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

Для запуска программы необходимо двойным щелчком мыши нажать на ярлык программы:

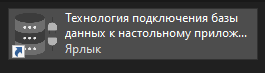
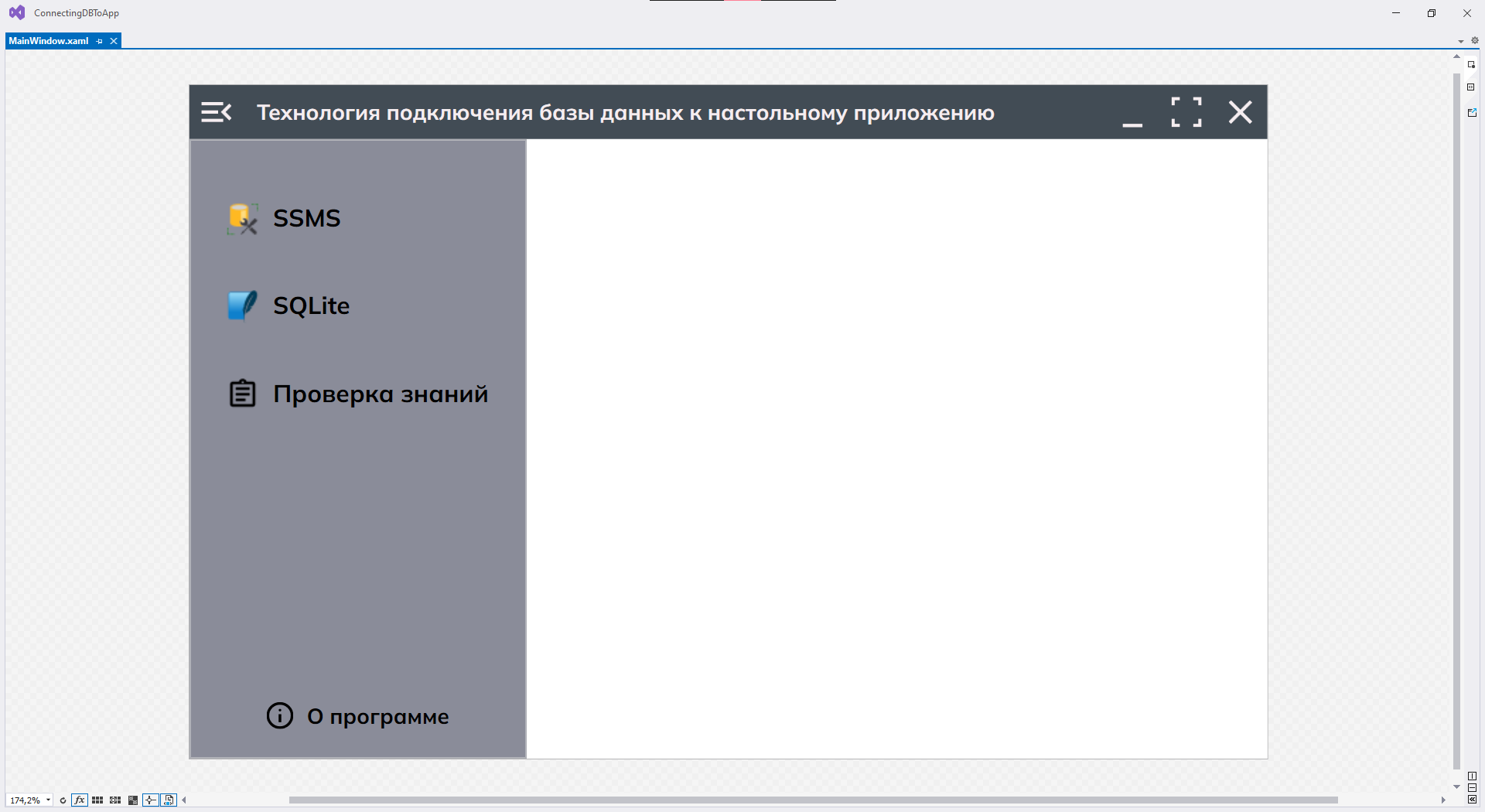


Рисунок 1 - Ярлык программного приложения «Технология подключения базы данных к настольному приложению»

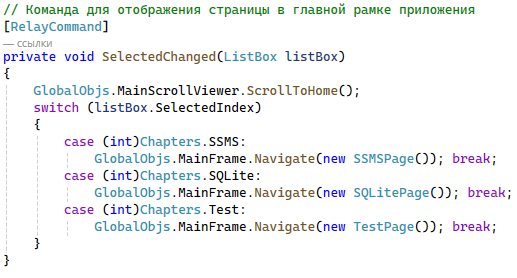
При запуске программного приложения открывается главное окно. С помощью этого окнаосуществляется навигация по всей программе.



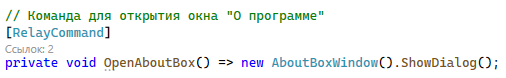
В левой части главного окна находится боковое меню, в котором находятся все разделы программного приложения. В правой части располагаются и отображаются страницы этих разделов.

К главному окну программы привязан свой класс – MainViewModel. Этот класс реализует всю логику для этого окна, а именно:

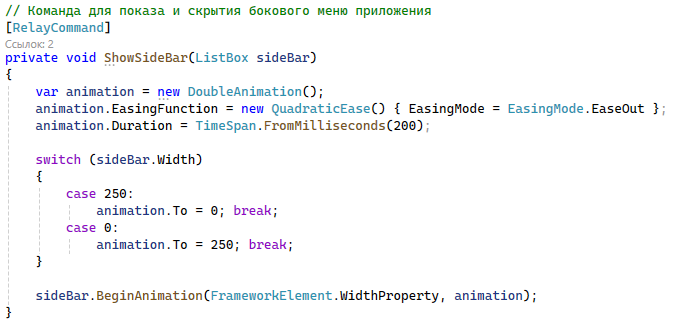
* переходы по разделам приложения



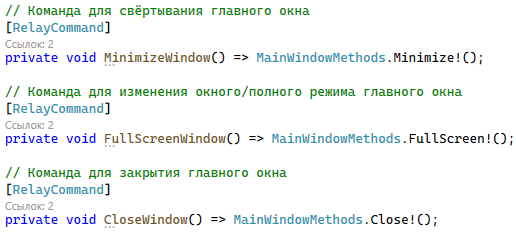
* открытие окна «О программе»



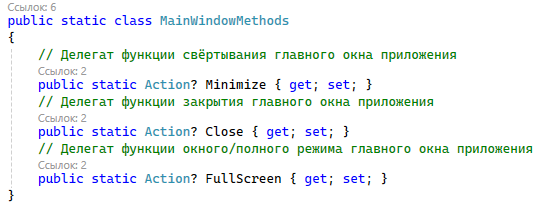
* показ/скрытие бокового меню



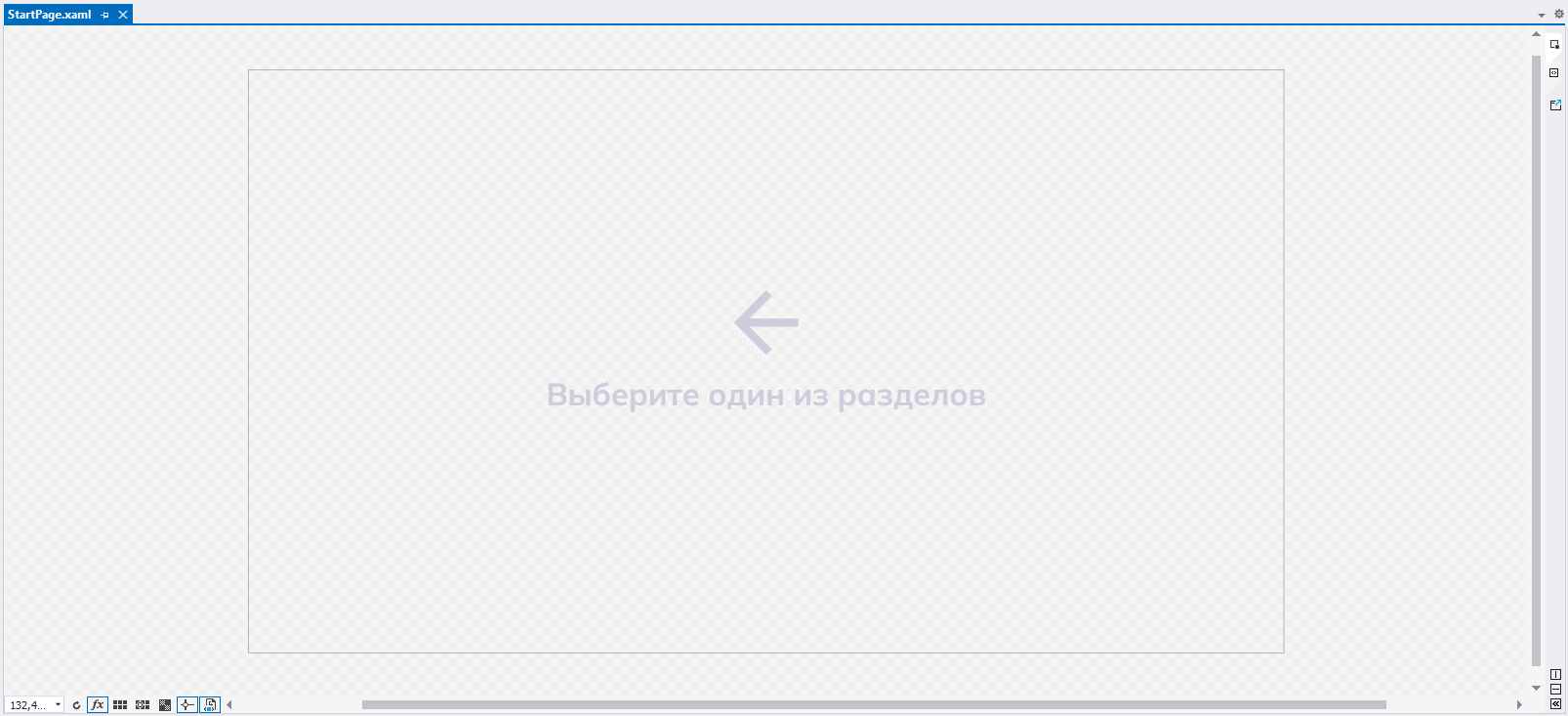
* реализация функционала для кнопок окна «Свернуть», «Полный экран», «Закрыть» при помощи класса MainWindowMethods.



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

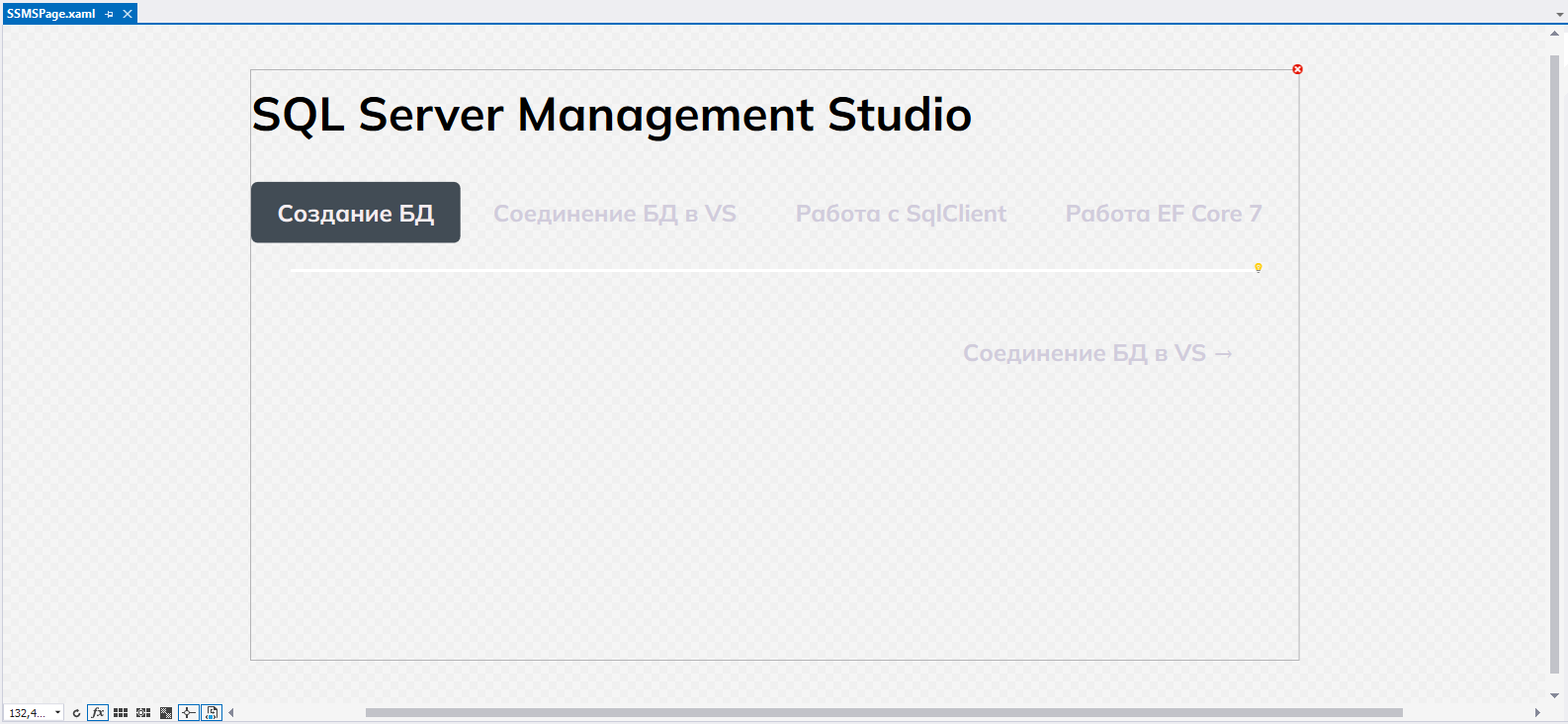


При запуске программы в правой части программы отображается стартовая страница приложения.



Назначение этой страницы предложить пользователю выбрать раздел, указывая на боковое меню.

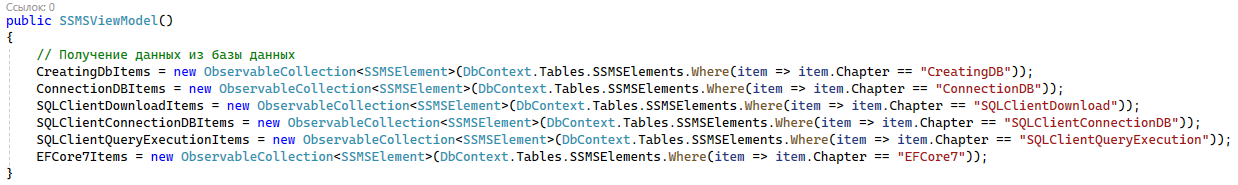
При выборе первого раздела приложения откроется страница с теоретическими сведениями о SQL Server Managament Studio.



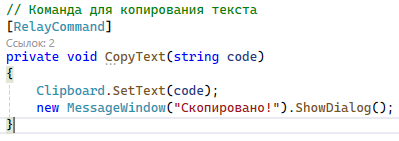
Данная страница выполнена в виде вкладок между которыми осуществляется перемещение по справочному материалу. Каждая вкладка является подразделом и в ней отображается часть определённой обучающей информации. А справочная информация рассказывает, как подключить базу данных SQL Server Managament Studio к своему проекту.

За логику данной страницы отвечает класс SSMSViewModel. Он реализует следующий функционал:

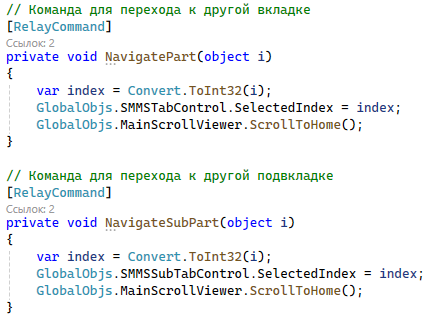
* получение данных из базы данных и последующая передача их странице



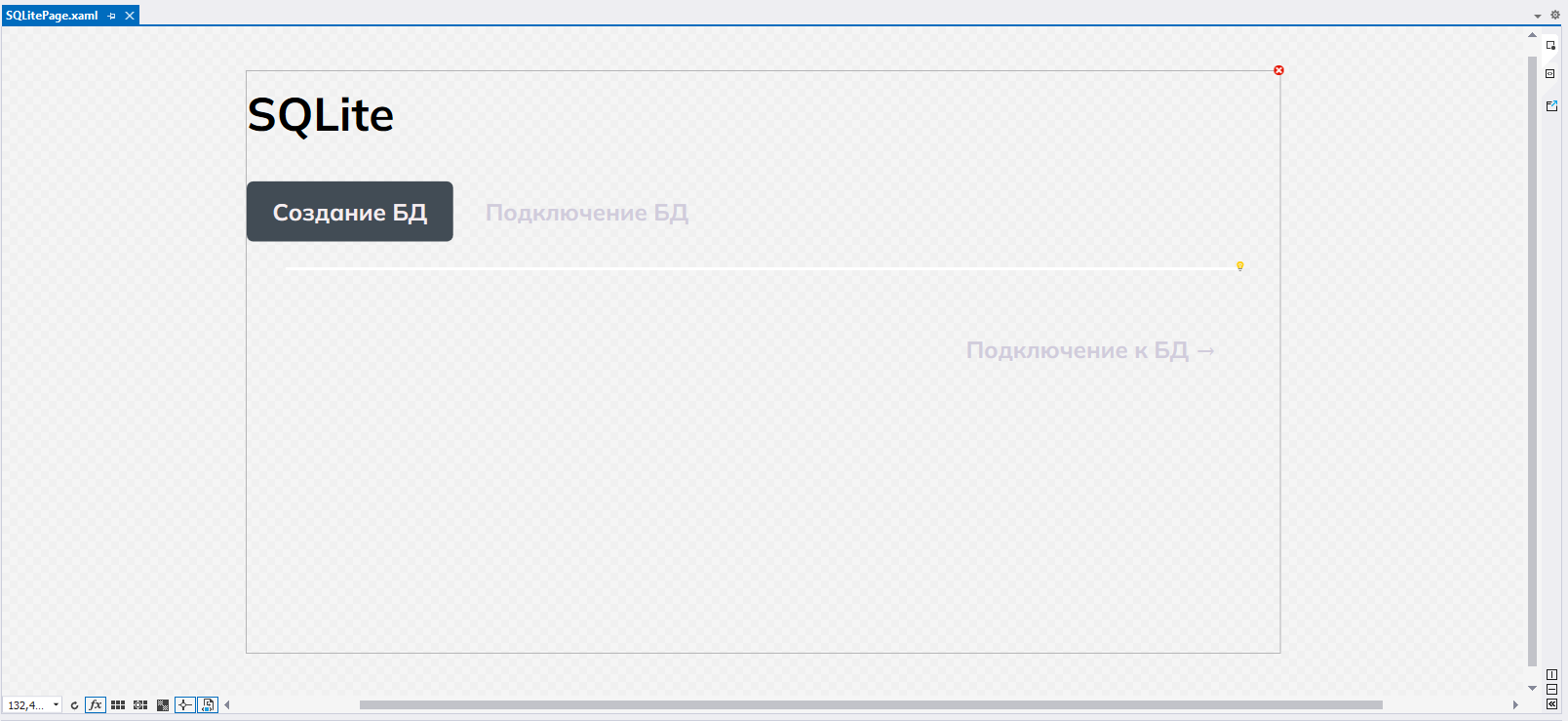
* функция для копирования текста в буфер обмена



* функции, реализующие перемещение по разделам с помощью гиперссылок в конце каждой вкладки страницы.



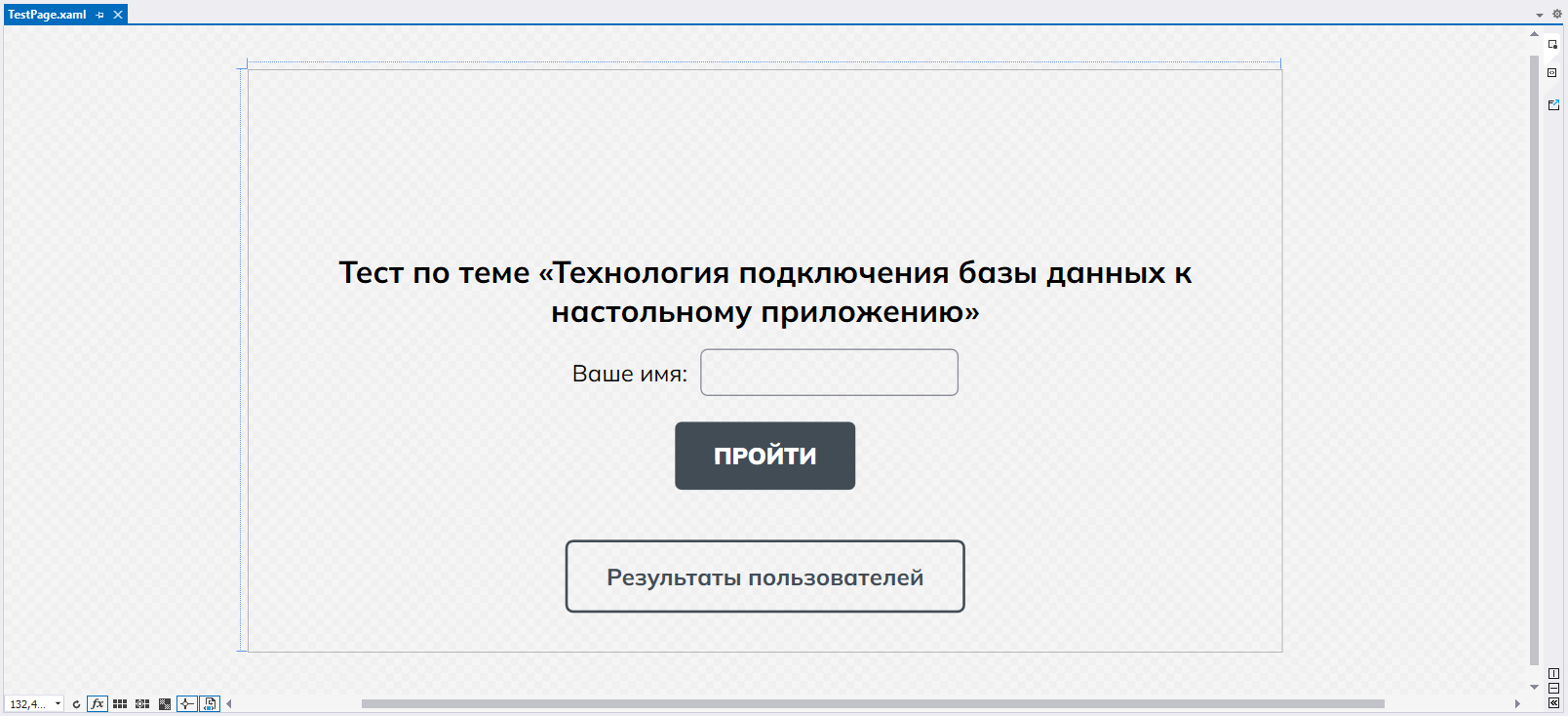
Вторым разделом приложения является страница с теоретическими сведениями о SQLite.



Пользовательский интерфейс полностью идентичен странице с SSMS. В нёмотображается обучающая информация, которая объясняет, как подключить базу данных SQLite к своему проекту.

Логику этого раздела определяет класс SQLiteViewModel. Он реализует тот же самый функционал, как и класс SSMSViewModel, только для страницы SQLite.

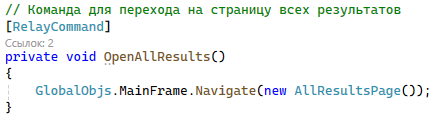
При выборе последнего раздела отобразиться начальная страница теста.

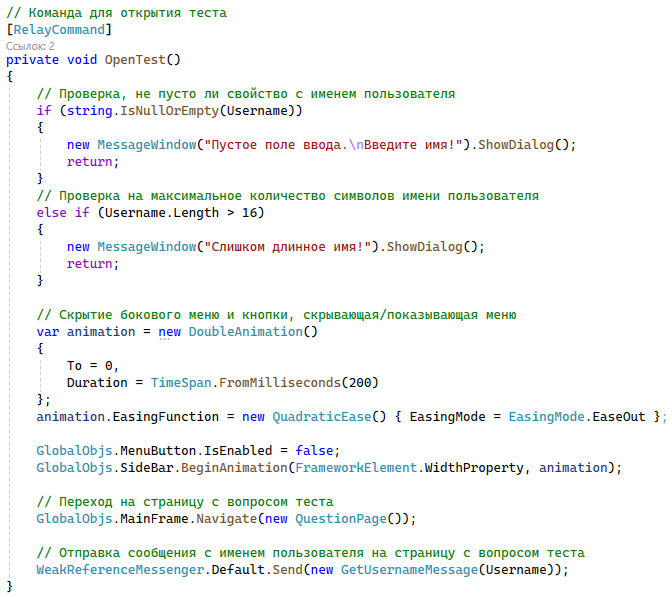


С помощью неё осуществляется начало прохождение теста просмотр результатов пользователей.

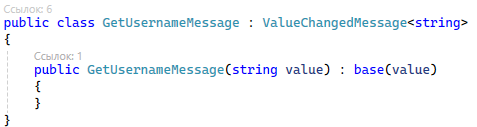
Начальная страница теста связана с классом TestViewModel. Он, в свою очередь, отвечает за следующие функции:

* переход на страницу со всеми результатами

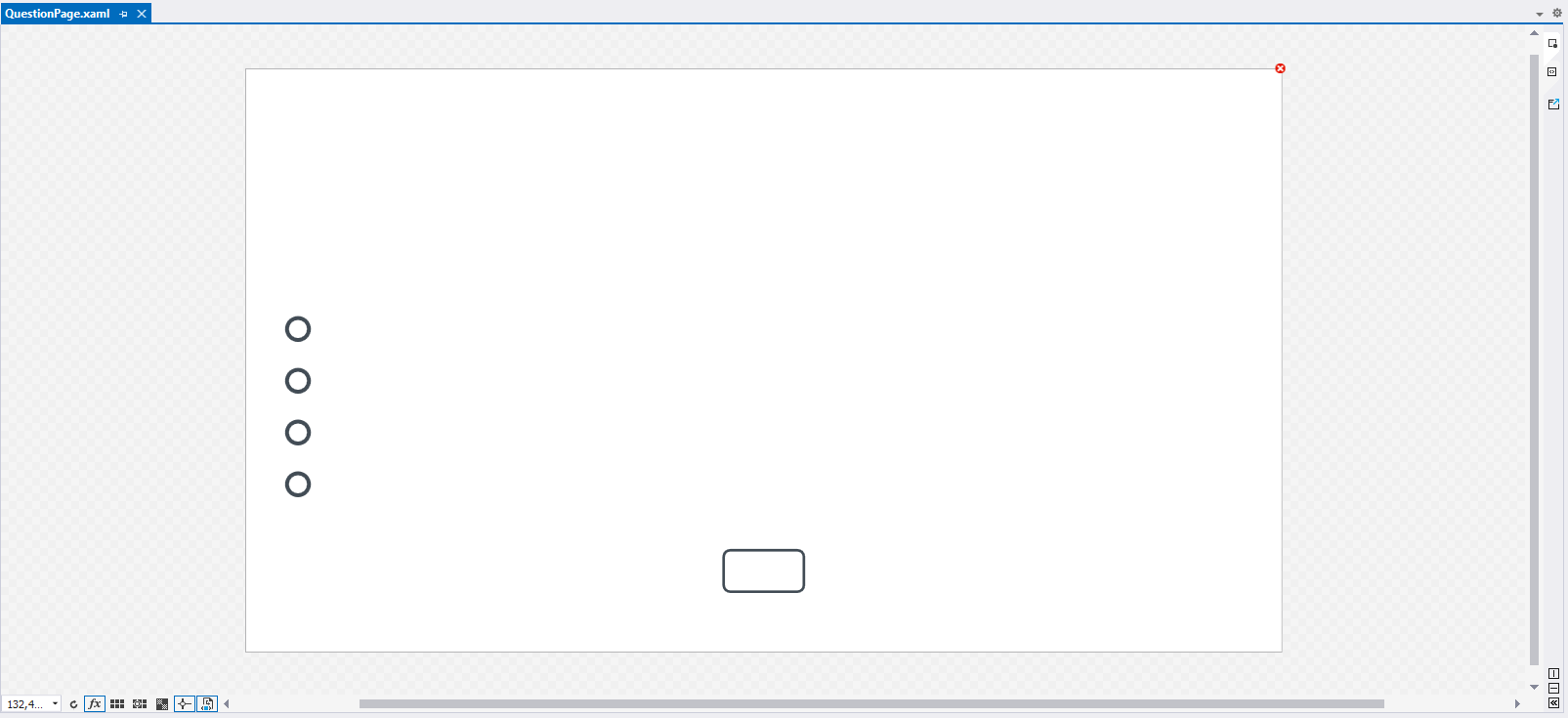


* начало прохождения теста
* отправка сообщения класса GetUsernameMessage на страницу с вопросом.
* 

Класс GetUsernameMessage, в свою очередь, предназначен для передачи имени пользователя.



Для того, чтобы начать прохождение теста, необходимо вписать в отведённое поле ввода имя пользователя, после чего нажать на кнопку «ПРОЙТИ». Выполнив эти действия, начнётся тестирование пользователя и отобразится страница с вопросом.



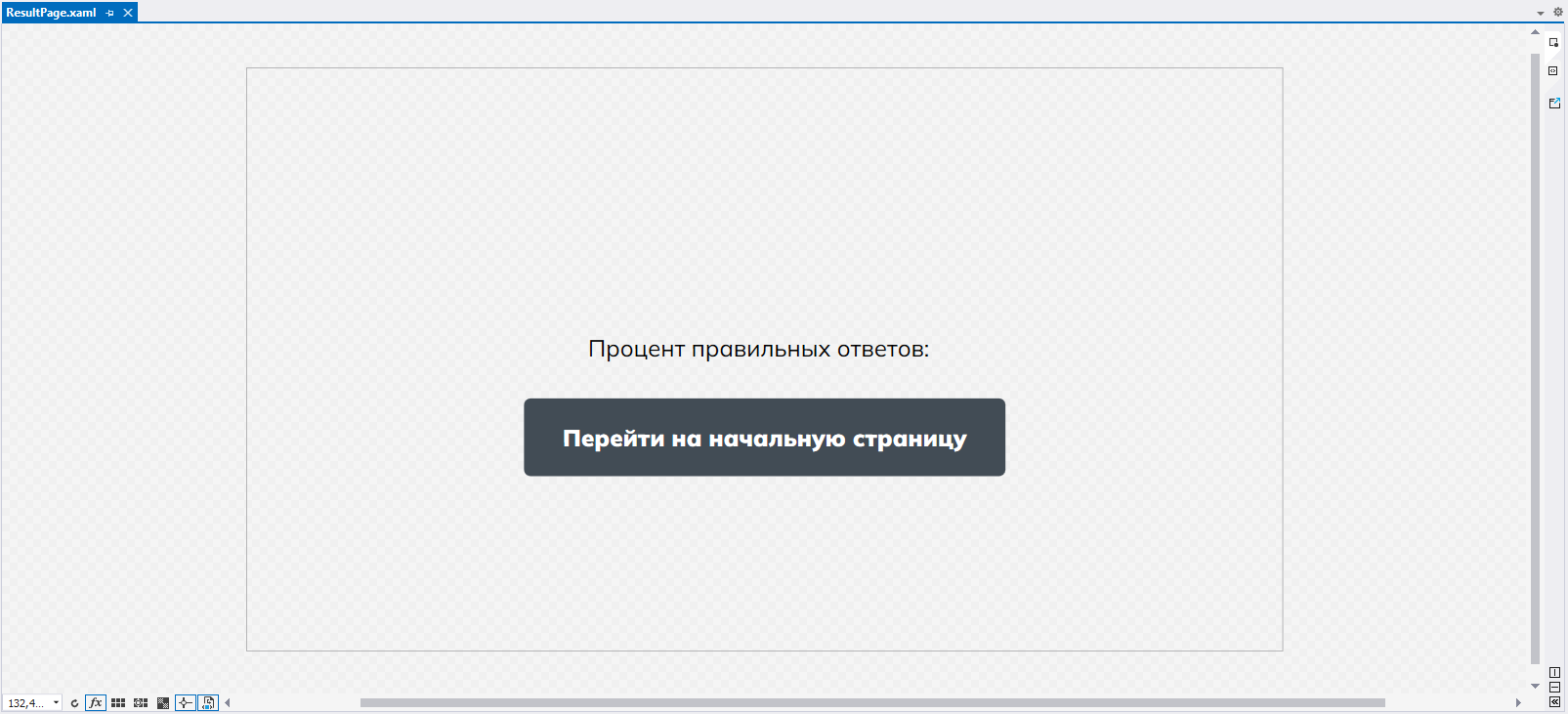
Эта страница состоит из текстового блока с вопросом, четырёх вариантов ответа в виде радиокнопок и кнопкой перехода на следующий вопрос.

Для получения и отображения вопроса страница связывается с таким классом как QuestionViewModel. Этот класс отвечает за получение всех вопросов из базы данных, а также выполняет некоторые функции страницы такие как:

* переход между вопросами теста
* высчитывание результата
* отправка сообщения класса GetTestResultMessage на страницу с итоговым результатом.

Класс GetTestResultMessage предназначен для передачи объекта TestResult. Этот объект описывает вопрос с вариантами ответа, а также хранит в себе правильный ответ.

При полном прохождении теста, в конце выполниться переход на страницу с результатом.



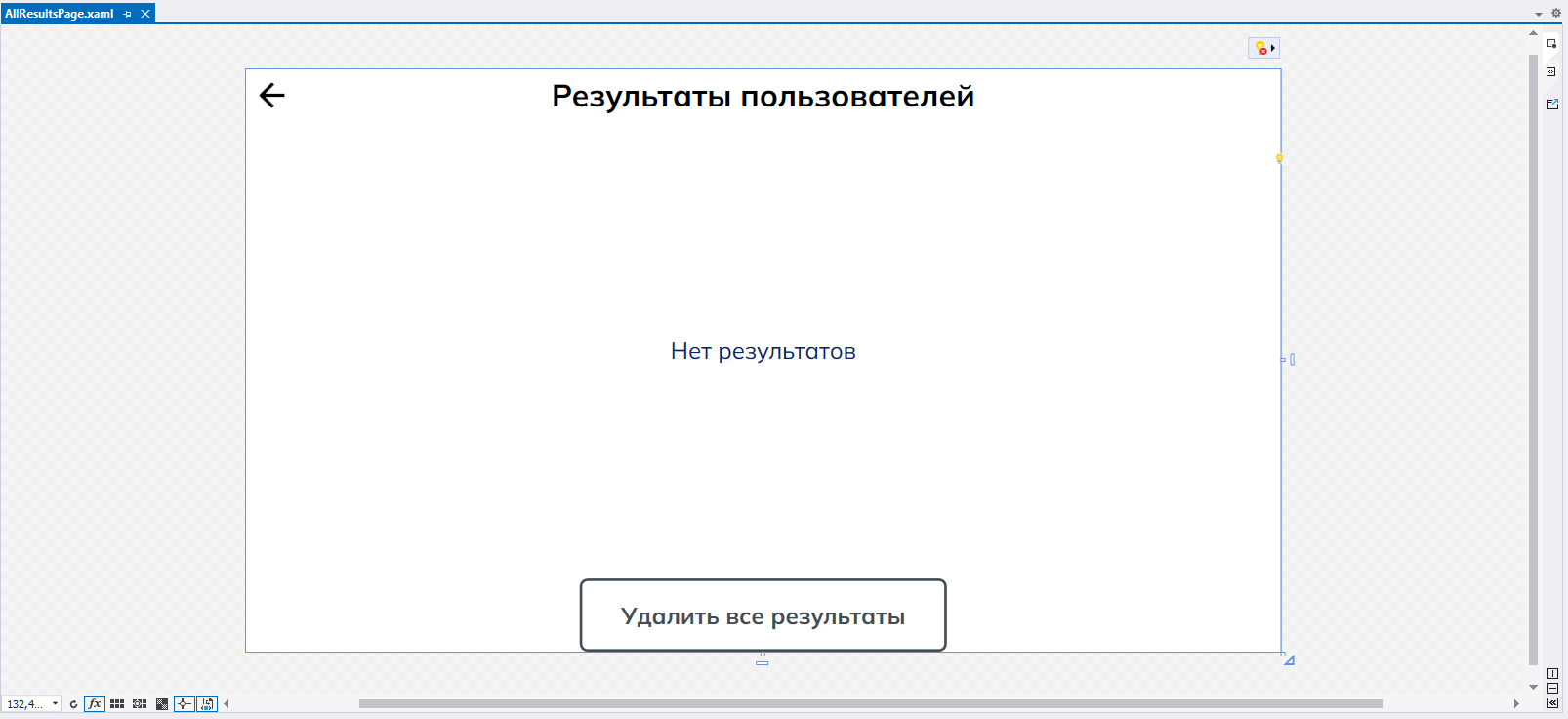
На ней находится текстовые блоки, отображающие имя пользователя, количество полученных баллов, процент правильный ответов, а также кнопка возвращения на начальную страницу теста.

К странице с результатом привязан класс ResultViewModel. Он принимает результат теста от класса QuestionViewModel и передаёт полученный результат на страницу для отображения результата теста.

Результат теста, в данном случае, представляется в виде объекта класса TestResult, который хранит в себе имя пользователя, количество вопросов, количество правильных ответов и процент правильных ответов.

Помимо этого, класс ResultViewModel реализует функцию возвращения на начальную страницу раздела с тестом.

Для того, чтобы посмотреть на свой результат вновь, на начальной странице теста есть кнопка «Результаты пользователей». Нажав на эту кнопку, отобразится страница результатов пользователей.

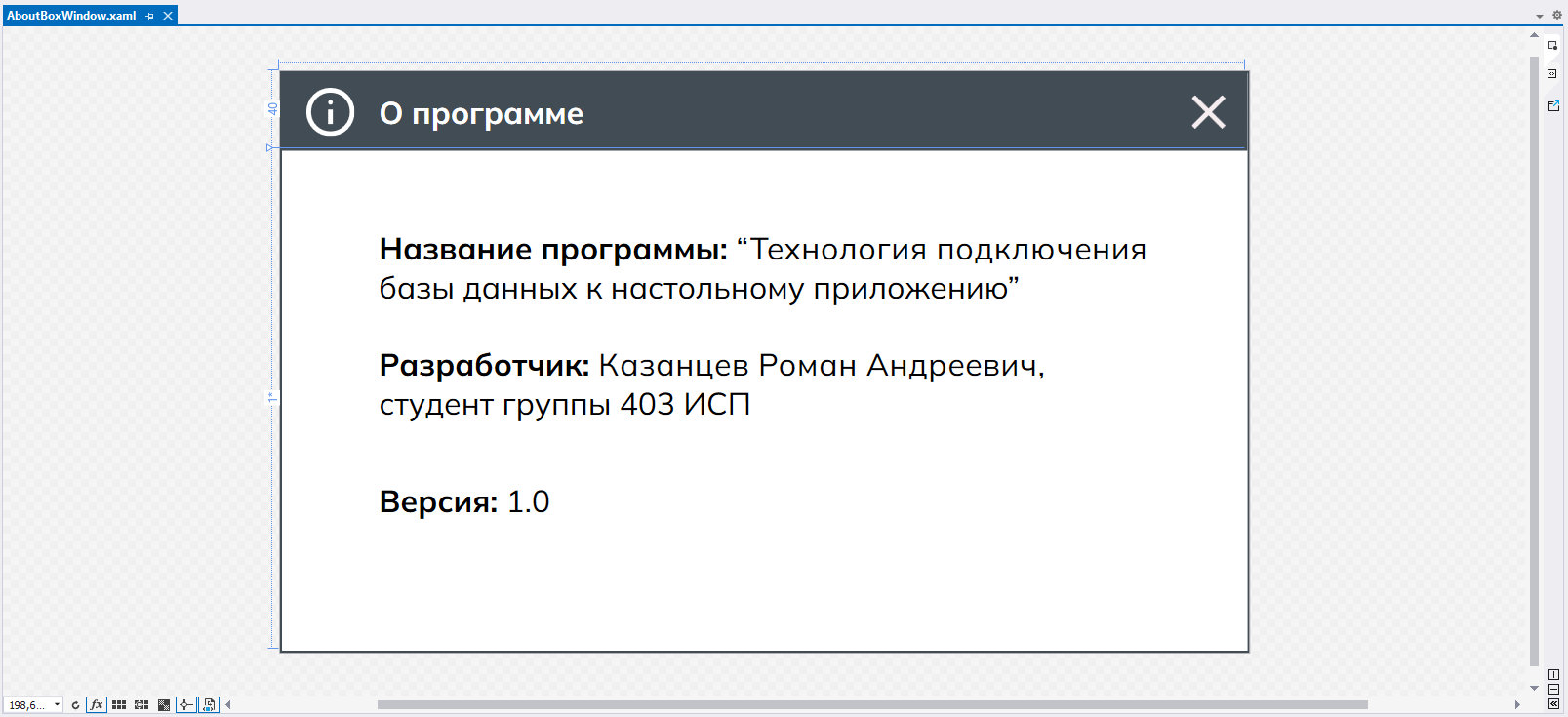


Интерфейс данной страницы состоит из заголовка, списка элементов и кнопки удаления результатов.

Список элементов страница получает от класса AllResultsViewModel в виде списка объектов класса TestResult. Помимо этого, класс реализует такие функции на странице как:

* возвращение на начальную страницу теста
* удаление всех результатов, которые отображаются в списке.

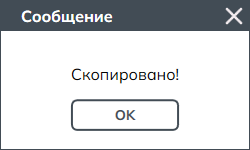
После разделов программного приложения, в самом низу бокового меню находится надпись «О программе». По нажатию этой надписи появится окно «О программе», состоящее из текстовых блоков, в которых отображаются общие сведения о программе.



В некоторых вкладках в первом и втором разделах встречаются кнопки копирования:

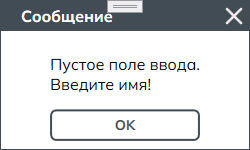
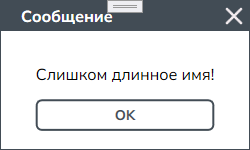


После нажатия по ним появляется всплывающее окно с сообщением:

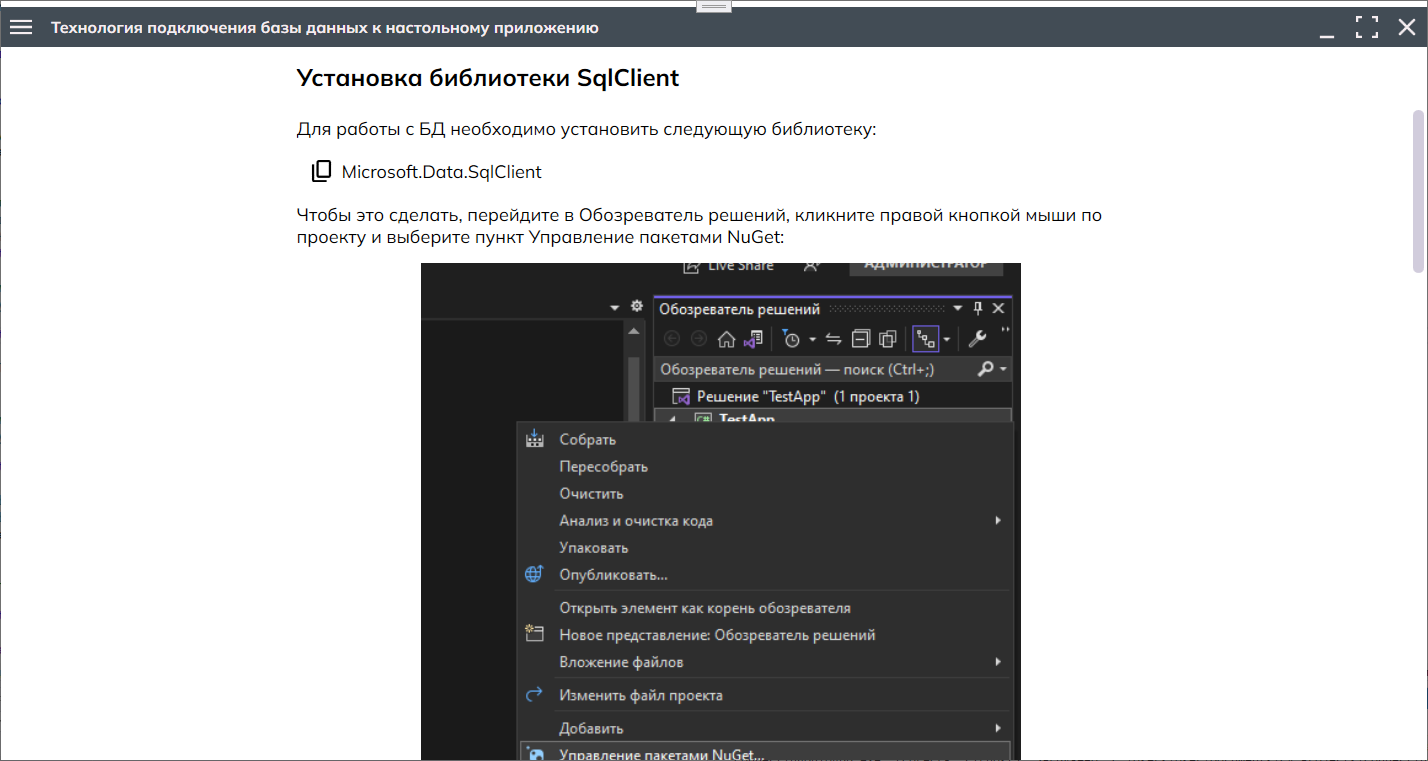


Окно содержит текстовый блок с сообщением и кнопку «ОК», закрывающей окно.

Также это окно может всплыть на начальной странице теста. Если при клике по кнопке «ПРОЙТИ» поле ввода окажется пустым, либо в него будет передано слишком большое имя, то отобразится всплывающее окно с соответствующим сообщением:

Каждый текстовый блок, изображение, каждая кнопка и гиперссылка являются элементами таких страниц как SSMS и SQLite:



Информация об этих элементах хранится в базе данных, а в программе они реализованы в виде объектов таких классов как SQLiteElement и SSMSElement. Каждый из этих классов наследуются от интерфейса ElementItem. А он в свою очередь описывает модель элемента, который хранит такие свойства как тип элемента, текст элемента, дополнительный текст, код.

Для того, чтобы к элементам страницы применялись разные стили, был создан специальный класс CustomStyleSelector. Он предназначен для выбора стиля в соответствии с типом элемента и его последующее применение к этому элементу.

Некоторыми элементами пользовательского интерфейса было необходимо было управлять из других классов. Для этого и был реализован такой класс как GlobalObjs. Он предназначен для хранения пользовательских элементов глобально. Это сделано для того, чтобы редактировать и управлять состоянием элементов из любого места программы.

Для взаимодействия с базой данных был сгенерирован особый класс – DataContext. Он представляет собой контекст базы данных. С его помощью реализуются все таблицы базы данных, настраиваются свойства определённых столбцов и осуществляется соединение с помощью строки подключения.

Объект контекста базы данных хранится в классе DbContext в единственном свойстве Tables. Это свойство отвечает за хранения контекста базы данных, за добавление, редактирование и удаление данных.

Таким образом программное приложение было спроектировано и разработано из множества классов и страниц, взаимодействующих между собой. Чтобы удостоверится в правильной работе программы, следующим шагом необходимо протестировать разработанное программное приложение.