Министерство образования и молодежной политики Свердловской области

государственное автономное профессиональное образовательное учреждение

Свердловской области «Алапаевский многопрофильный техникум»

(ГАПОУ СО «АМТ»)

Специальность 09.02.07

Информационные

системы и

программирование

Группа 403 ИСП

очная форма обучения

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ**

**Разработка программного приложения «Технология подключения базы данных к настольному приложению»**

**КП.09.02.07.403.10.2023.ПЗ**

Разработал: Казанцев Роман Андреевич,

Руководитель проекта: Кондратьева Виктория Валентиновна

Оценка проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Алапаевск 2023

**Содержание**

[**Введение** 3](#_Toc152168436)

[**глава 1. Постановка задачи на разработку программного приложения «Технология подключения базы данных к настольному приложению»** 4](#_Toc152168437)

[**1.1 Техническое задание на разработку программного приложения «Технология подключения базы данных к настольному приложению»** 4](#_Toc152168438)

[**1.2 Обзор существующих решений** 5](#_Toc152168439)

[**1.3 Обзор и обоснование выбора программных средств** 6](#_Toc152168440)

[**Глава 2. проектирование и Разработка программного приложения «Технология подключения базы данных к настольному приложению»** 7](#_Toc152168441)

[**2.1 Описание этапов проектирования и разработки программного приложения** 7](#_Toc152168442)

[**2.2 Тестирование и отладка программного приложения** 8](#_Toc152168443)

[**2.3 Описание программного приложения** 9](#_Toc152168444)

[**Заключение** 10](#_Toc152168445)

[**Список использованных источников** 11](#_Toc152168446)

[**Приложение** 12](#_Toc152168447)

**Введение**

**Актуальность** выбранной темы курсового проекта заключается в том, что с ростом объемов данных и распространением цифровых технологий, потребность в подключении настольных приложений к внешним источникам данных, таким как базы данных, становится все более важной. Это позволяет организациям собирать, хранить, обрабатывать и анализировать данные, а также предоставлять своим пользователям опыт работы с информацией.

**Проблема** заключается в том, что студентам трудно понять, как именно подключать базу данных к настольному приложению, а также с помощью каких средств это осуществляется.

**Гипотеза:** Данное программное приложение поможет студентам освоить навык подключения базы данных к настольному приложению разными способами и сэкономит их время на поиск информации.

**Объект исследования:** Программное приложение

**Предмет исследования:** Программное приложение «Технология подключения базы данных к настольному приложению»

**Цель исследования:** Разработка программного приложения «Технология подключения базы данных к настольному приложению»

**Задачи исследования:**

1. Проанализировать предметную область с помощью различных источников информации;
2. Разработать техническое задание на разработку программного приложения (ПП);
3. Проанализировать существующие решения;
4. Сделать и обосновать выбор программных средств;
5. Описать этапы проектирования и разработки ПП;
6. Протестировать программное приложение;
7. Описать программное приложение.

**глава 1. Постановка задачи на разработку программного приложения «Технология подключения базы данных к настольному приложению»**

**1.1 Техническое задание на разработку программного приложения «Технология подключения базы данных к настольному приложению»**

**1. Введение**

Наименование программы: «Технология подключения базы данных к настольному приложению».

Краткая характеристика области применения: учёба

**2. Основания для разработки**

Наименование предприятий заказчика системы: ГАПОУ СО «Алапаевский многопрофильный техникум», юридический адрес — 624601, Свердловская область, город Алапаевск, улица Ленина, д.11, корпус 1; телефон: +7 (343) 462-16-64.

Наименование предприятий разработчика системы: «DictCompany», адрес: Свердловская область, посёлок Заря, улица Авиационная 13; телефон: +7 (912) 271-72-56.

Перечень документов, на основании которых создается система: техническое задание.

Полное наименование программы: «Технология подключения базы данных к настольному приложению».

Условное обозначение: ПП «Технология подключения БД к настольному приложению».

**3. Назначение разработки**

«Технология подключения базы данных к настольному приложению» - программа, выполненная в виде электронного справочника. Эта программа предназначена для хранения справочных материалов по технологиям подключения БД к приложению, а также для тестирования пользователей по этим справочным материалам.

**4. Требования к программе**

*Требования к функциональным характеристикам:*

ПП «Технология подключения базы данных к настольному приложению» должен иметь следующие функции: просмотр справочных материалов, проверка знаний с помощью теста.

*Требования к надежности:*

Состав и количественные значения показателей надежности для системы в целом или ее подсистем:

* Применение соответствующего программного обеспечения и технических средств;
* Соблюдение правил эксплуатации и технического обслуживания программно-аппаратных средств.

Перечень аварийных ситуаций, по которым должны быть регламентированы требования к надежности, и значения соответствующих показателей:

* Сбой рабочей станции пользователя;
* Сбой информационной системы.

Требования к надежности технических средств и программного обеспечения:

* Должны использоваться средства с высокой надёжностью, выполняющие соответствующие задачи для работы;
* Должно использоваться необходимое программное обеспечение последних или стабильных версий.

*Условия эксплуатации:*

Рабочее место должно находится в оборудованном кабинете либо помещении.

Все внешние элементы технических средств системы, находящиеся под напряжением, должны иметь защиту от случайного прикосновения.

Для эксплуатации понадобиться пользователь системы.

Основными возможности пользователя ИС являются просмотр справочных материалов, а также прохождение теста по этим материалам.

Пользователи системы должны иметь опыт работы с персональным компьютером на базе операционной системы Microsoft Windows 10 или выше на уровне квалифицированного пользователя и свободно осуществлять базовые операции в стандартных программах Windows.

Рекомендуемая численность для эксплуатации системы:

* + Пользователь – число штатных единиц определяется структурой предприятия.

*Требования к составу и параметрам технических средств:*

* Процессор: 32- или 64-разрядный процессор с тактовой частотой 1 ГГц или выше;
* Оперативная память: 2 ГБ;
* Свободное место на жестком диске: 200 МБ свободного места на диске;
* Монитор: Разрешение 1280 x 800;
* Клавиатура, мышь.

*Требования к информационной и программной совместимости:*

Состав, структура в программе: программа должна иметь подсистему с теоретическими сведениями и подсистему с тестом.

*Требования к лингвистическому обеспечению АС:*

Для разработки программы должны применяться система для построения клиентских приложений – WPF и язык программирования высокого уровня – С# с языком разметки XAML.

Языки взаимодействия пользователей и технических средств системы: русский язык.

Требования к кодированию и декодированию данных: UTF-8.

Для организации диалога системы с пользователем должен применяться графический пользовательский интерфейс в виде страниц и окон.

Программа должна запускаться на ОС Windows 10 и выше.

*Требования к маркировке и упаковке:*

Требования не предъявляются.

*Требования к транспортированию и хранению:*

Требования не предъявляются.

**5. Требования к программной документации**

Руководство оператора, описание программы.

**6. Стадии и этапы разработки**

|  |  |
| --- | --- |
| **Стадия работ** | **Срок выполнения** |
| 1. Анализ предметной области | 6.10.2023 – 24.10.2023 |
| 2. Разработка технического задания | 23.10.2023 – 15.11.2023 |
| 3. Проектирование и разработка программного приложения | 24.10.2023 – 18.12.2023 |
| 3.1 Проектирование программного приложения | 24.10.2023 – 15.11.2023 |
| 3.2 Разработка программного приложения | 4.11.2023 – 15.12.2023 |
| 3.3 Описание контрольного примера | 15.12.2023 – 18.12.2023 |
| 4. Ввод в действие | 19.12.2023– 25.12.2023 |
| 5. Сопровождение ИС | 25.12.2023- 25.01.2024 |

**7. Порядок контроля и приёмки**

Виды, состав, объем, и методы испытаний программы должны быть изложены в программе и методике испытаний, разрабатываемой в составе рабочей документации.

Общие требования к приемке работ по стадиям, порядок согласования и утверждения приемочной документации: сдача-приёмка работ производится поэтапно, в соответствии с рабочей программой и календарным планом.

**1.2 Обзор существующих решений**

Чтобы приступить к разработке своего программного приложения, было принято решение посмотреть существующие решения и оценить их возможности.

Первый сайт, который был проанализирован, это *Доступ к данным в приложениях Windows [1]* - официальная документация Microsoft, описывающая как использовать различные базы данных в своих проектах.

Ознакомившись с данной документацией, можно выделить следующие её преимущества, а именно:

* Примеры программного кода и его подробное объяснение;
* Описано использование целых пяти различных СУБД.

Но в данном справочном материале были выделены и минусы:

* Некоторые из статей слишком коротки или сложны. А это ограничивает людей, которые только начали изучать данную тему;
* Необходим доступ к Интернету для прочтения документации.

Технология доступа к данным ADO.NET [2] – файл на сайте StudFiles, который рассказывает о технологии ADO.NET и описывает работу с этой технологией.

После изучения данного файла, были выделены такие плюсы как:

* Подробное объяснение работы всех классов, объектов, свойств и методов. Это очень важно для новичков, только начавших узнавать, как связать свою программу с базой данных;
* Данный файл можно скачать и пользоваться им без Интернета.

Один минус, который стал выделяться после прочтения документации Майкрософт, так это то, что в данном файле делается упор на один способ подключения базы данных. Также там затрагивается тема работы с различными СУБД, но она не была сильно раскрыта.

**1.3 Обзор и обоснование выбора программных средств для разработки программного приложения «Технология подключения базы данных к настольному приложению»**

Для разработки desktop-приложения необходимо выбрать подходящую для себя среду разработки, определиться с языком программирования и инструментом.

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки (IDE) программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать все виды приложений.

Visual Studio поддерживает множество языков программирования, а именно: Ajax, ASP.NET, DHTML, ASP.NET, JavaScript, Visual Basic, Visual C#, Visual C++, Visual F#, XAML и другие.

Преимущества Visual Studio:

* Возможность создать любое приложение.
* Поддержка большого количества языков программирования
* Имеет множество инструментов для разработчика
* Есть бесплатная версия (Community) для частного использования, студентов и создателей опенсорсовых проектов.

Недостаток Visual Studio является то, что эта среда не кроссплатформенная и доступна только на операционной системе Windows.

Visual Studio Code - лёгкий, но мощный редактор исходного кода, который работает на вашем настольном компьютере и доступен для Windows, macOS и Linux.

Он включает встроенную поддержку JavaScript, TypeScript и Node.js и имеет богатую экосистему расширений для других языков и сред выполнения (например, C++, C#, Java, Python, PHP, Go, .NET).

Преимущества Visual Studio Code:

* Огромный набор плагинов, с помощью которого можно настроить редактор на свой вкус
* Поддержка большого количества языков программирования
* Кроссплатформенность
* Полностью бесплатен

Но насколько мощным Visual Studio Code не казался, он больше подходит для разработки веб-приложений и сайтов. Если возникнет желание создать desktop-приложение, то могут возникнут трудности по настройке редактора.

CLion - интегрированная среда разработки (ИСР) для языков программирования Си и C++, разрабатываемая компанией JetBrains. Подходит для операционных систем Windows, macOS и Linux.

Преимущества CLion:

* Поддержка многих языков программирования
* Кроссплатформенность

Недостатком CLion является то, что эта интегрированная среда разработки платная.

Из рассмотренных сред разработки была выбрана Visual Studio. Выбрана она по следующим причинам:

1. Немалый выбор инструментов для создания desktop-приложений.
2. Бóльший опыт работы с этой мощной средой разработки.

Определившись со средой разработки, осталось выбрать инструменты, с помощью которых будет создаваться приложение.

Electron - фреймворк, разработанный GitHub. Позволяет разрабатывать нативные графические приложения для операционных систем с помощью веб-технологий, комбинируя возможности Node.js для работы с back-end и браузера Chromium.

Плюсы Electron:

* Использование наработок из Web (HTML, CSS, JS)
* Кроссплатформенность
* Качественная (и русифицированная) документация
* Поддержка сообщества и GitHub

Минусы Electron:

* Высокое потребление памяти (физическая и ОЗУ)
* Легко написать плохой код
* Плохая нативность

Qt – библиотека классов C++ и набор инструментального программного обеспечения для создания кросс-платформенных приложений с графическим интерфейсом (GUI).

Существуют вариации для других языков:

* PyQt для Python,
* QtRuby для Ruby,
* Qt Jambi для Java.

Плюсы Qt:

* Визуальный редактор интерфейса
* Огромное количество модулей в составе фреймворка
* Поддержка большого количества языков программирования
* Кроссфплатформенность

Минусы Qt:

* Большой вес приложения
* Высокая стоимость коммерческой лицензии

Windows Presentation Foundation (WPF) — это платформа разработки графических интерфейсов от Microsoft, которая предлагает богатый набор инструментов для создания красивых и функциональных пользовательских интерфейсов.

Плюсы WPF:

* Широкая интеграция. WPF поддерживает ЗD-графику, видео, речевые технологии, работу с документами.
* Декларативное программирование. Конструктор доступен, но разработчики обычно предпочитают создавать проекты на основе кода с помощью декларативного XAML.
* Богатые возможности композиции и настройки.

Минусы WPF:

* Отсутствие кроссплатформенности. Созданные приложения будут работать только на ОС Windows.

В качестве инструмента разработки была выбрана платформа WPF по ряду причин:

1. Разделение пользовательского интерфейса и функционального программного кода в лице языка разметки XAML и языка программирования C#.
2. С помощью XAML можно сверстать любой пользовательский интерфейс любой сложности.
3. Встроенная поддержка паттерна разработки MVVM.
4. В отличие от других платформ, с WPF имеется больше опыта работы.

И для хранения данных использовалась база данных SQLite. SQLite - компактная встраиваемая СУБД. Основной причиной выбора именно этой СУБД является то, что она не требует сервера баз данных и может быть интегрирована непосредственно в приложение.

**Глава 2. проектирование и Разработка программного приложения «Технология подключения базы данных к настольному приложению»**

**2.1 Описание этапов проектирования и разработки программного приложения**

**Обзор различных способов и форм взаимодействия пользователя с системой.**

Взаимодействие пользователя с системой может происходить различными способами. Сейчас будут рассмотрены самые распространённые их них.

Текстовый интерфейс пользователя (CLI) — это тип пользовательского интерфейса, в котором команды и данные вводятся через командную строку. CLI часто используется в операционных системах и некоторых приложениях для технических задач.

Графический интерфейс пользователя (GUI) — это наиболее распространенный тип интерфейса в IT-системах. В интерфейсе применяются графические элементы (кнопки, поля ввода, поля выбора и др.), что облегчает взаимодействие пользователя с системой.

Жестовый интерфейс (GBI, Gesture-Based Interface или GDI — Gesture Driven). Это технология взаимодействия, с помощью которой можно управлять устройствами, которые «понимают» жесты (движения тела). Используется в ноутбуках, смартфонах и планшетах. Бывают двух типов: контактный (тактильный) и бесконтактный.

Голосовой интерфейс (VUI, Voice user interface). Пользователь вводит голосовую команду на своем родном языке, смартфон распознает ее и выполняет. Практическая реализация голосовых интерфейсов – сложная задача, программисты должны учитывать множество параметров. Пример: программы Алиса у Яндекс, OK Google.

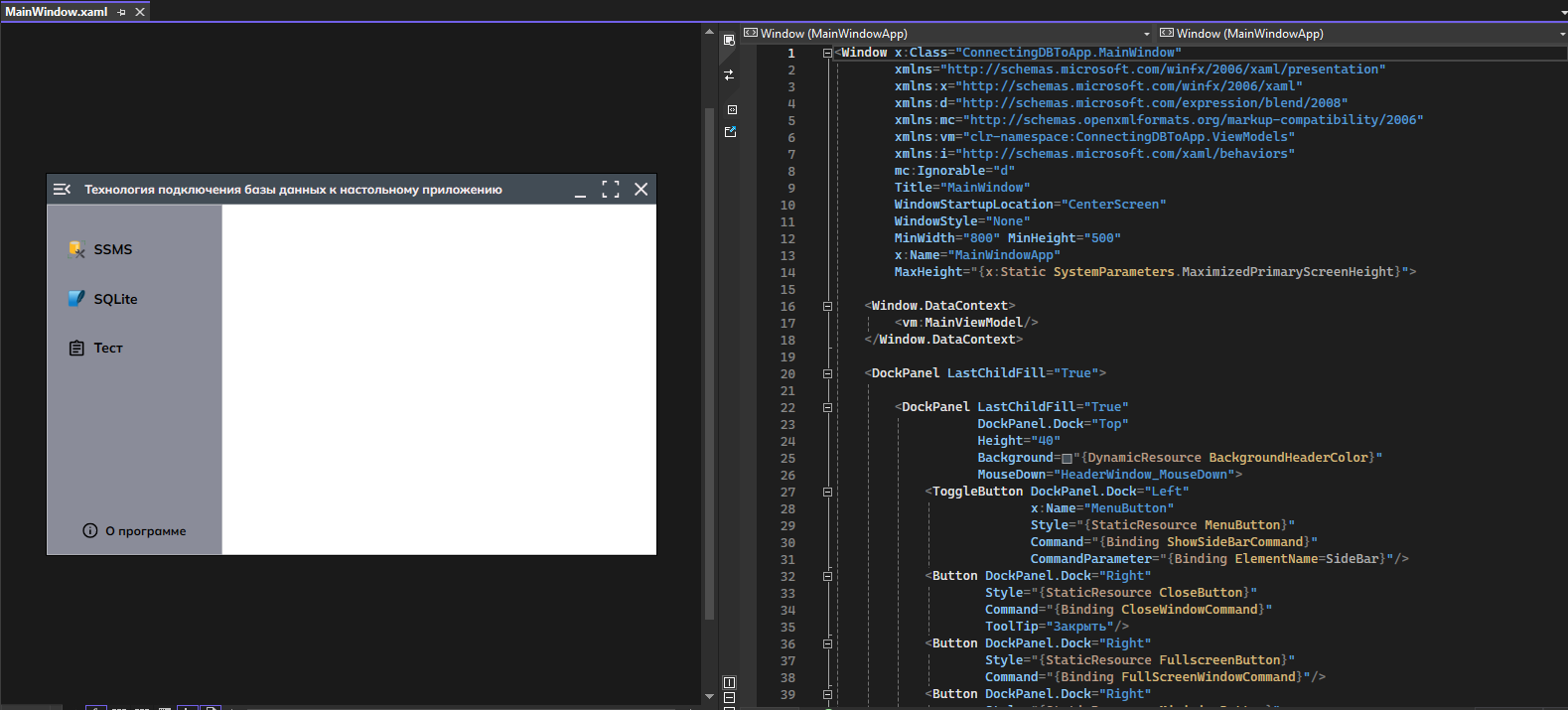
Для общения с пользователем был выбран способ взаимодействия пользователя с системой с помощью графического интерфейса. Этот способ был выбран потому, что подавать и объяснять информацию пользователю так будет более наглядно, доступно и удобно. Это будет достигаться с помощью визуальных примеров, текстовым содержанием.

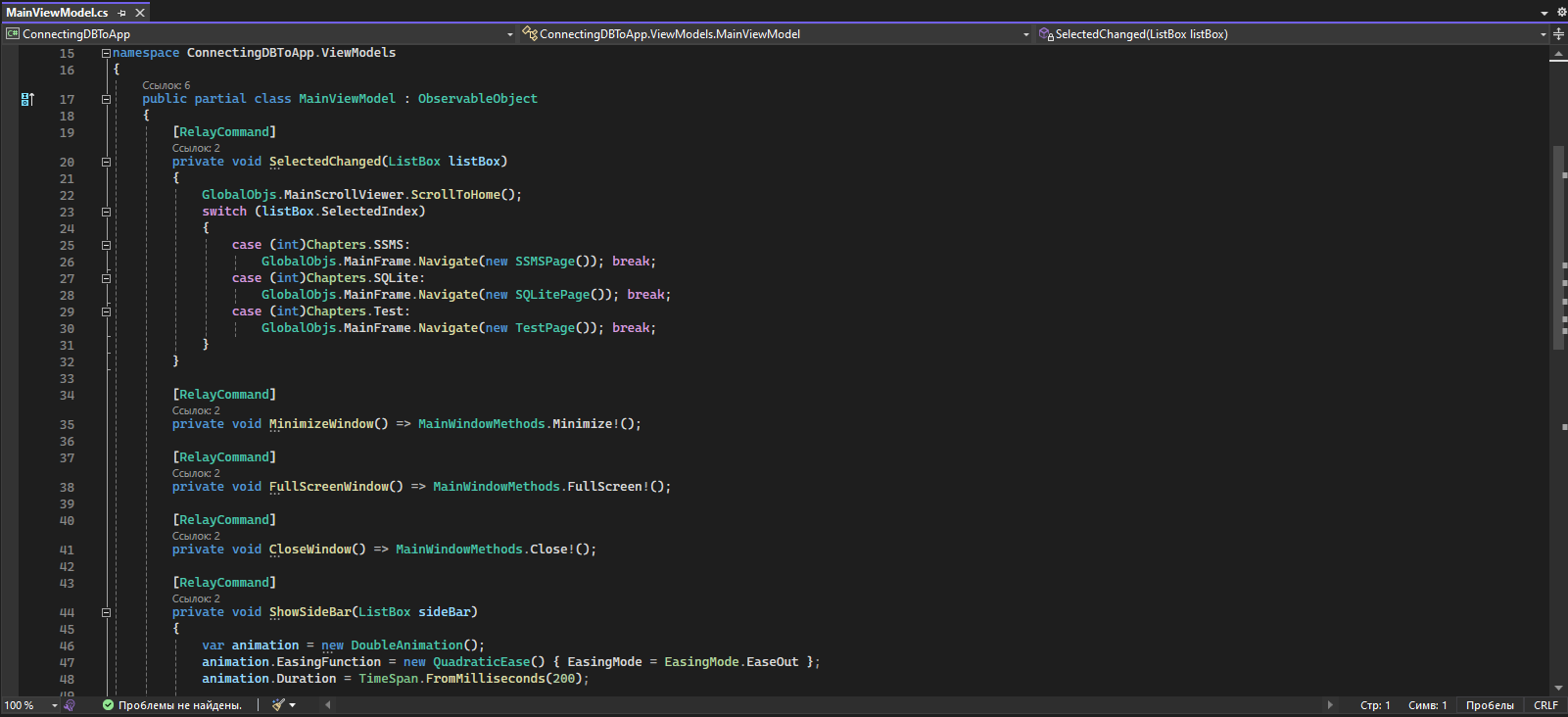
**Разработка интерфейса программного приложения и проектирование программного кода приложения**

*Главное окно приложения* осуществляет навигацию по всему приложению. В левой части окна находится боковая панель приложения, в которой находятся все разделы приложения. В правой части окна непосредственно отображаются страницы этих разделов.

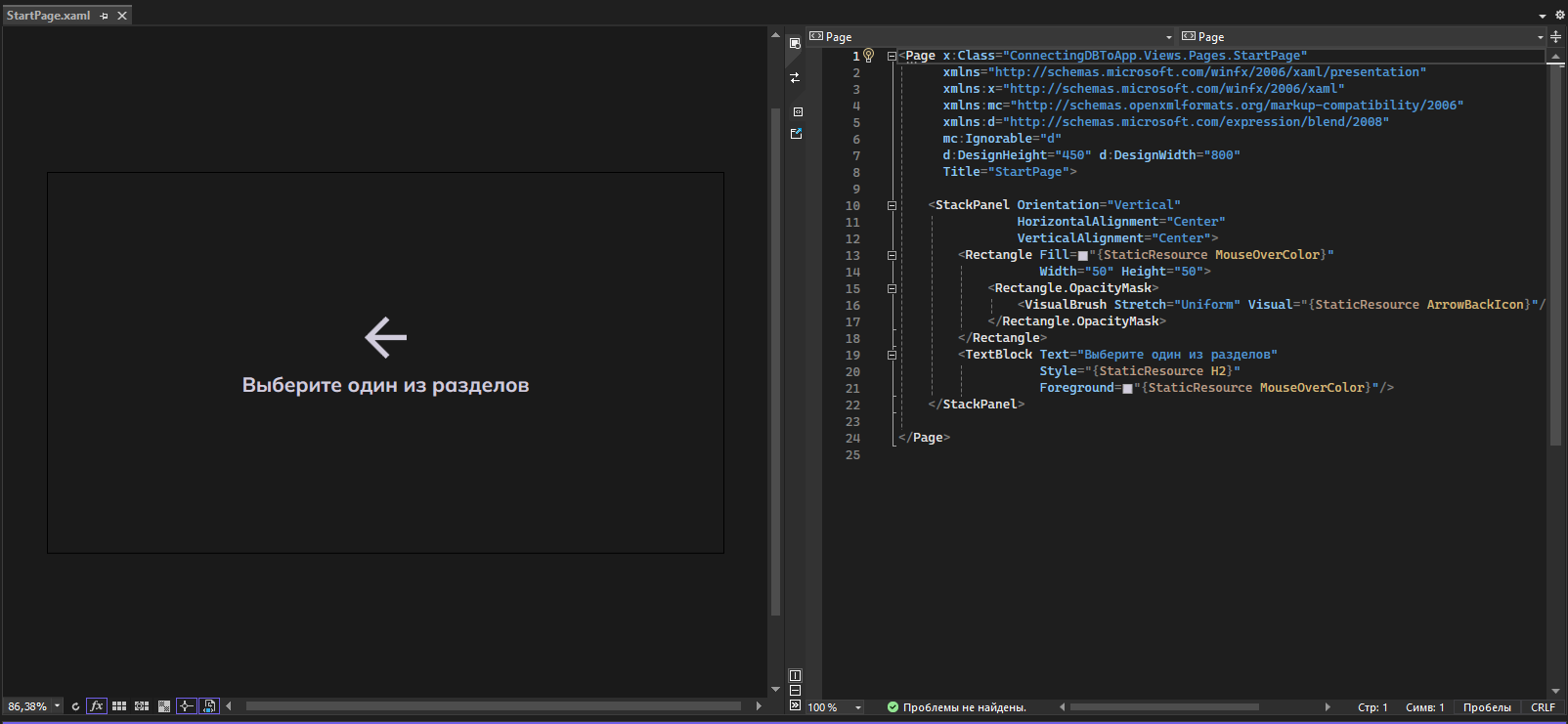
Класс MainViewModel реализуют логику для главного окна. В нём реализованы:

* переходы по разделам
* открытие окна «О программе»
* показ/скрытие меню навигации
* реализация функционала для кастомизированных кнопок «Свернуть», «Полный экран», «Закрыть»

Рисунок 1. Интерфейс и разметка главного окна приложения

 Рисунок 2. Класс с логикой главного окна приложения

*Стартовая страница приложения* – это страница, котораяотображается при запуске программного приложения.

 Рисунок 3. Интерфейс и разметка стартовой страницы

*Страница с теоретическими сведениями о SQL Server Managament Studio* отображается теоретические сведения о работе с такой СУБД как SQL Server Managament Studio. Эта страница выполнена в виде вкладок по которым можно перемещаться. В каждой вкладке отображается часть определённой обучающей информации.

Для этой страницы класс SSMSViewModel реализует получение данных из базы данных, функцию для копирования текста в буфер обмена и функции, реализующие перемещение по разделам с помощью гиперссылок в конце каждого раздела страницы.

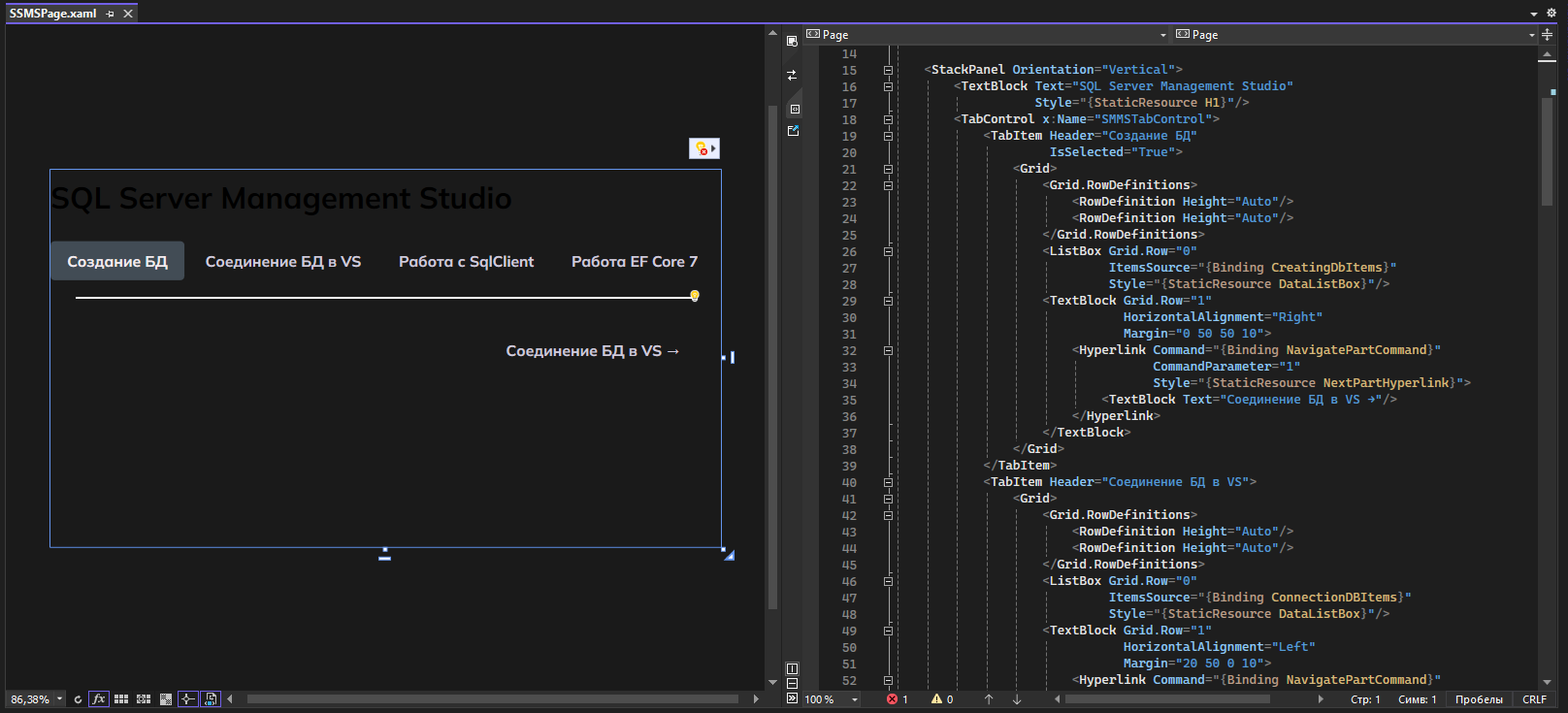
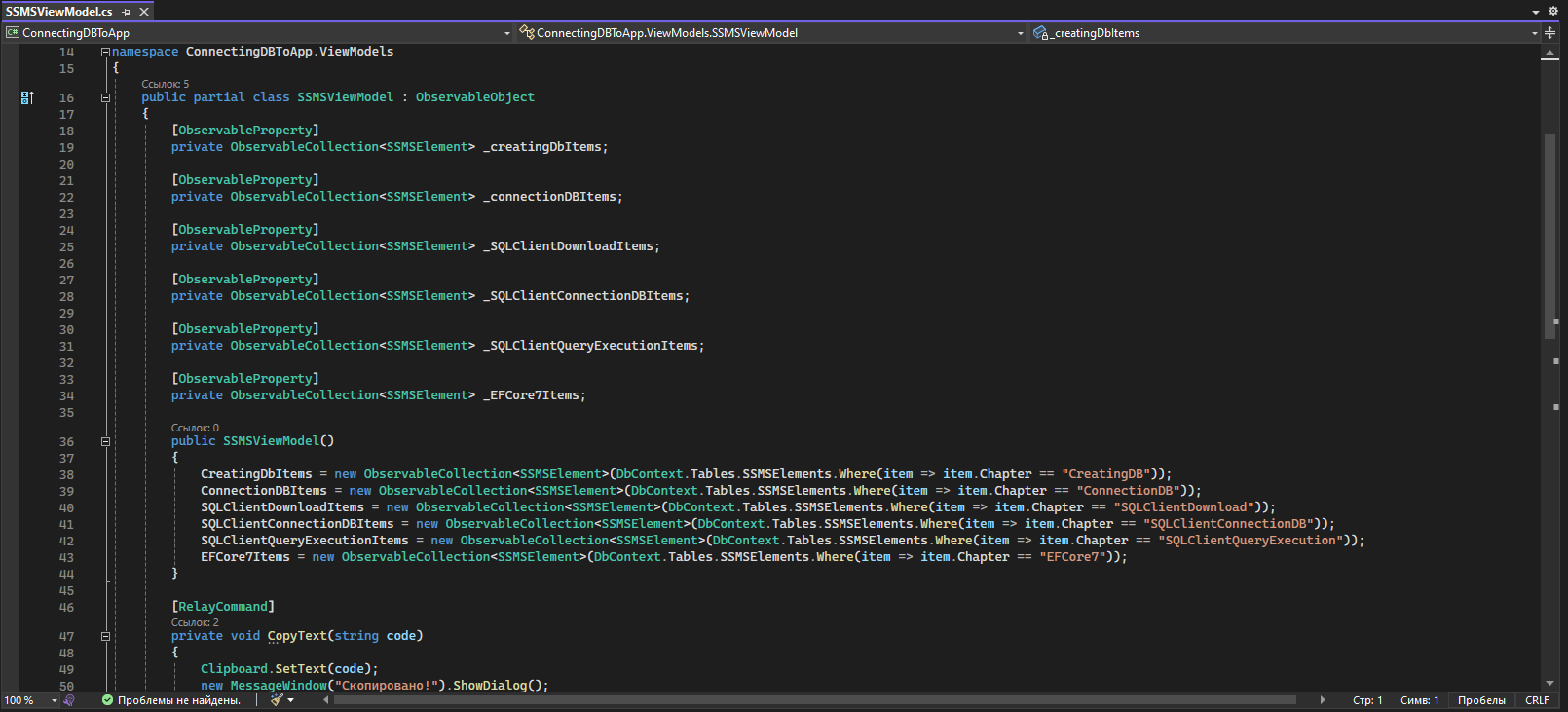
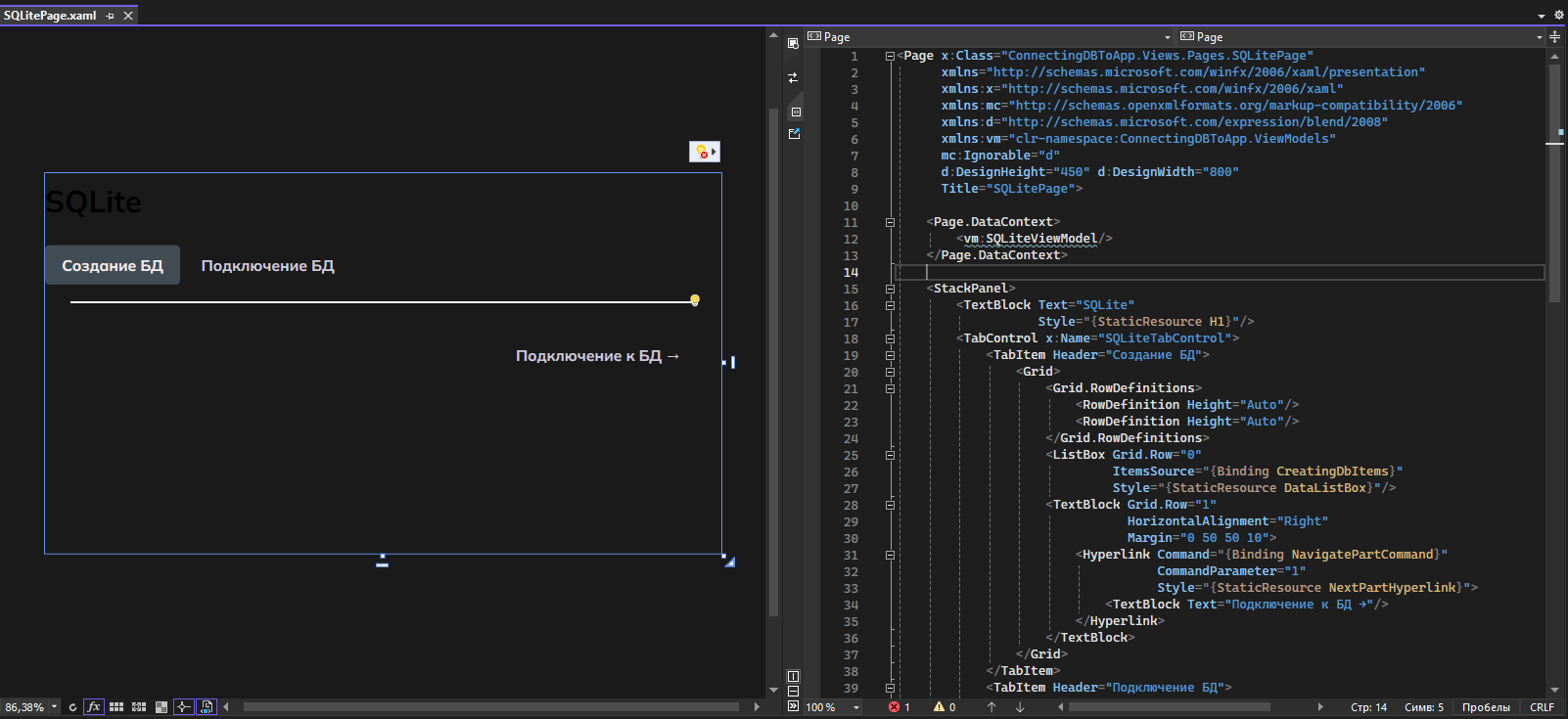


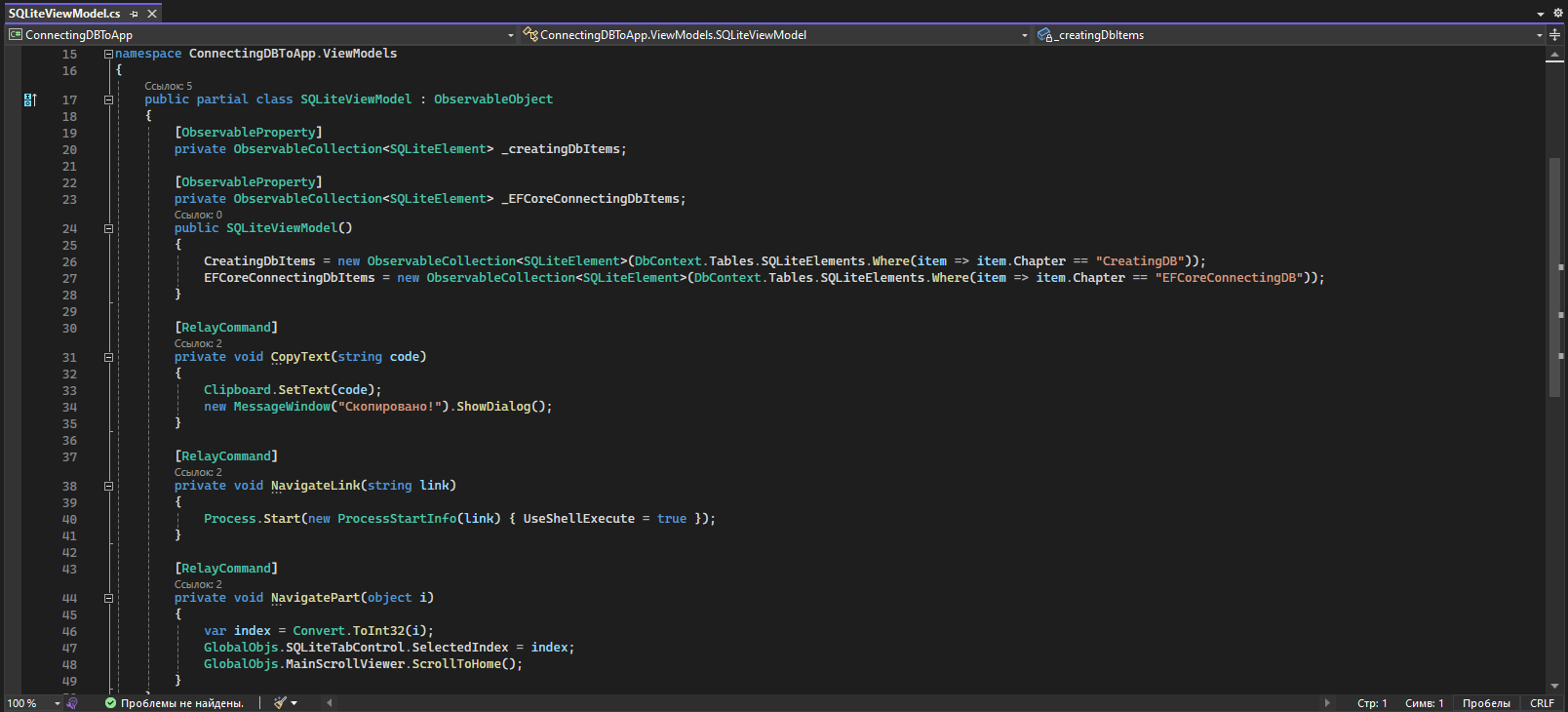
Рисунок 4. Интерфейс и разметка страницы SSMS

Рисунок 5. Класс с логикой страницы SSMS

*Страница с теоретическими сведениями о SQLite* отображается теоретические сведения о работе с такой СУБД как SQLite. Пользовательский интерфейс идентичен странице с SSMS.

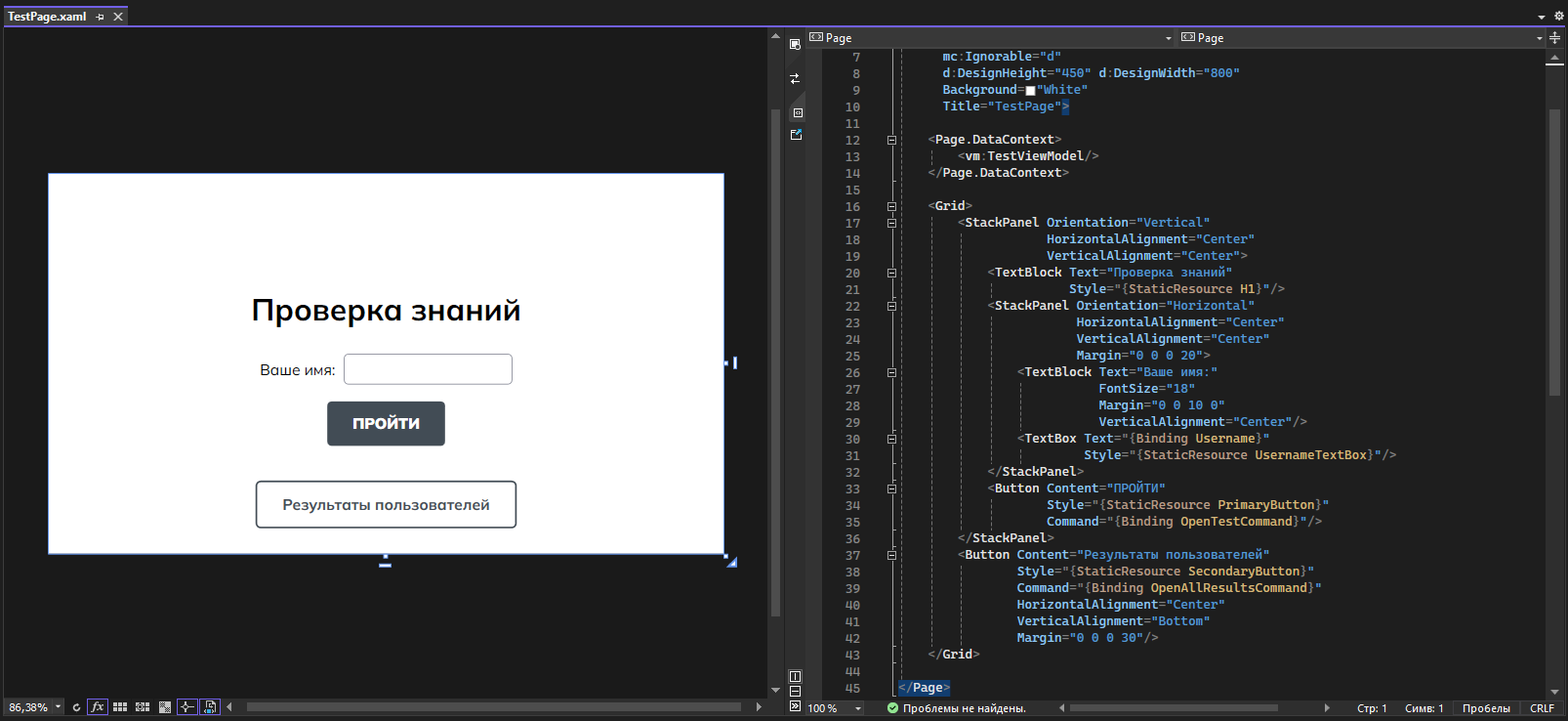
Класс SQLiteViewModel реализует тот же самый функционал, как и класс SSMSViewModel, только для страницы с SQLite.

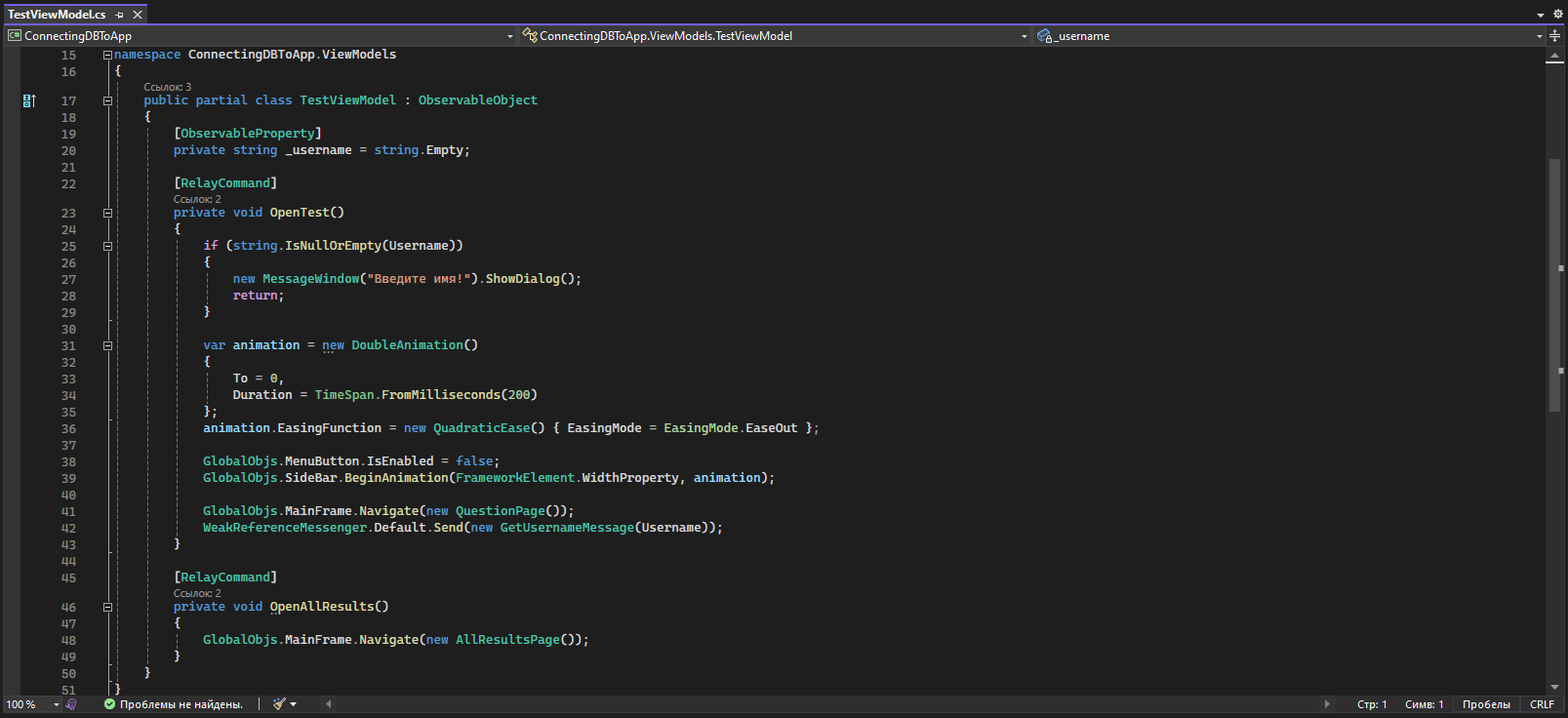
 Рисунок 6. Интерфейс и разметка страницы SQLite

 Рисунок 7. Класс с логикой страницы SQLite

*Начальная страница теста* отображается при переходе в раздел с тестом. С помощью неё можно ввести имя и начать прохождение теста, либо просмотреть все результаты пользователей.

Класс TestViewModel реализует такой функционал как переход на страницу со всеми результатами и функцию начала теста с проверкой введённого имени в поле ввода.

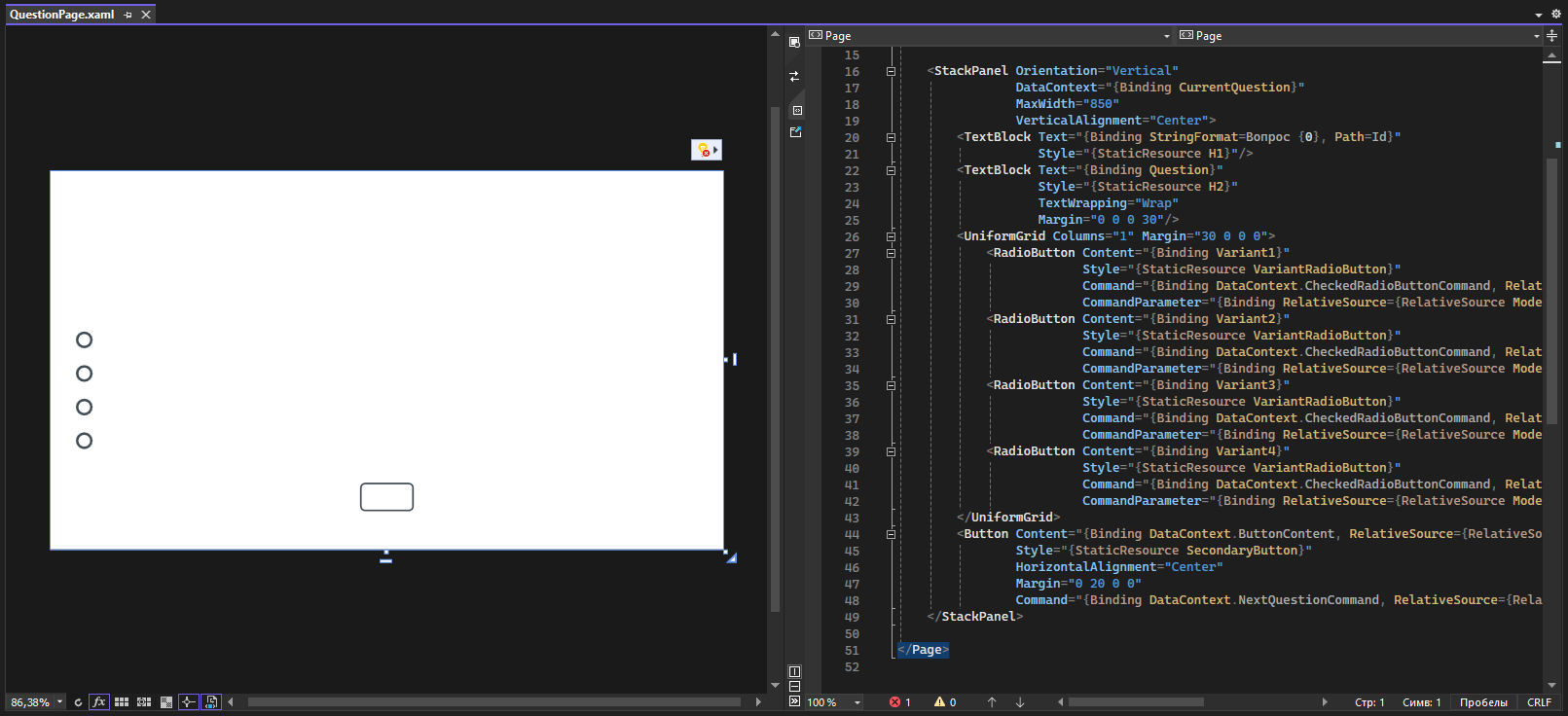
 Рисунок 8. Интерфейс и разметка начальной страницы теста

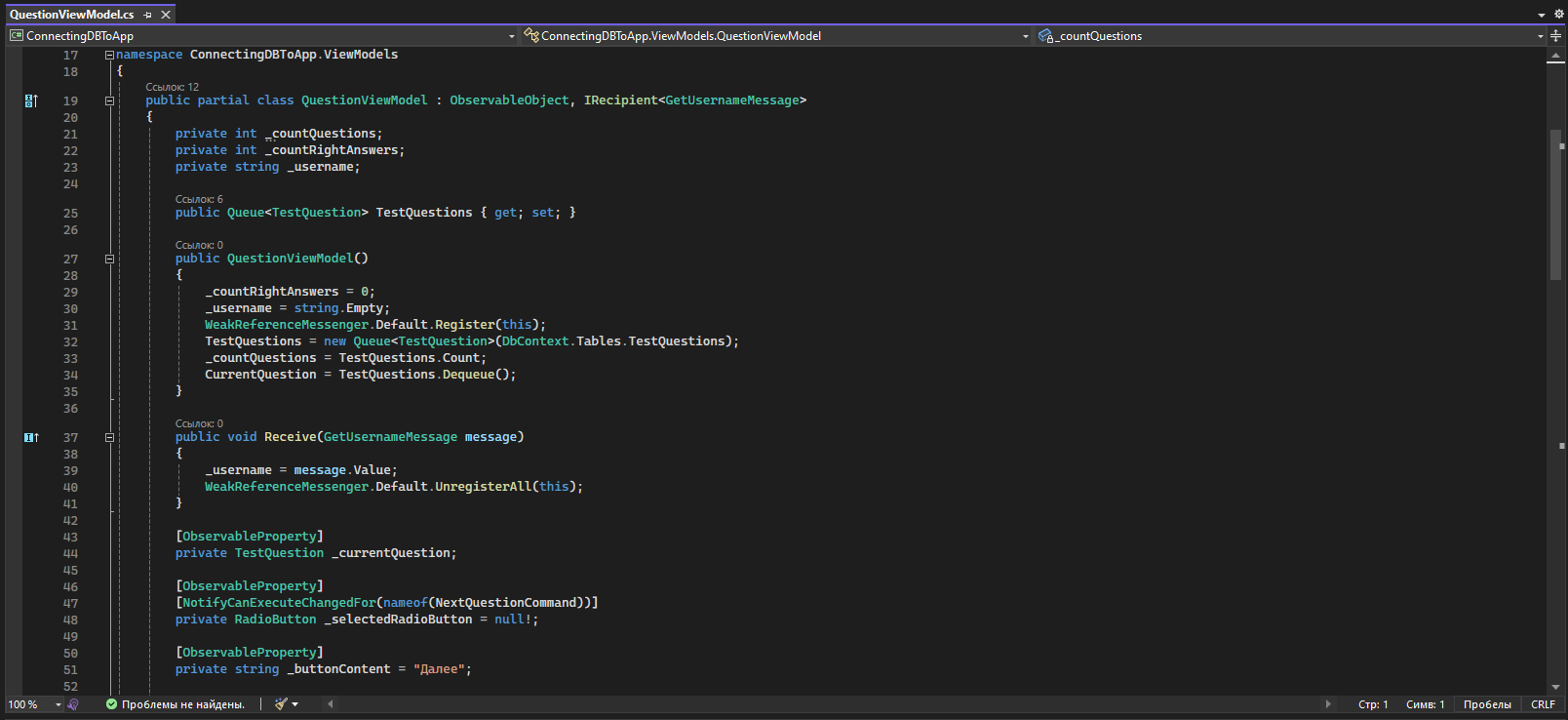
 Рисунок 9. Класс с логикой начальной страницы теста

*Страница с вопросом* отображает текущий вопрос теста. Интерфейс состоит из вопроса, четырёх вариантов ответа и кнопкой перехода на следующий вопрос.

Класс QuestionViewModel реализует:

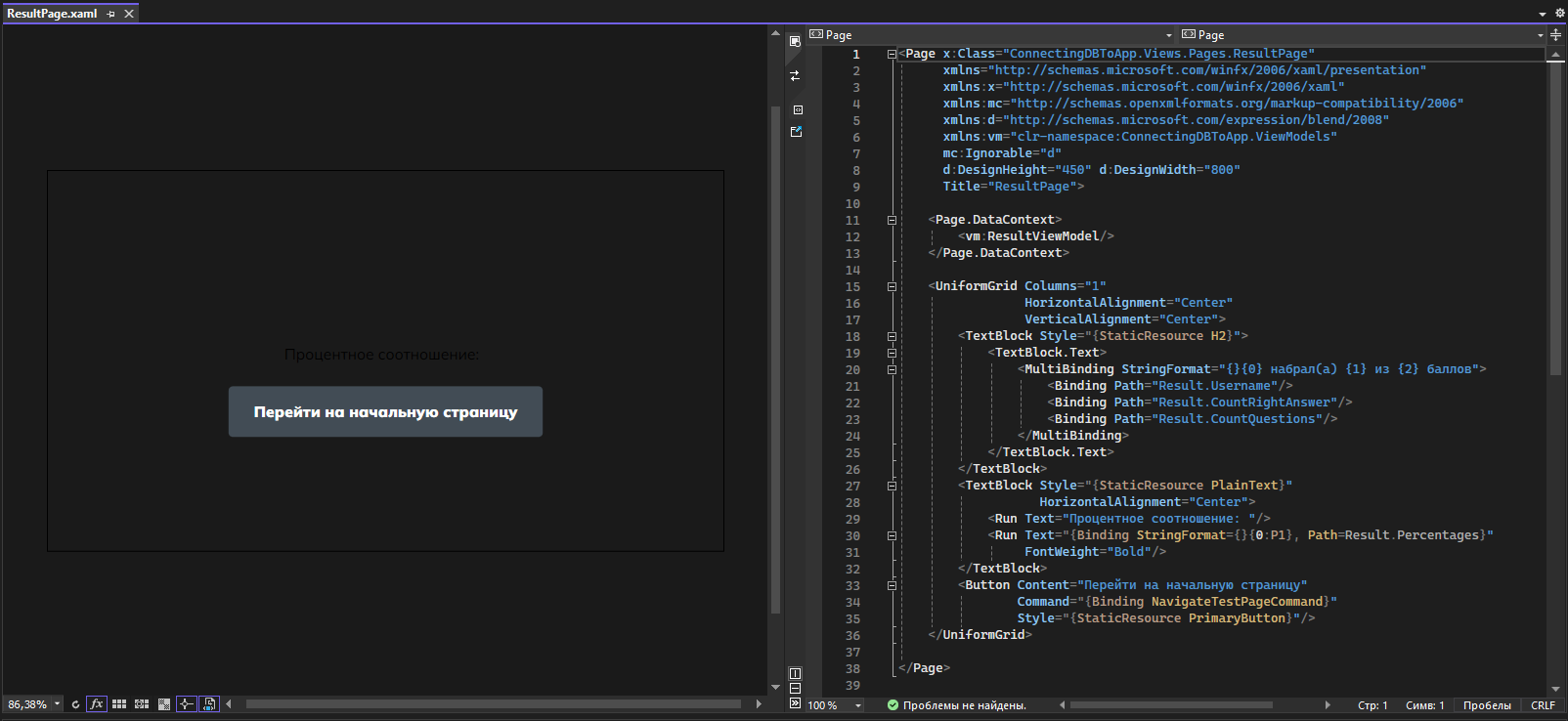
* переход между вопросами теста
* высчитывание результата
* отправка результата на страницу с итоговым результатом

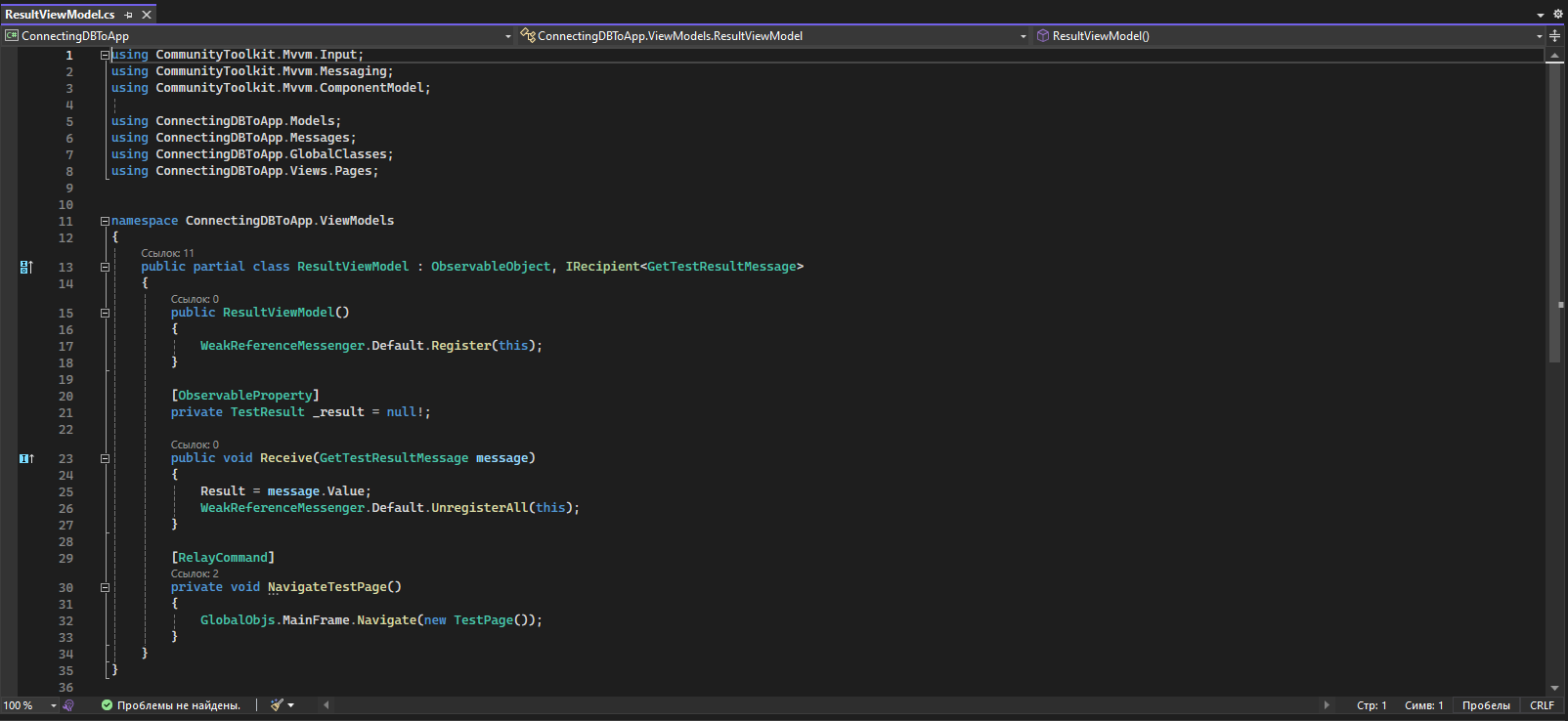
Рисунок 10. Интерфейс и разметка страницы с вопросом

Рисунок 11. Класс с логикой страницы с вопросом

*Страница с результатом теста* отображает результат теста. На ней находится текстовые блоки с результатом и кнопкой возвращения на начальную страницу теста.

Класс ResultViewModel принимает вычисленный результат от класса QuestionViewModel и передаёт его странице с результатом теста для его отображения. Ещё этот класс реализует ту самую функцию возвращения на начальную страницу раздела с тестом.

Рисунок 12. Интерфейс и разметка страницы с результатом теста

Рисунок 13. Класс с логикой страницы с результатом теста

*Страница со списком всех результатов* отображает все результаты пользователей, проходивших тест. Состоит из текстового блока, списка элементов и кнопкой удаления результатов.

Класс AllResultsViewModel передаёт данные о всех результатах пользователей данной странице, реализует функции возвращения на начальную страницу раздела с тестом и удаления всех результатов.

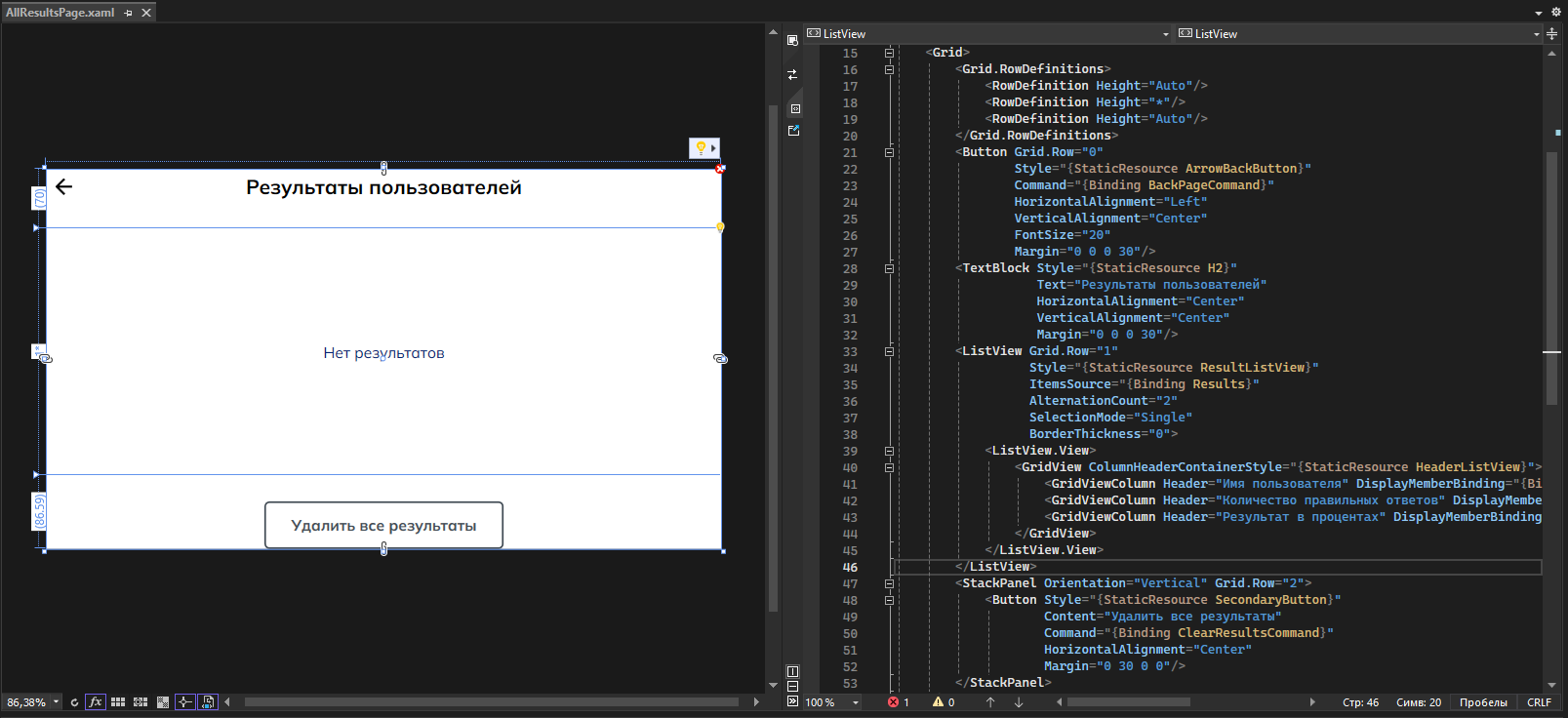


Рисунок 14. Интерфейс и разметка страницы со списком всех результатов

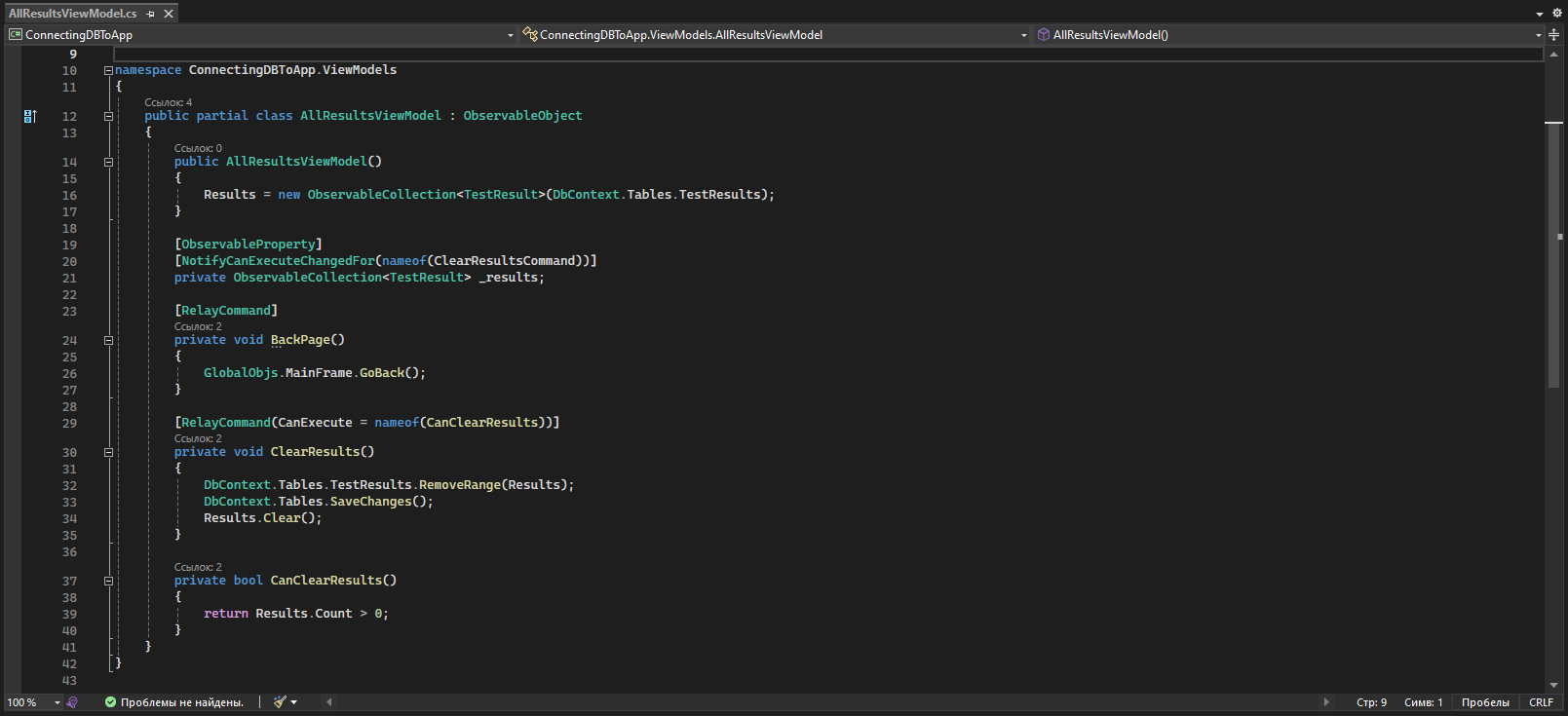
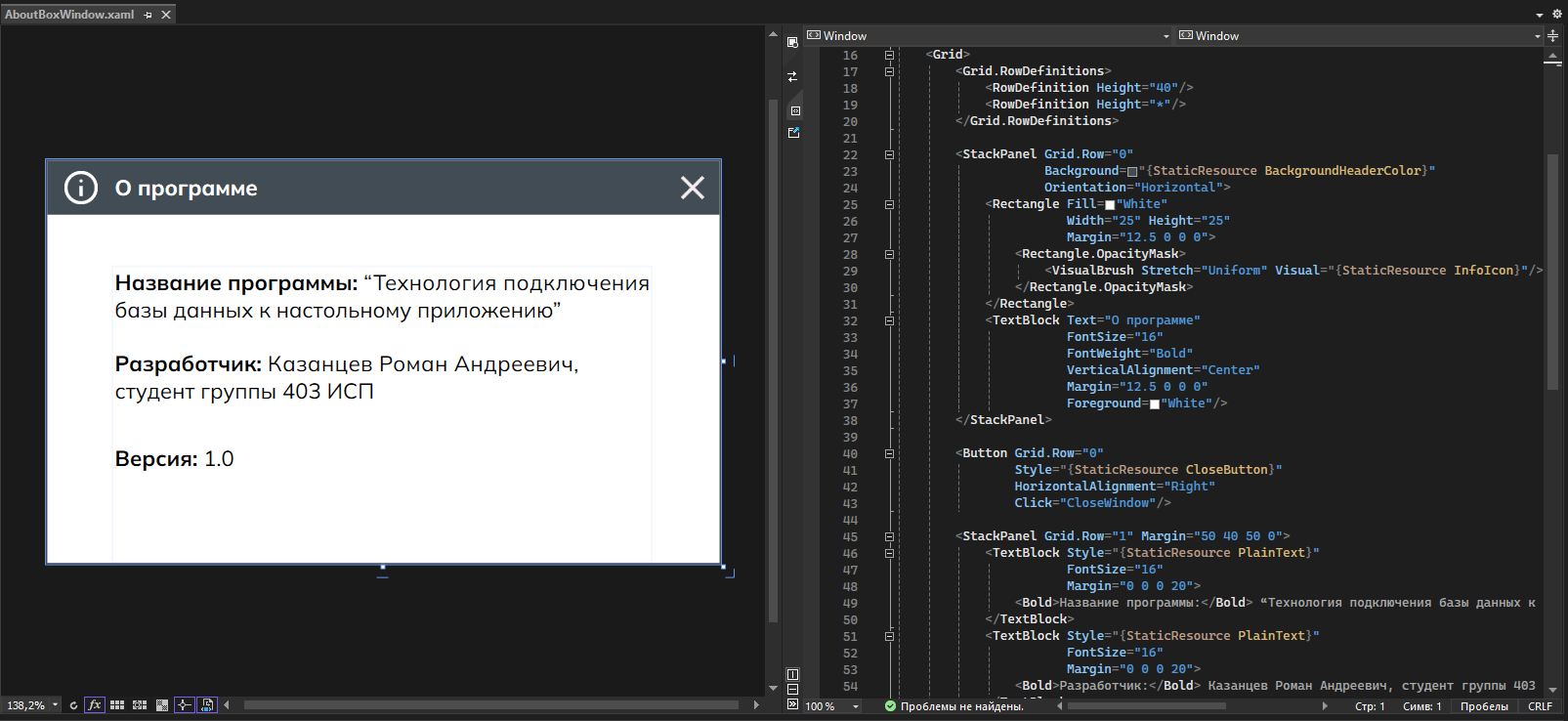


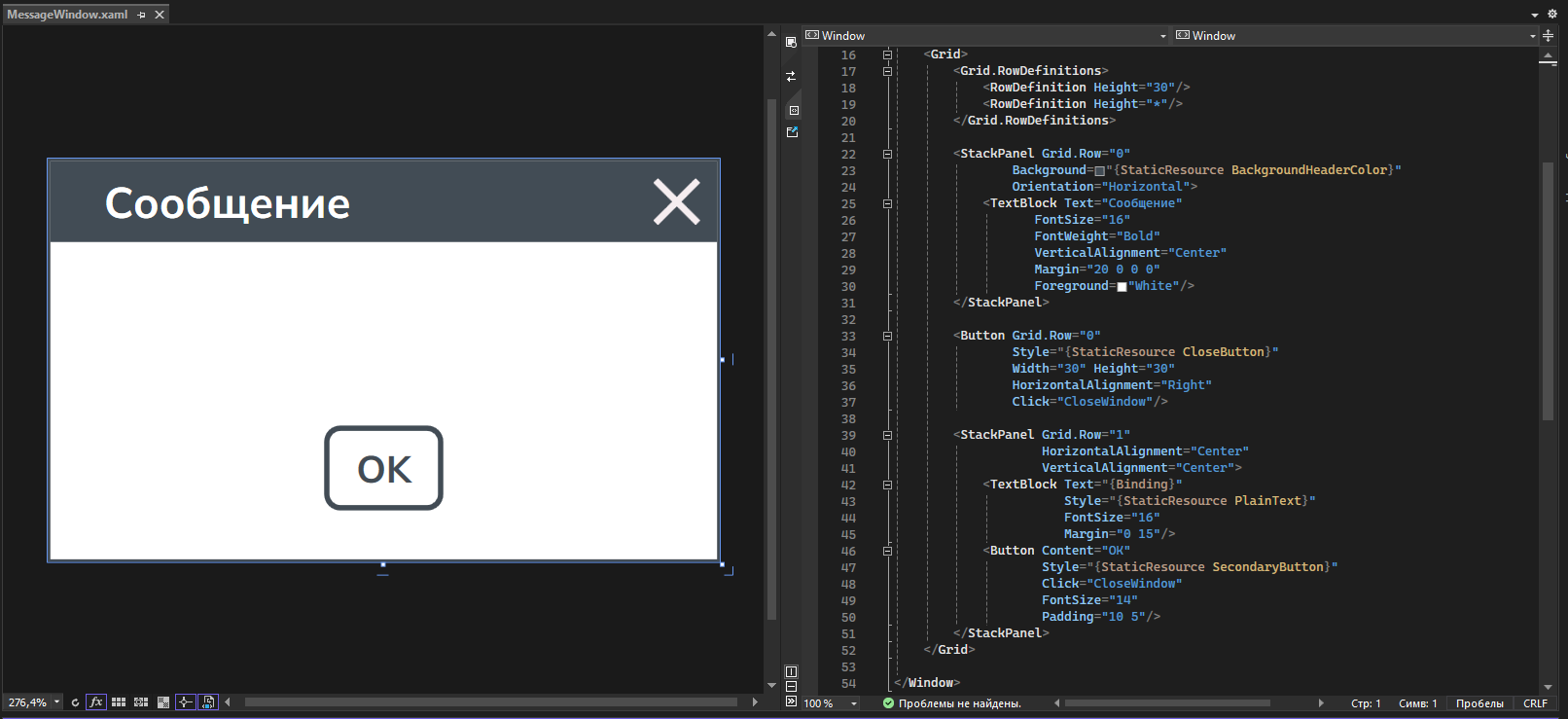
Рисунок 15. Класс с логикой страницы со списком всех результатов

Данная страница

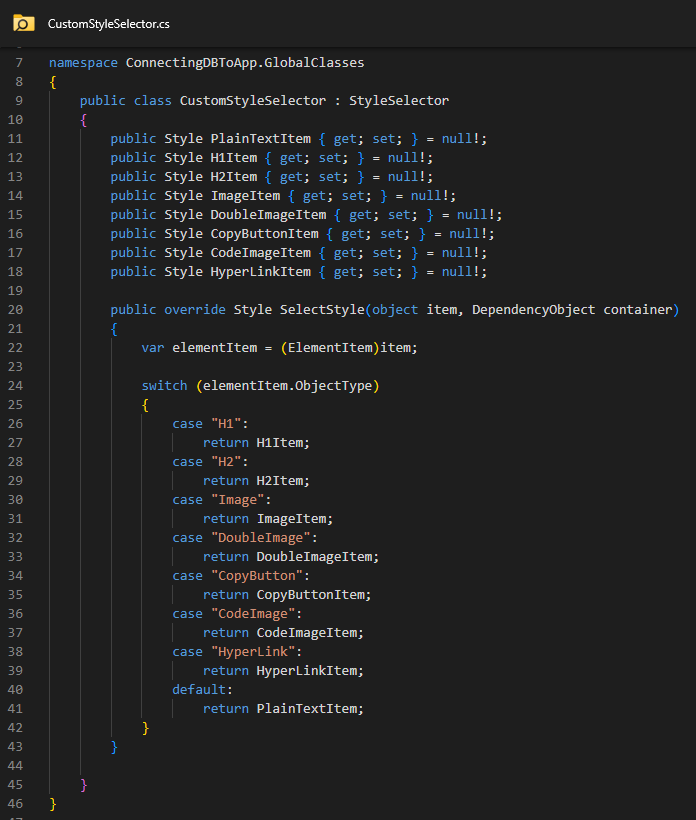
*Окно «О программе»*, состоящее только из текстовых блоков, отображает общие сведения о программе.

 Рисунок 15. Интерфейс и разметка окна «О программе»

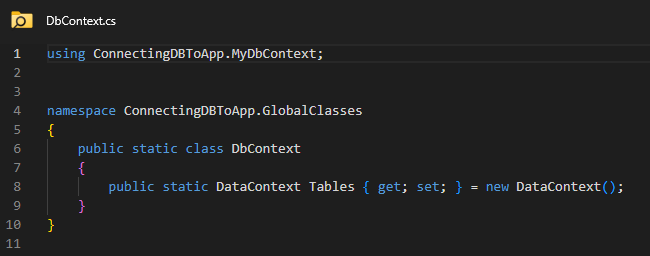
*Всплывающее окно с сообщением* предназначено для передачи какого-либо сообщения пользователю. Окно содержит текстовый блок с сообщением и кнопкой «ОК», закрывающей окно.

 Рисунок 16. Интерфейс и разметка всплывающего окна

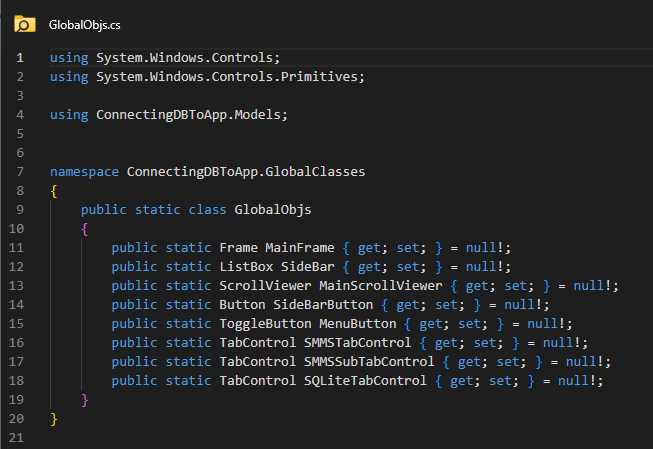
Класс CustomStyleSelector предназначен для выбора стиля, который нужно применить элементам на таких страницах как SSMS и SQLite.

Рисунок 17. Класс CustomStyleSelector

Класс DbContext предназначен для хранения контекста данных из базы данных. Этот класс объявлен глобально для того, чтобы работать с данными в любом классе, где это необходимо.

 Рисунок 18. Класс DbContext

Класс GlobalObjs предназначен для хранения пользовательских элементов глобально. Это сделано для того, чтобы редактировать и управлять их состоянием из любого места программы.

 Рисунок 19. Класс GlobalObjs

Класс MainWindowMethods предназначен для делегирования таких функций окна как свёртывание, открытие в оконном/полноэкранном режиме, закрытие.

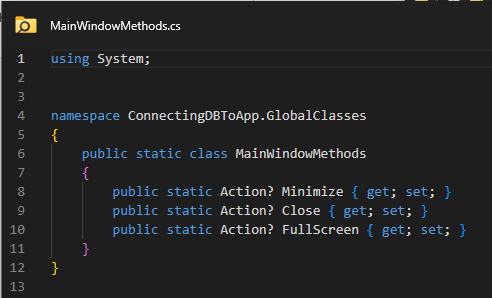


Рисунок 18. Класс MainWindowMethods

Классы GetTestResultMessage и GetUsernameMessage предназначены для создания сообщений, которые передаются между страницами.

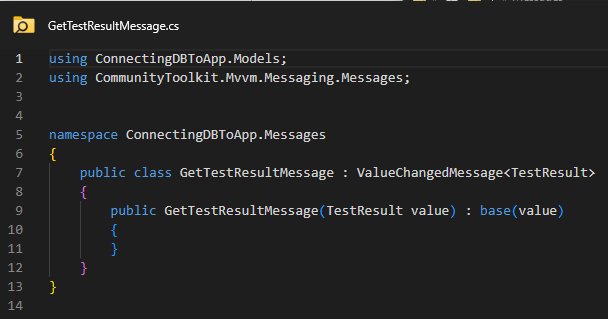


Рисунок 18. Класс GetTestResultMessage

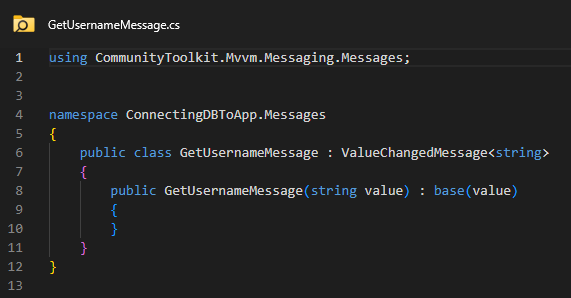


Рисунок 18. Класс GetUsernameMessage

Интерфейс ElementItem описывает данные элементов страниц SSMS и SQLite.

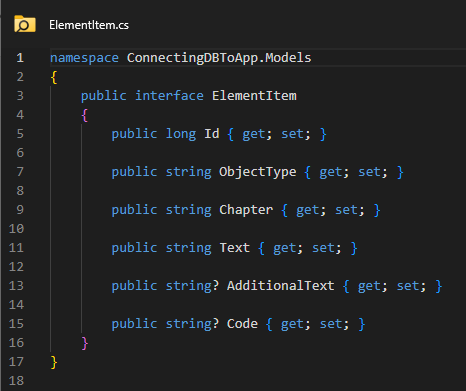
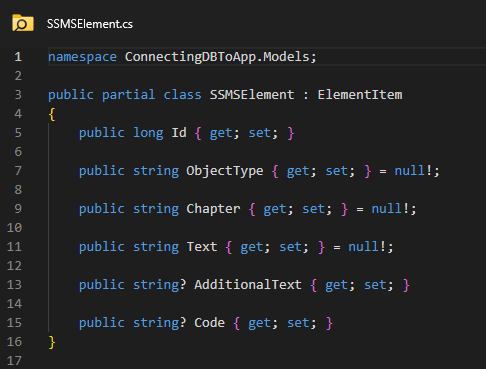
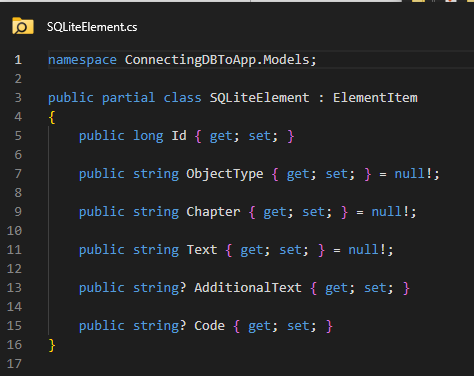


Рисунок 18. Класс ElementItem

Класс SQLiteElement и SSMSElement абсолютно идентичны и реализуют интерфейс ElementItem. Они описывают элементы соответствующих страниц из базы данных.

 Рисунок 18, 19. Классы SQLiteElement и SSMSElement

Класс TestQuestion описывает вопрос теста, который берётся из базы данных.

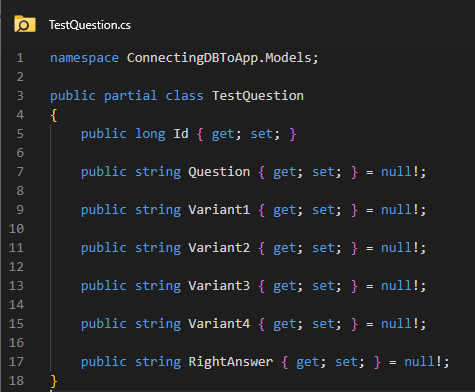


Рисунок 18. Класс TestQuestion

Класс TestResult описывает результат теста.

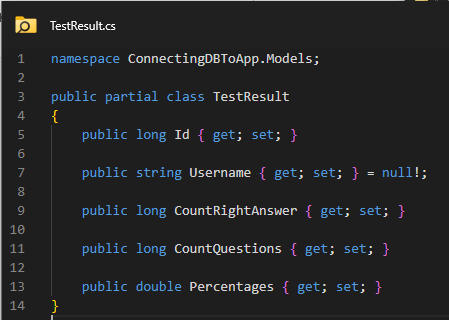
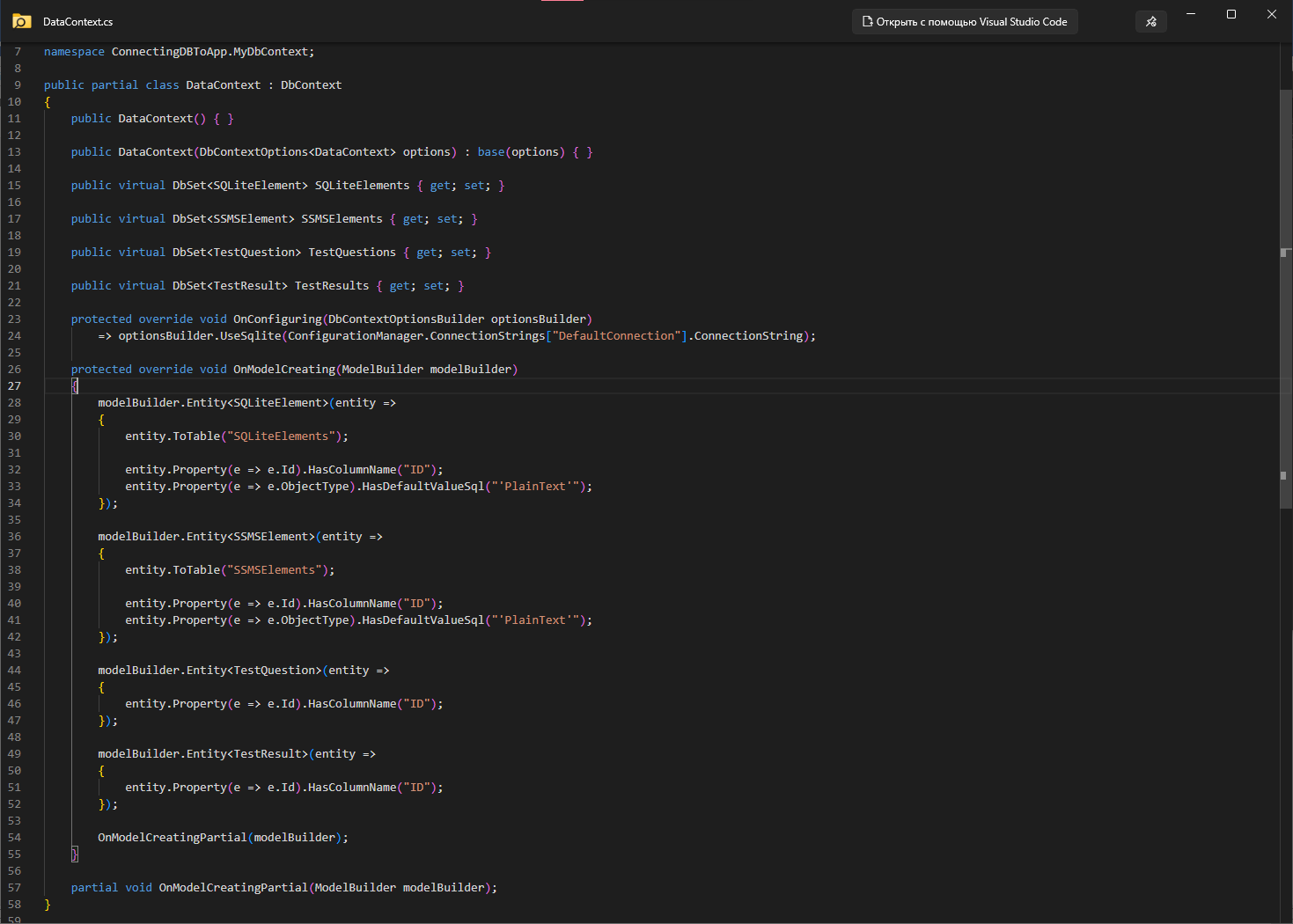


Рисунок 18. Класс TestResult

Класс DataContext предназначен для взаимодействия с базой данных. С помощью него реализуется связь с ней. Также с помощью него осуществляется добавление, редактирование и удаление данных.

 Рисунок 18. Класс DataContext

**2.2 Тестирование и отладка программного приложения**

Тестирование поля ввода на начальной странице теста

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Входные данные** | **Ожидаемый результат выполнения теста** | **Результат выполнения теста** |
| 1 | Пустое поле ввода | Сообщение о том, что имя пользователя не было введено |  |
| 2 | Иванов Иван | Прохождение теста. | Открывается страница с первым вопросом теста. Начался тест. |
| 3 | Иванов Иван Иванович | Сообщение о том, что имя пользователя слишком длинное. |  |

**2.3 Описание программного приложения**

1. Общие сведения

Наименование программы: «Технология подключения базы данных к настольному приложению»

Программное обеспечение, необходимое для функционирования программы: exe-файл программы

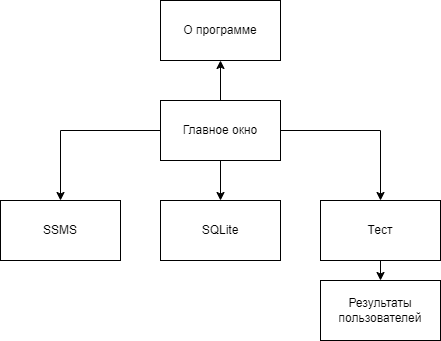
Языки программирования, на которых написана программа: язык программирования C# и язык разметки XAML

2. Функциональное назначение

Назначение программы: «Технология подключения базы данных к настольному приложению» - программа, выполненная в виде электронного справочника. Эта программа предназначена для хранения справочных материалов по технологиям подключения БД к приложению, а также для тестирования пользователей по этим справочным материалам.

3. Описание логической структуры

Структура программы с описанием функций составных частей и связи между ними:



4. Используемые технические средства

*Требования к составу и параметрам технических средств:*

* Процессор: 32- или 64-разрядный процессор с тактовой частотой 1 ГГц или выше;
* Оперативная память: 2 ГБ;
* Свободное место на жестком диске: 200 МБ свободного места на диске;
* Монитор: Разрешение 1280 x 800;
* Клавиатура, мышь.

5. Вызов и загрузка

Способ вызова программы с соответствующего носителя данных: открытие ярлыка «Технология подключения базы данных к настольному приложению».

Входные точки в программу: ярлык «Технология подключения базы данных к настольному приложению», exe-файл «Технология подключения базы данных к настольному приложению».

6. Входные/выходные данные

Входные и выходные данные в программе присутствуют только в разделе с тестом. Входными данными являются фамилия и имя пользователя, а выходными данными являются результат теста и просмотр всех результатов тестов.

**Заключение**

В ходе выполнения курсового проекта была достигнута установленная цель, а именно было разработано программное приложение «Технология подключения базы данных к настольному приложению»

В теоретической части курсового проекта была проанализирована предметная область, разработано техническое задание на разработку программного приложения, проанализированы существующие решения и был сделан выбор программных средств с его обоснованием.

В практической части курсового проекта были выполнены следующие задачи:

1. Описаны этапы проектирования и разработки программного приложения;
2. Было произведено тестирование программного приложения;
3. Описано созданное программное приложение.

После всей проделанной работы, разработанное приложения решило проблему и подтвердило гипотезу курсового проекта.

**Список использованных источников**

1. Доступ к данным в приложениях Windows - Windows apps | Microsoft Learn — URL: <https://learn.microsoft.com/ru-ru/windows/apps/develop/data-access/>
2. Подключение к базе данных — URL: <https://studfile.net/preview/16469734/>
3. Типы пользовательского интерфейса: классификация IT-систем — URL: <https://journalovirus.ru/tipy-polzovatelskogo-interfeysa-klassifikatsiya-it-sistem>
4. Интерфейс - что это такое простыми словами и какие бывают виды интерфейсов — URL: <https://neiros.ru/blog/code/chto-znachit-termin-interfeys-i-18-vidov-ego-realizatsii/>
5. Языки - Visual Studio — URL: <https://visualstudio.microsoft.com/ru/vs/features/web/languages/>
6. Visual Studio: IDE и редактор кода для разработчиков и групп, работающих с программным обеспечением — URL: <https://visualstudio.microsoft.com/ru/>
7. CLion — Википедия — URL: <https://ru.wikipedia.org/wiki/CLion>
8. Electron (фреймворк) — Википедия — URL: <https://ru.wikipedia.org/wiki/Electron_(фреймворк)>
9. Фреймворки и библиотеки для кроссплатформенной разработки десктопных программ / Хабр — URL: <https://habr.com/ru/articles/528614/>
10. Windows Presentation Foundation (WPF) — URL: <https://csharp.webdelphi.ru/windows-presentation-foundation-wpf/#_wpf>
11. Причины использования современных классических приложений - .NET | Microsoft Learn — URL: <https://learn.microsoft.com/ru-ru/dotnet/architecture/modernize-desktop/why-modern-applications>

**Приложение А**

Таблица 1 – Ответы на вопросы к тесту

|  |  |
| --- | --- |
| **Вопрос** | **Ответ** |
| 1. С помощью какого диспетчера пакетов можно установливать библиотеки в свой проект? | NuGet |
| 2. С помощью какой вкладки можно посмотреть подключенные БД из SQL Management Studio? | Обозреватель серверов |
| 3. Какой класс отвечает за подключение к базе данных в библиотеке SqlClient? | SqlConnection |
| 4. Какие параметры имеет объект класса SqlCommand, передаваемые в конструктор? | SQL-запрос в виде строки, объект SqlConnection |
| 5. Для чего нужен класс SqlDataReader? | Для чтения данных |
| 6. На основе какого свойства DataGrid формирует таблицу? | ItemsSource |
| 7. Какая команда отвечает за создание контекста и моделей базы данных в библиотеке Entity Framework Core 7? | Scaffold-DbContext |
| 8. Какие библиотеки нужны для работы с SQLite & EF Core 7? | Microsoft.EntityFrameworkCore.Sqlite & Microsoft.EntityFrameworkCore.Tools |
| 9. С помощью какой строчки кода можно проверить, была ли создана БД SQLite в проекте (в EF Core 7)? | <имя\_контекста>.Database.EnsureCreated() |
| 10. Какая строчка кода загружает таблицу для работы с ней в SQLite & EF Core 7? | <имя\_контекста>.<имя\_таблицы>.Load() |

**Приложение Б**

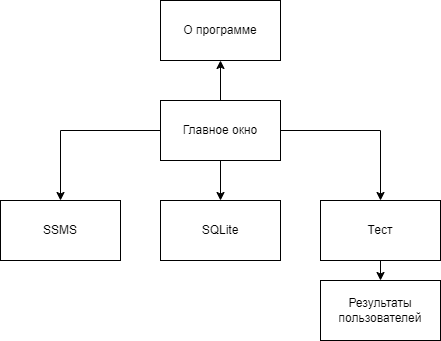
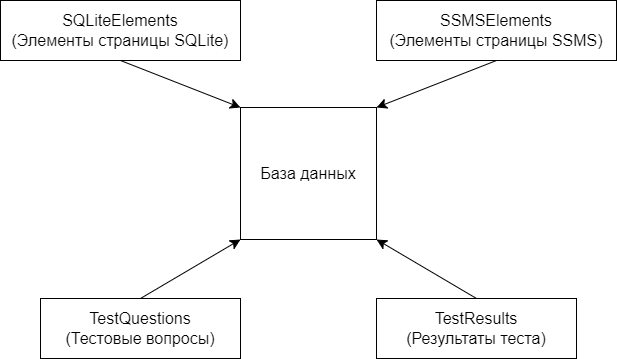


Рисунок Х - **Схема программного приложения**

 Рисунок Х - **Информационная модель базы данных**