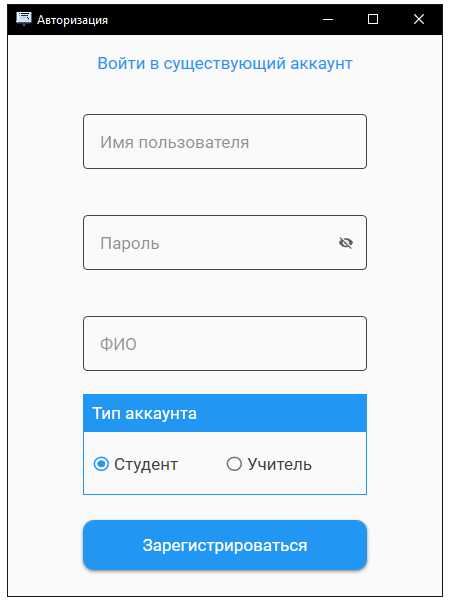
****

Рис. 10.2

После успешного входа или регистрации, открывается главное окно программы, которое отличается в зависимости от того, какой пользователь вошёл в систему.

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ: УЧИТЕЛЬ

После успешного входа или регистрации, открывается главное окно программы (Рис. 11.1)

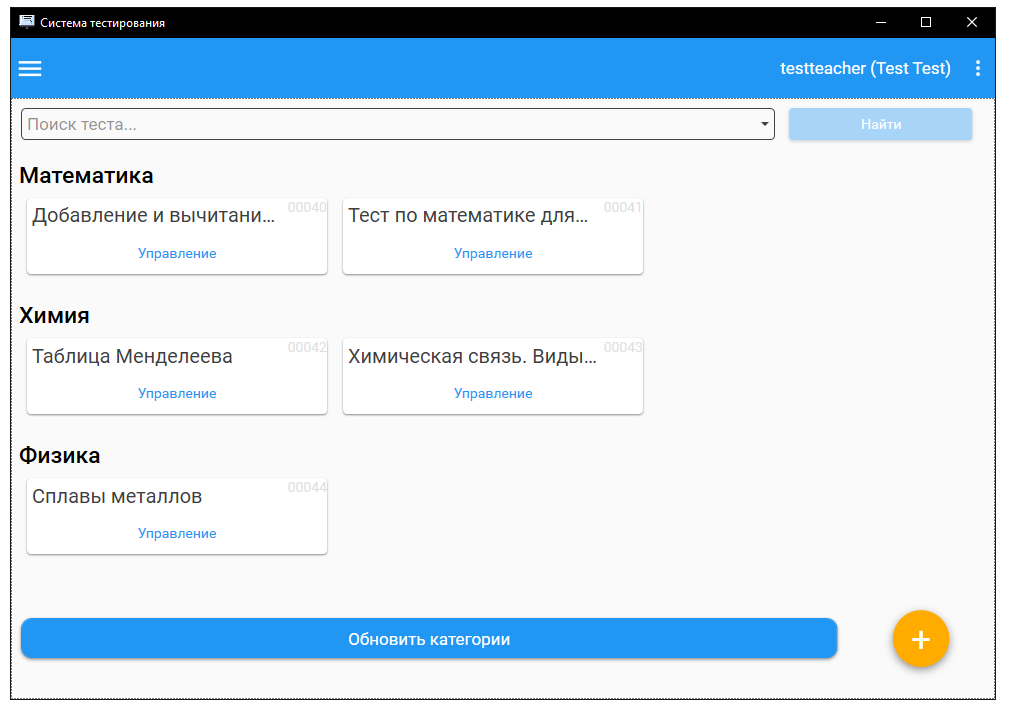


Рис. 11.1

Нажав на название категории, пользователь может отредактировать или удалить её, но только если имеет соответствующие права (Рис. 11.2)

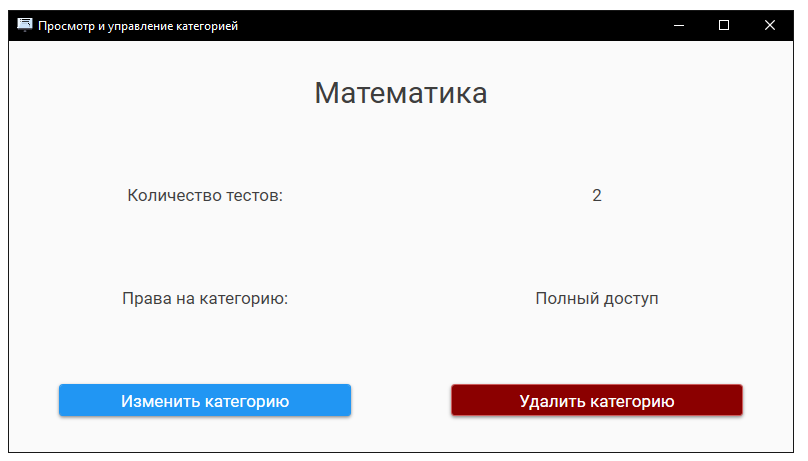


Рис. 11.2

Права на категорию определяются по следующим условиям:

* Если учитель является владельцем всех тестов в категории - он имеет полный доступ
* Если учитель является владельцем хотя бы одного теста в категории - он имеет доступ на чтение и редактирование
* Если учитель не является владельцем ни одного из тестов - он имеет доступ только на чтение

Нажав на троеточие в правом верхнем углу окна, пользователь может выйти из аккаунта и программы (Рис. 11.3)

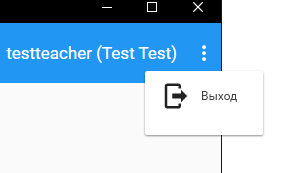


Рис 11.3

Нажав на кнопку “Управление” у одного из тестов, пользователь переходит в окно с информацией о тесте, из которого можно просмотреть информацию о нём, изменить тест или удалить его (но только если пользователь имеет соответствующие права) (Рис. 11.4)

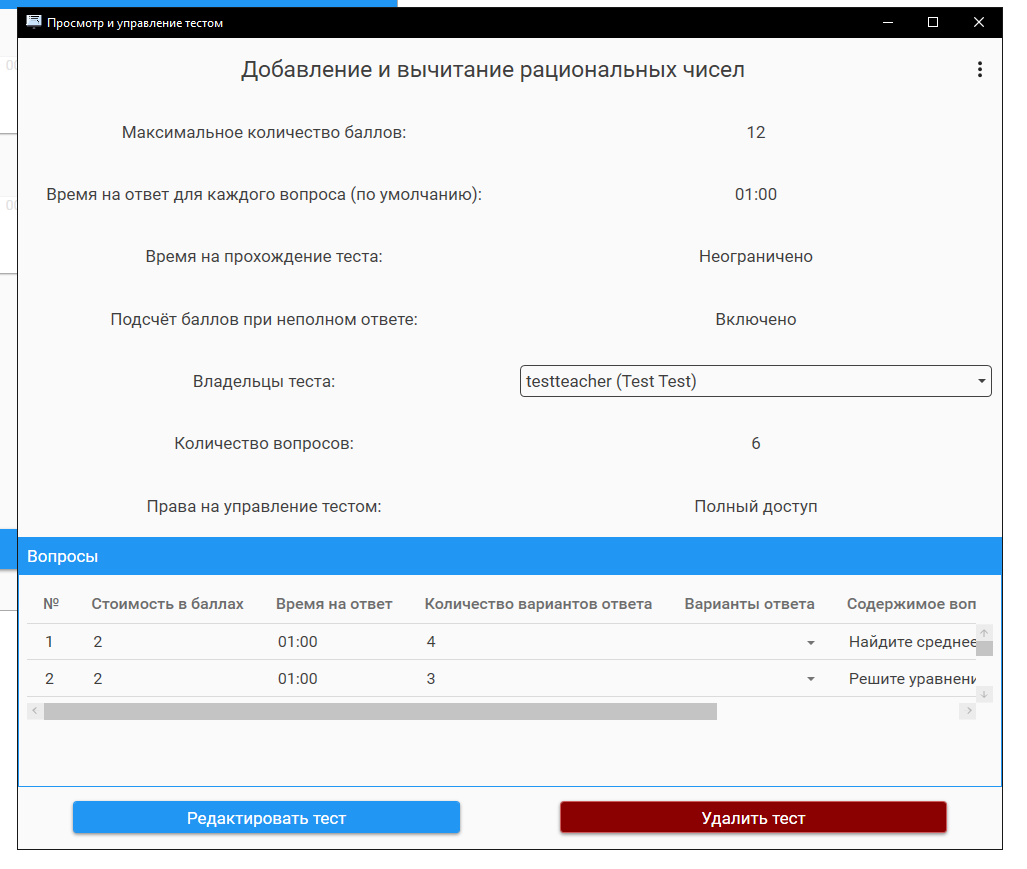


Рис. 11.4

Права на тест определяются по следующим условиям:

* Если учитель является владельцем теста - он имеет полный доступ
* Если учитель не является владельцем теста - он имеет доступ только на чтение

В этом же окне пользователь, нажав на троеточие в правом верхнем углу окна, может сохранить тест в текстовый файл, или сохранить тест в текстовый файл включая правильные ответы (но только если имеет полный доступ к тесту) (Рис. 11.5)

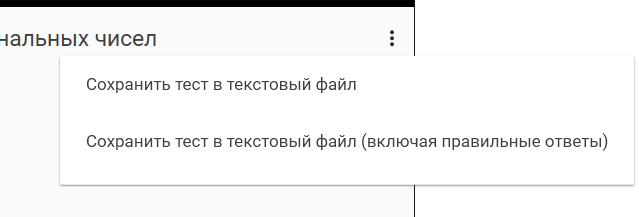


Рис. 11.5

Нажав на круглую кнопку “+”, пользователь может добавить тест или категорию (Рис. 12.1)

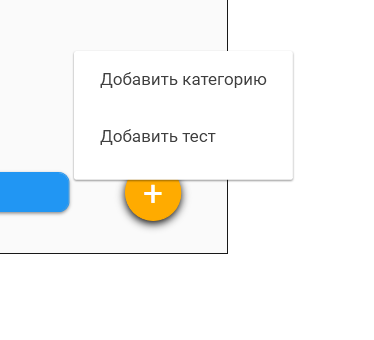


Рис. 12.1

Нажав на кнопку “Добавить категорию”, пользователь попадает в окно редактирования новосозданной категории (Рис. 12.2)

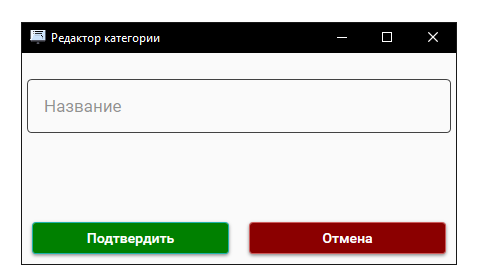


Рис. 12.2

Нажав на кнопку “Добавить тест”, пользователь попадает в окно редактирования новосозданного теста (Рис. 12.3).

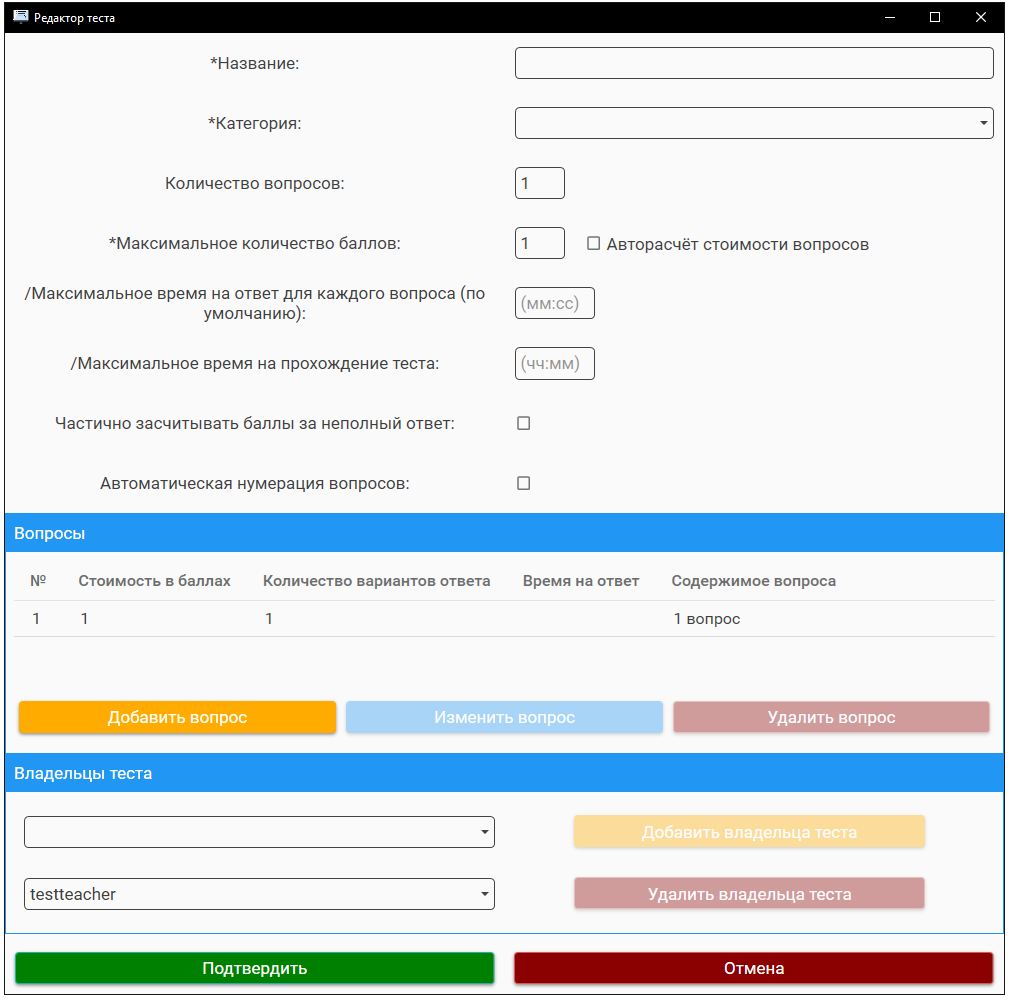


Рис. 12.3

Обозначения возле названия полей

* \* - поле обязательно для заполнения
* ничего - опциональное поле (необязательно для заполнения)
* / - исключающее опциональное поле - то же опциональное поле, только при этом рядом существует также второе поле с таким же обозначением, и если одно из них заполняется - второе автоматически становится не заданным

Отсюда же пользователь может как редактировать вопросы поверхностно (используя поля таблицы; Рис. 12.4), так и полноценно, выбрав конкретный вопрос и нажав на кнопку “Изменить вопрос”, после чего он попадает в отдельное окно редактора вопроса (Рис 12.5)

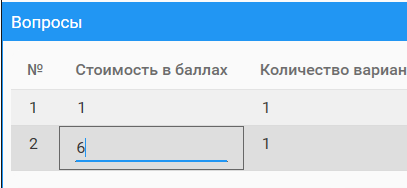


Рис. 12.4

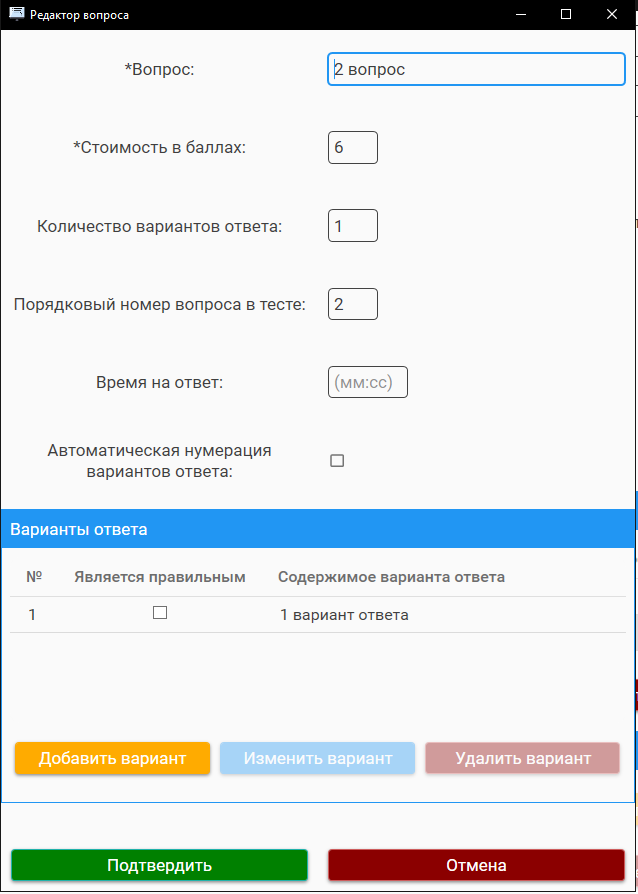


Рис. 12.5

По такому же принципу пользователь может редактировать и варианты ответа внутри вопросов.

Также на главном экране можно найти тест используя строку поиска (Рис. 12.6), и боковое меню навигации по нажатию на три полоски (Рис. 12.7), откуда можно попасть в окно просмотра статистики (Рис. 12.8), и окно информации о программе.

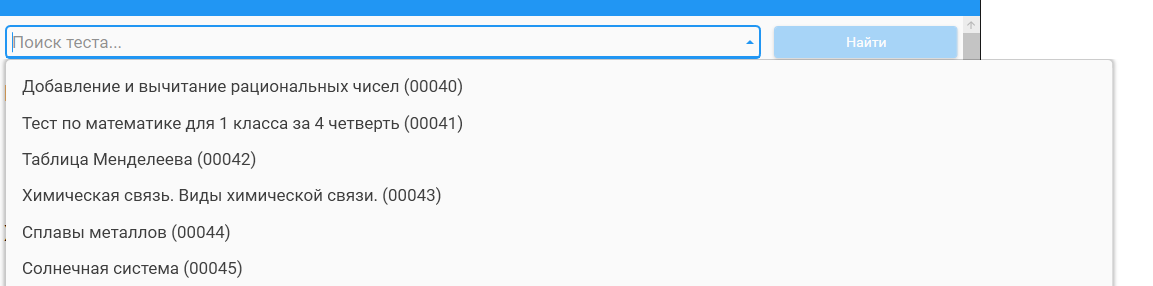


Рис. 12.6

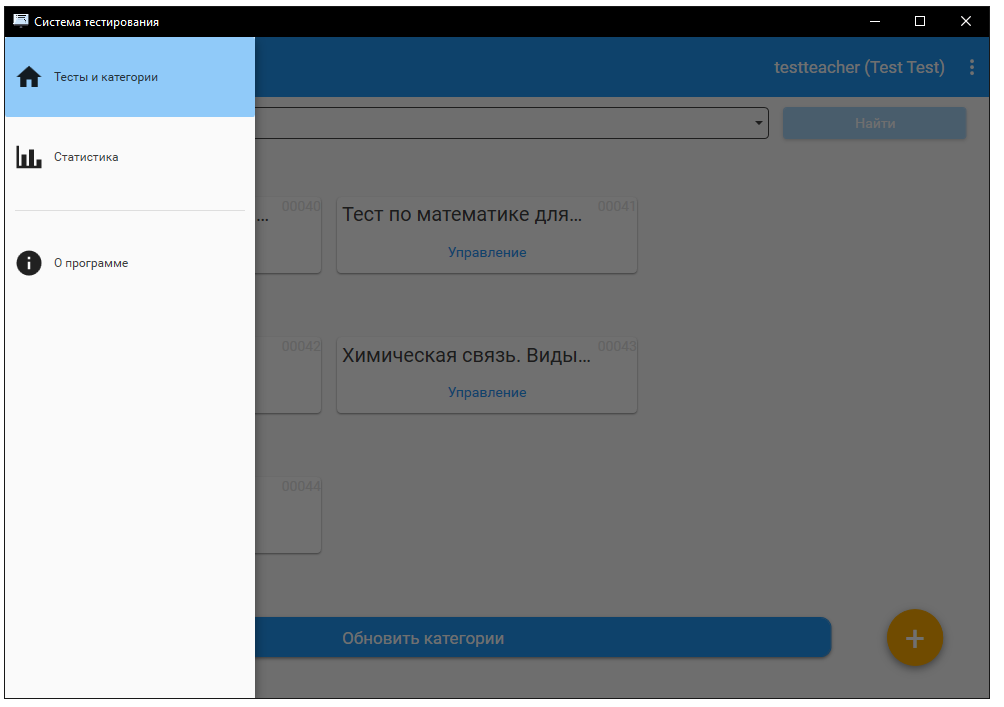


Рис. 12.7

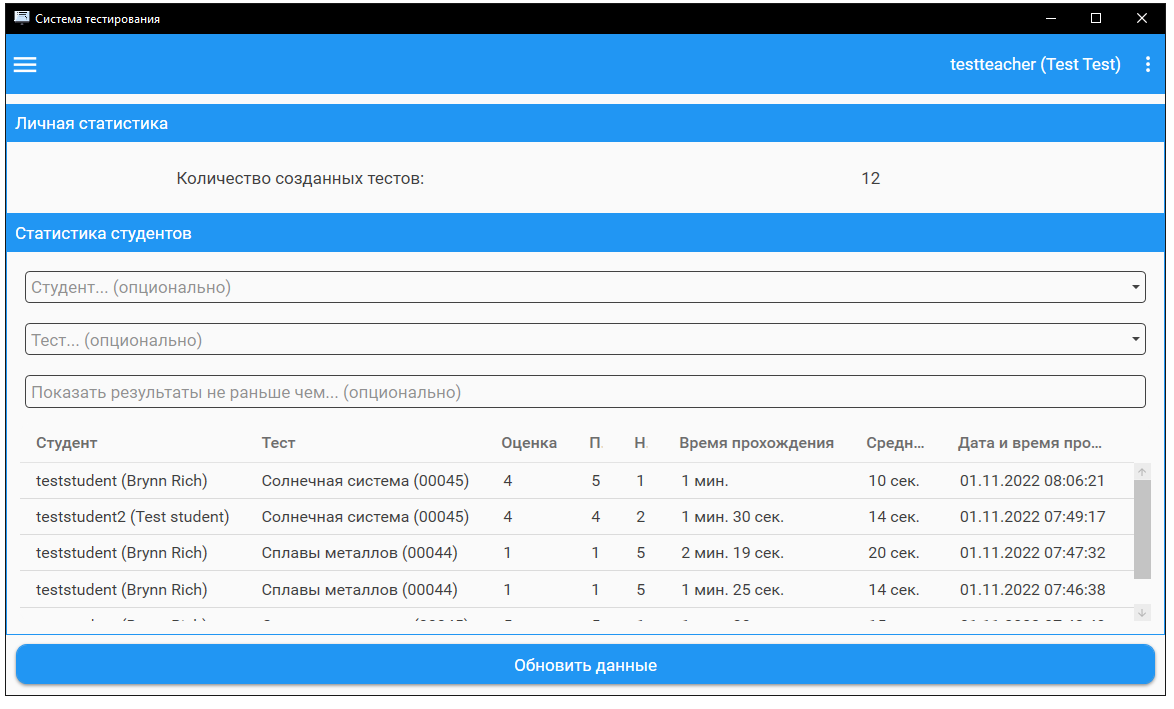


Рис. 12.8

Здесь можно просмотреть личную статистику, всю статистику по всем тестам и студентам, а также по желанию отфильтровать её по дате и/или конкретному студенту и/или тесту.

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ: СТУДЕНТ

После успешного входа или регистрации, открывается главное окно программы (Рис. 13.1)

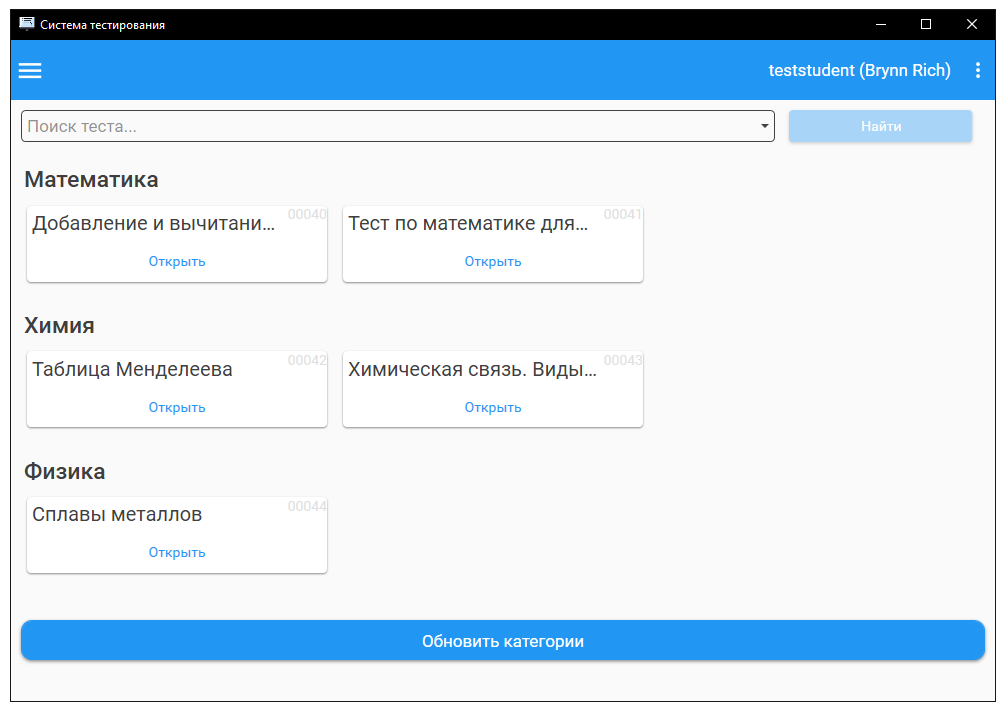


Рис. 13.1

Нажав на троеточие в правом верхнем углу экрана, пользователь может выйти из аккаунта и программы (Рис. 13.2)

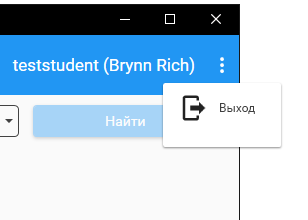


Рис 13.2

Нажав на кнопку “Открыть” у одного из тестов, пользователь переходит в окно с информацией о тесте, из которого далее можно перейти в прохождение теста нажав на кнопку “Пройти тест” (Рис. 13.3)

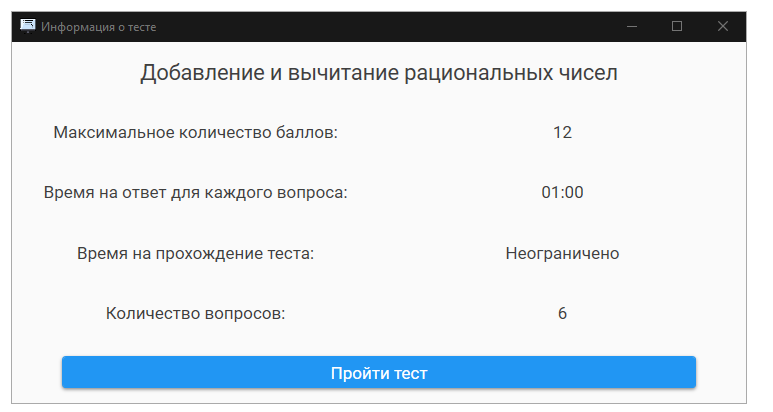


Рис. 13.3

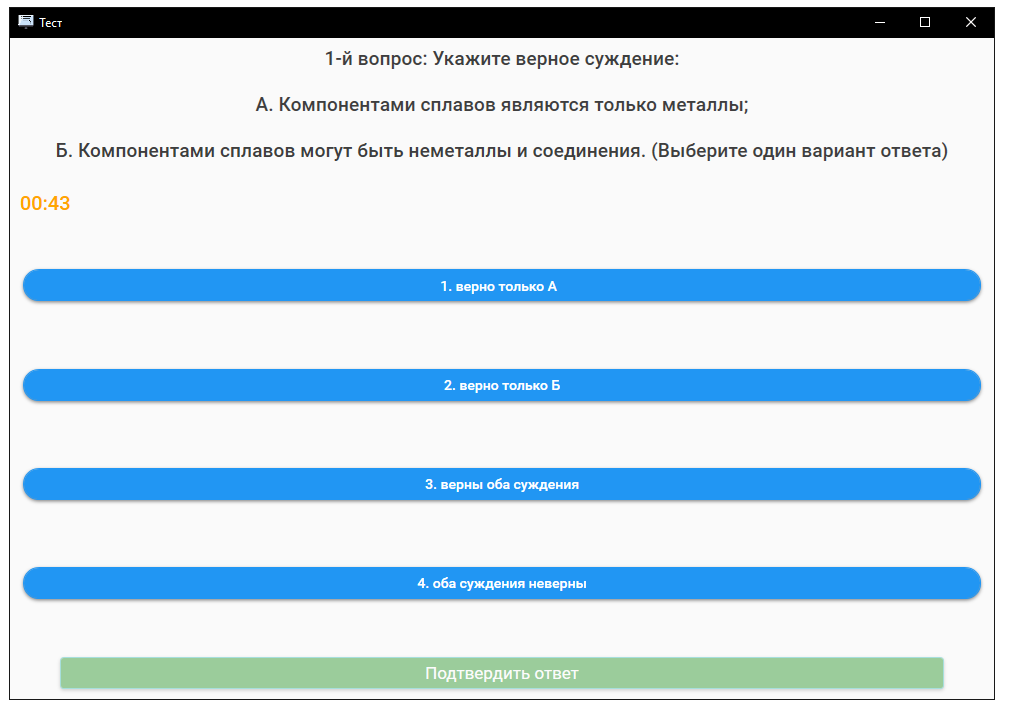


Рис. 13.4

При прохождении теста пользователю нужно выбрать как минимум один вариант ответа (Рис. 13.5). При этом можно ориентироваться на указание в скобках в конце вопроса, в котором отображается один ли правильный ответ в вопросе или несколько.

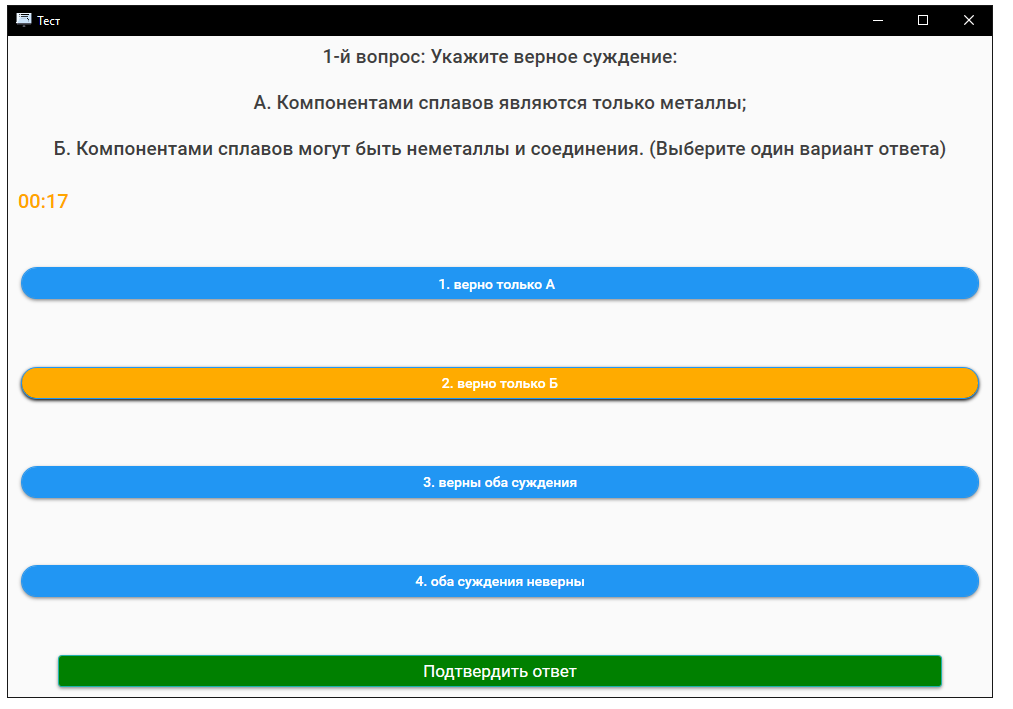


Рис. 13.5

Если вопрос содержит ограничение по времени, оставшееся время можно найти вверху слева под вопросом (Рис. 13.6). По истечении времени вопрос автоматически засчитывается как неправильный и происходит переход на следующий вопрос (или к результатам, если вопрос был последним).

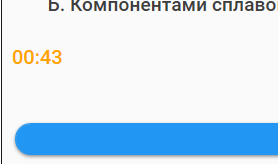
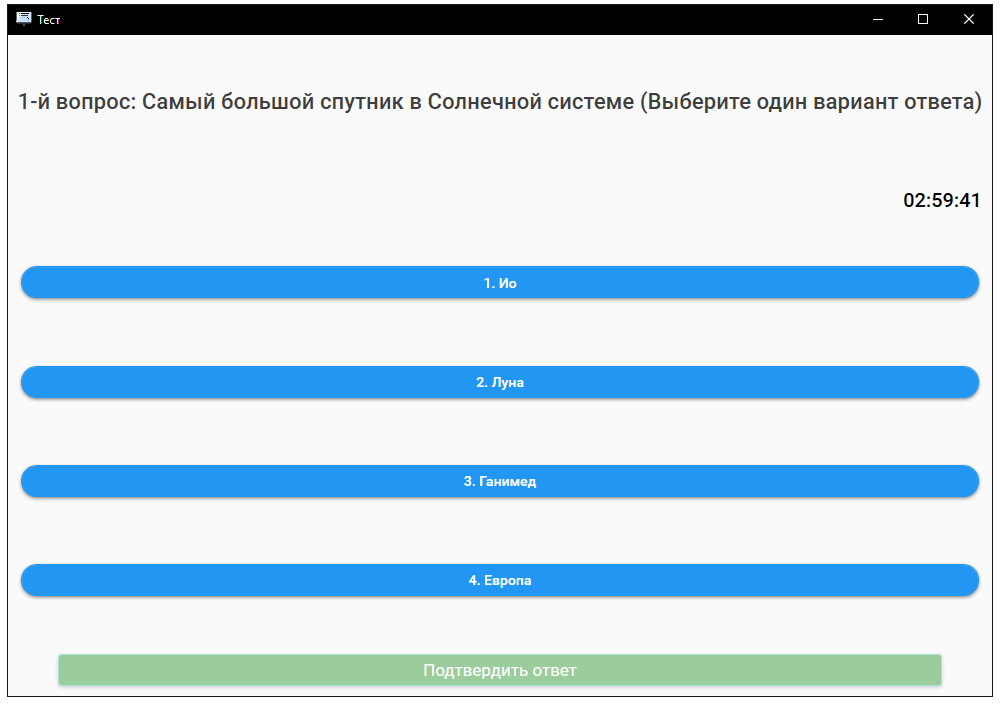


Рис. 13.6

Если тест содержит ограничение по времени, оставшееся время можно найти вверху справа под вопросом (Рис. 13.7). По истечении времени все вопросы, на которые не был дан ответ, автоматически засчитываются как неправильные, и происходит переход к результатам (Рис. 13.8).



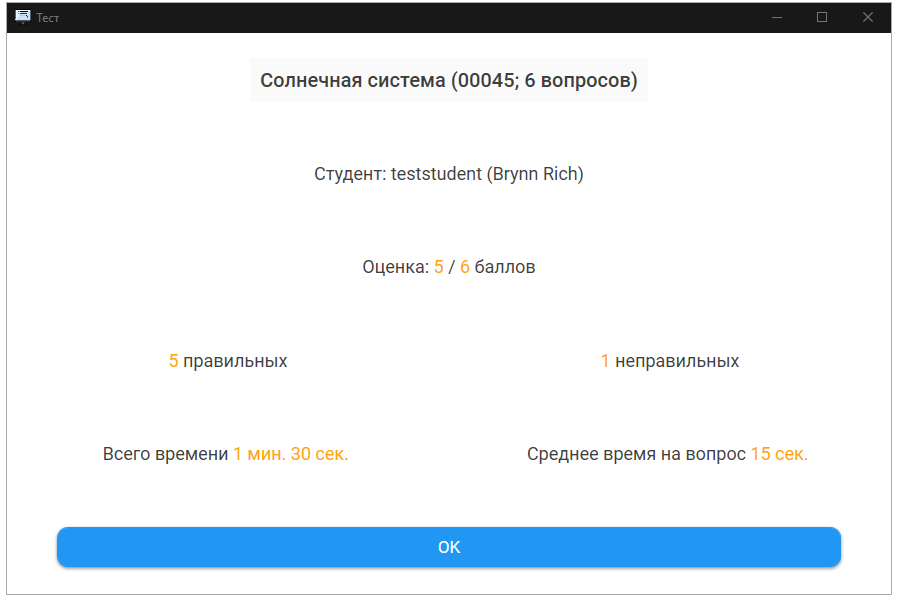


Рис. 13.8

Также на главном экране можно найти тест используя строку поиска (Рис. 13.9), и боковое меню навигации по нажатию на три полоски (Рис. 14.1), откуда можно попасть в окно просмотра статистики (Рис. 14.2), и окно информации о программе (Рис. 14.3).

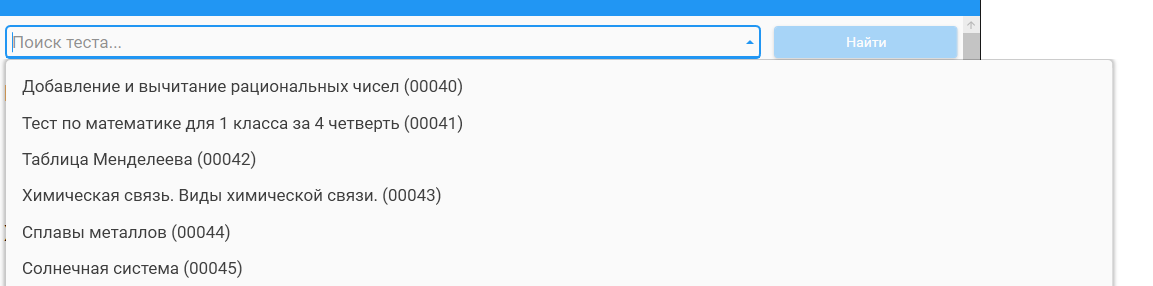


Рис. 13.9

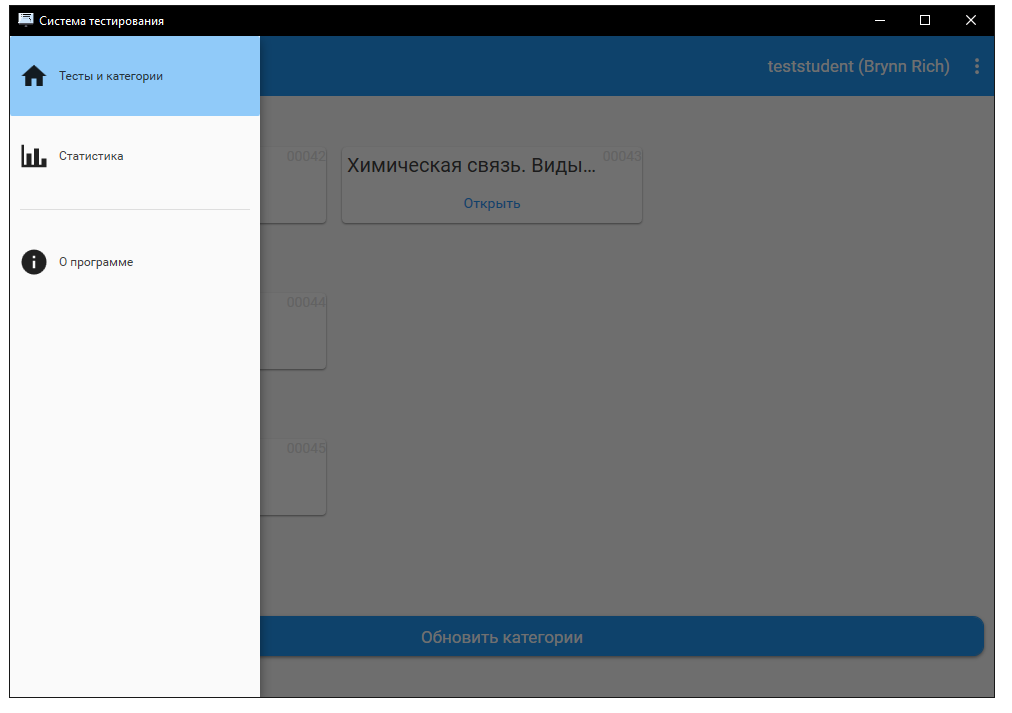


Рис. 14.1

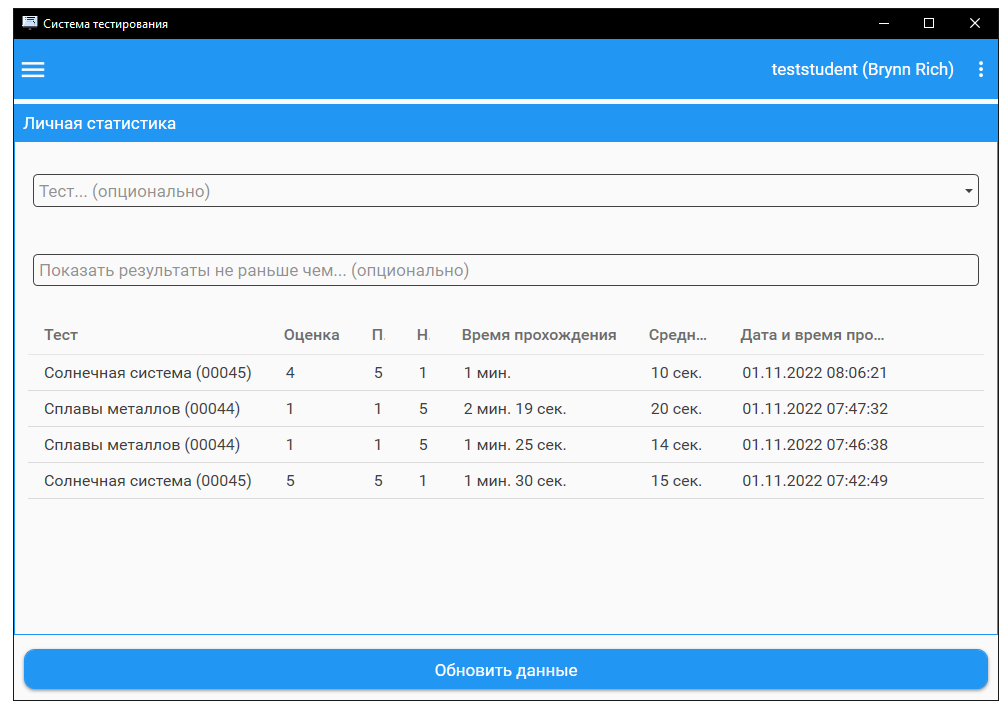


Рис. 14.2

Здесь можно просмотреть личную статистику, а также по желанию отфильтровать её по дате и/или конкретному тесту.

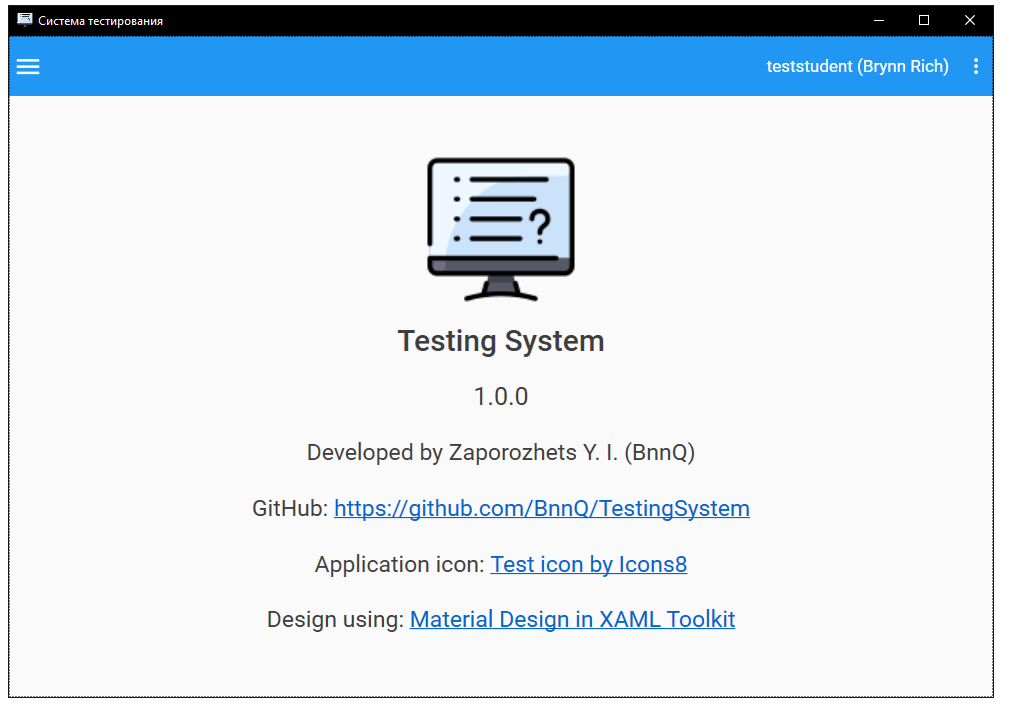


Рис. 14.3

# СПИСОК ЛИТЕРАТУРЫ

1. [Google's Material Design 3](https://m3.material.io)
2. [MaterialDesignInXaml Wiki](https://github.com/MaterialDesignInXAML/MaterialDesignInXamlToolkit/wiki)
3. [Entity Framework documentation](https://learn.microsoft.com/en-us/ef/)
4. [MVVM navigation library by Egor92 documentation](https://github.com/Egor92/MvvmNavigation#mvvmnavigation)
5. [Reactive Validation in WPF with MVVM library by Karnah documentation](https://github.com/Karnah/ReactiveValidation#reactivevalidation)
6. [Article "What’s The Best Algorithm For User Credential Storage In 2022?"](https://infosecscout.com/best-algorithm-password-storage/)
7. [Library implementing Scrypt hashing in .NET documentation](https://github.com/viniciuschiele/scrypt#scryptnet)
8. [Meziantou's Thread-Safe Observable Collection for WPF implementation](https://www.meziantou.net/thread-safe-observable-collection-in-dotnet.htm)
9. [Peoky's BusyIndicator library documentation](https://github.com/Peoky/BusyIndicator#busyindicator)

Исходный код проекта находится здесь:

<https://github.com/BnnQ/TestingSystem>