**Государственное бюджетное профессиональное образовательное учреждение**

**«Нижегородский Губернский колледж»**

Методическая комиссия «Информатика и вычислительная техника»

Допущен к защите: преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Мухин

\_\_\_\_\_\_\_\_\_\_\_\_ Ю.С. Мамшева

\_\_\_\_\_\_\_\_\_\_\_\_ Л.В. Мухина

«21»\_\_марта\_\_2023г.

**ОТЧЕТ** **ПО**

**УЧЕБНОЙ ПРАКТИКЕ**

ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

Руководитель: \_\_\_\_\_\_\_\_\_\_ Мухин Н.А. 21.03.2023г.

\_\_\_\_\_\_\_\_\_\_ Мамшева Ю.С. 21.03.2023г.

\_\_\_\_\_\_\_\_\_\_ Мухина Л.В. 21.03.2023г.

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_ Малев А.В. 21.03.2023г.

Специальность, группа: 09.02.07, 41П

Нижний Новгород

2023г.

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc130234365)

[1. МДК 01.01 Разработка программных модулей 5](#_Toc130234366)

[1.2 Создание баз данных из скрипта 5](#_Toc130234367)

[1.3 Импорт данных в БД 7](#_Toc130234368)

[1.4 Разработка WPF-приложений 12](#_Toc130234369)

[2. МДК 01.02 Поддержка и тестирование программных модулей 20](#_Toc130234370)

[2.3 Сессия 1 20](#_Toc130234371)

[2.2.1 Создание базы данных 20](#_Toc130234372)

[2.2.2 Создание импорт данных 21](#_Toc130234373)

[2.2.3 Разработка WPF-приложений 22](#_Toc130234374)

[2.3 Сессия 2 28](#_Toc130234375)

[2.3.1 Создание базы данных 28](#_Toc130234376)

[2.3.2 Создание импорт данных 30](#_Toc130234377)

[2.3.3 Разработка WPF-приложений 40](#_Toc130234378)

[3. МДК 01.03 Разработка мобильных приложений 45](#_Toc130234379)

[Заключение 49](#_Toc130234380)

[Список литератур 50](#_Toc130234381)

[Приложения 51](#_Toc130234382)

# Введение

**Цель и задачи учебной практики.**

Цель учебной практики заключается в практическом применении теоретических знаний, полученных студентами. Учебная практика предназначена для ознакомления в производственных условиях с технологическими процессами на предприятии, а также для подготовки студентов к дальнейшему изучению дисциплин учебного плана специальности.

**Задачи учебной практики:**

1. Проанализировать предметную область разрабатываемой программы;
2. Разработка программного модуля Visual Studio 2022 WPF на языке C#;
3. Поддержка и тестирование программных модулей
4. Разработка мобильных приложений
5. Разработка библиотеки классов;

Любые персональные компьютеры, вычислительные системы, сети, мобильные устройства работают под управлением установленного на них программного обеспечения и служат для решения прикладных задач в различных сферах человеческой деятельности. Программное обеспечение — неотъемлемая часть компьютерной системы. Оно является логическим продолжением технических средств. Сфера применения конкретного компьютера определяется созданным для него программным обеспечением. Сам по себе компьютер не обладает знаниями ни в одной области применения. Все эти знания сосредоточены в выполняемых на компьютерах программах.

При создании программного обеспечения выделяются многократно используемые модули, проводится их типизация и унификация, за счет чего сокращаются сроки и трудозатраты на разработку программного продукта в целом. Некоторые программные продукты используют модули из готовых библиотек стандартных подпрограмм, процедур, функций, объектов, методов обработки данных.

Современные средства разработки программных продуктов обладают большим набором визуальных средств.

Любой программный продукт должен выполнять те функции, для которых был создан. Качественный программный продукт должен обладать еще рядом свойств, позволяющих успешно его использовать в течение продолжительного времени.

Качество программного обеспечения — совокупность его характеристик, которые определяют способность программного обеспечения удовлетворять заданные потребности пользователя. Это, однако, не означает, что разные программные продукты должны обладать одним и тем же набором свойств с одинаковыми значениями количественных показателей. Как и в случае технических устройств, показатели качества являются противоречивыми, что означает следующее: улучшение одних показателей качества может быть достигнуто за счет ухудшения других. Качество программного обеспечения является удовлетворительным, если количественные показатели свойств гарантируют успешное его использование.

Задача разработчиков программного продукта — создание качественного программного обеспечения — системного, прикладного или инструментального, обладающего требуемой функциональностью и набором технической документации.

# МДК 01.01 Разработка программных модулей

## **Создание баз данных из скрипта**

Для восстановления таблиц в созданную базы данных воспользуемся предоставленным скриптом (mssql\_script\_trade или mysql\_script\_trade в зависимости от выбранной СУБД). В процессе разработки приложения мы можем изменять базу данных на свое усмотрение. Нужно у нас два варианта: ms или my. Я выбрал mssql\_script\_trade.



Рисунок 1 – База данных

Папка можно мышь взять(копирование), а потом туда на SQL Server MS 2022.

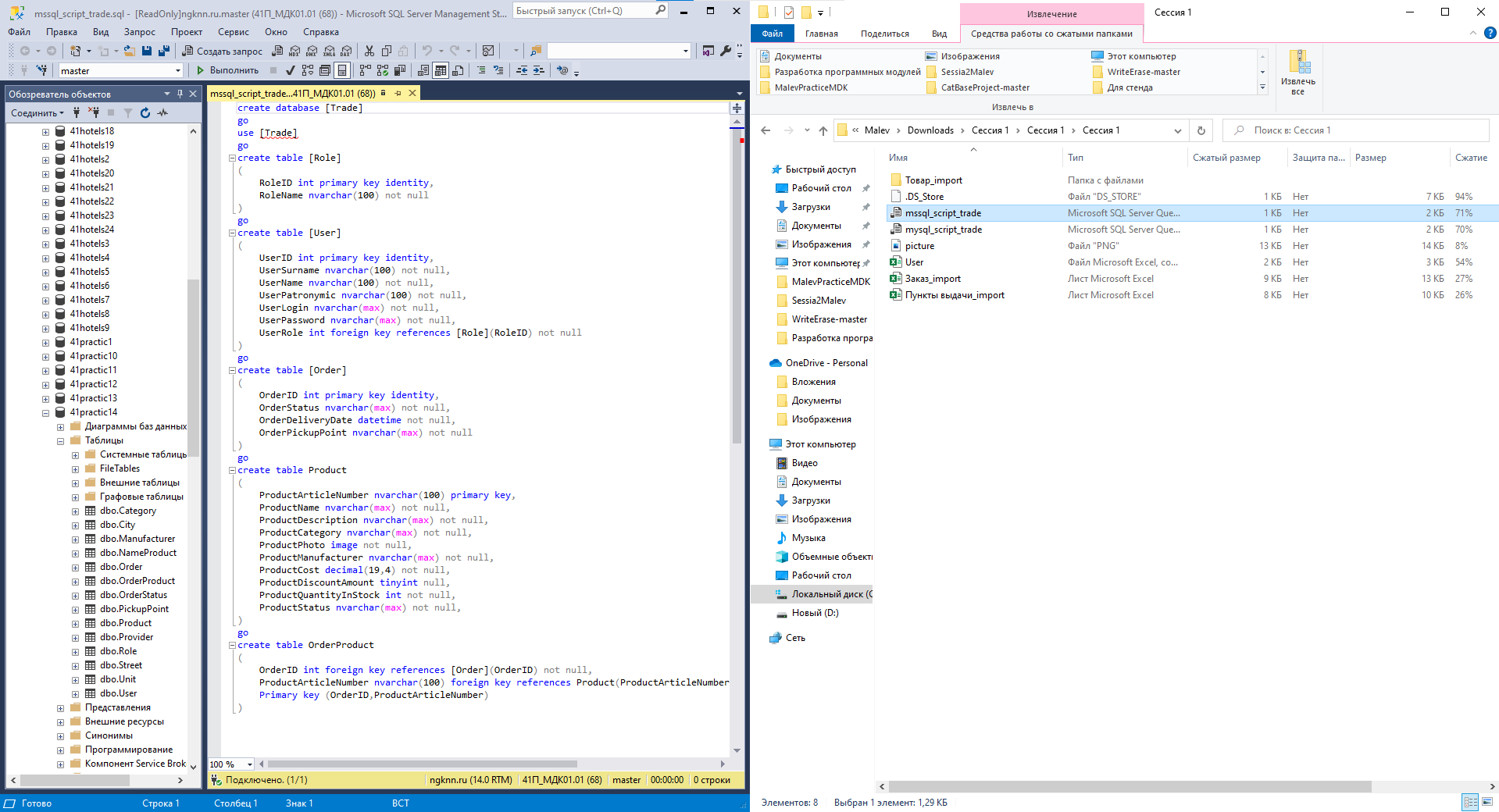


Рисунок 2 – Вставка к БД

На правую кнопку мыши нужно нажать и выбрать “Выполнить”, либо F5.

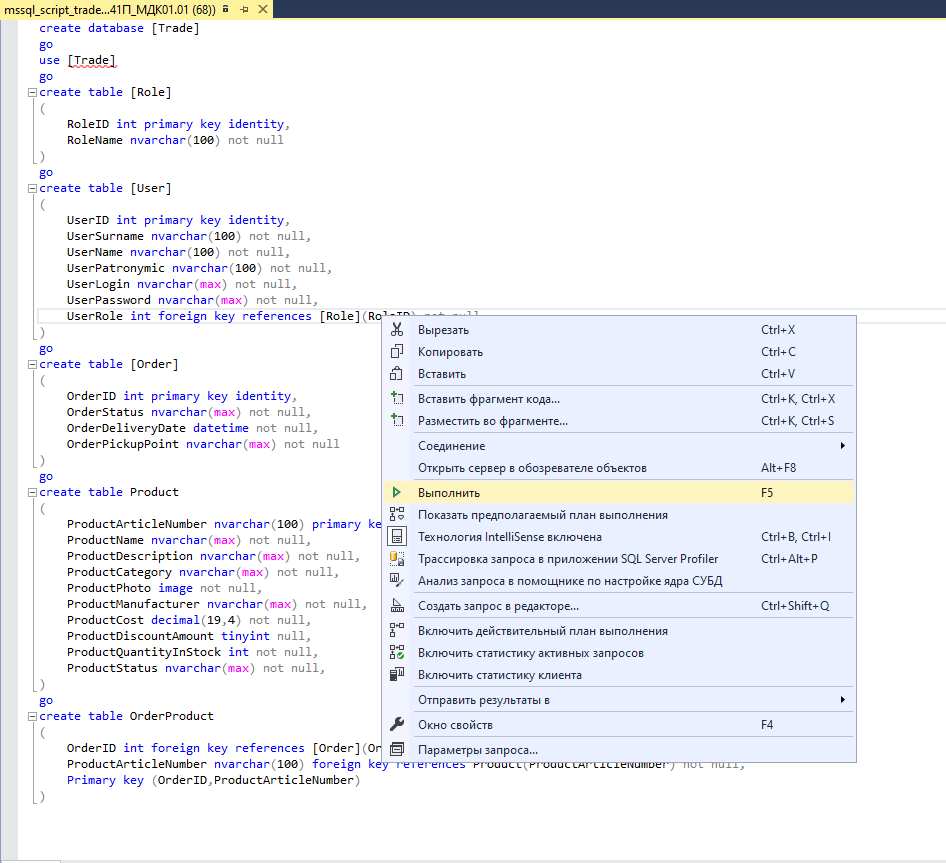


Рисунок 3 – SQL Server Management Studio 2022 «Выполнить»

У нас появились таблицы

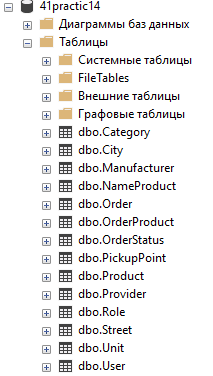


Рисунок 4 – Таблицы БД

Нужно создать диаграмму баз данных:

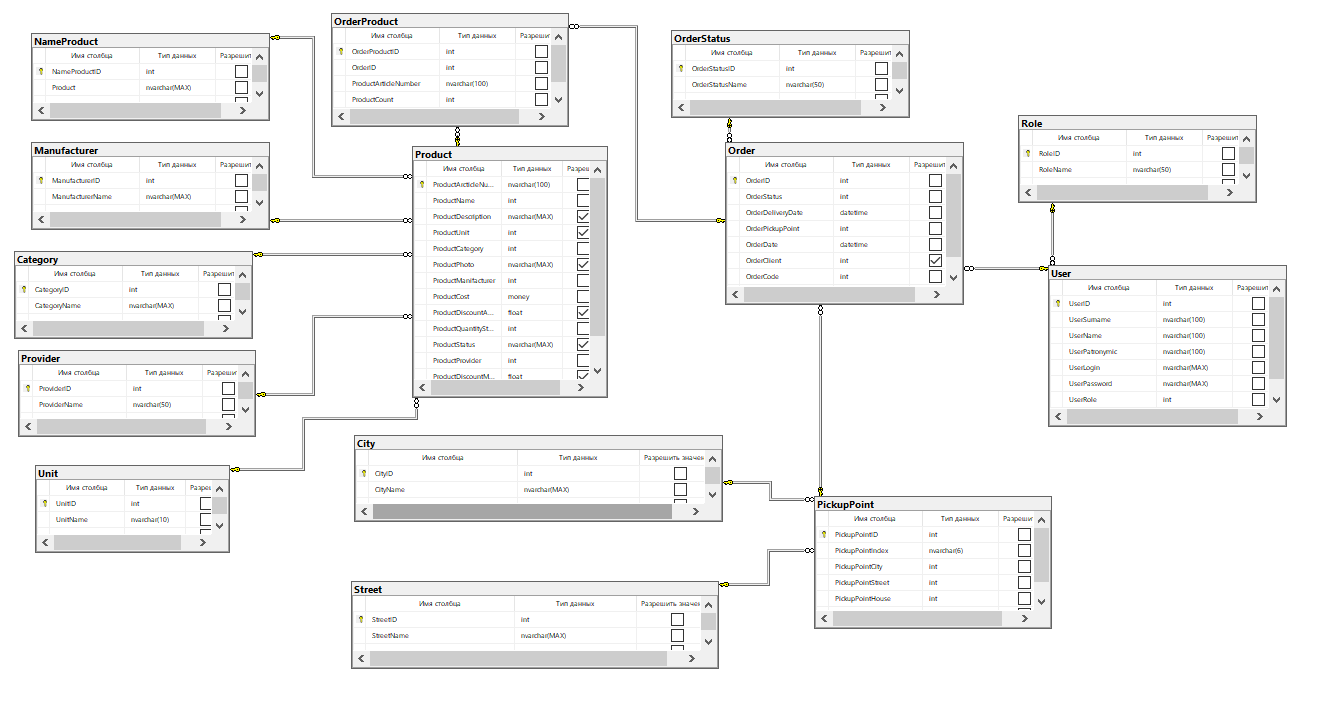


Рисунок 5 – Диаграмма БД

## **Импорт данных в БД**

Заказчик системы предоставил файлы с данными (с пометкой import в ресурсах) для переноса в новую систему. Подготовимся данные файлов для импорта и загрузите в разработанную базу данных. Папка вот это нужно брать:

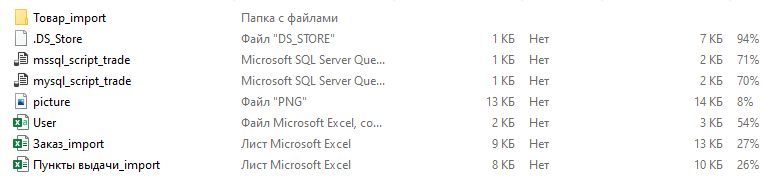


Рисунок 6 – Файлы с данными

Так нужно выбрать один из файлов и редактировать, чтобы были точно и информативно вставлены в БД. Так и во всех файлов.

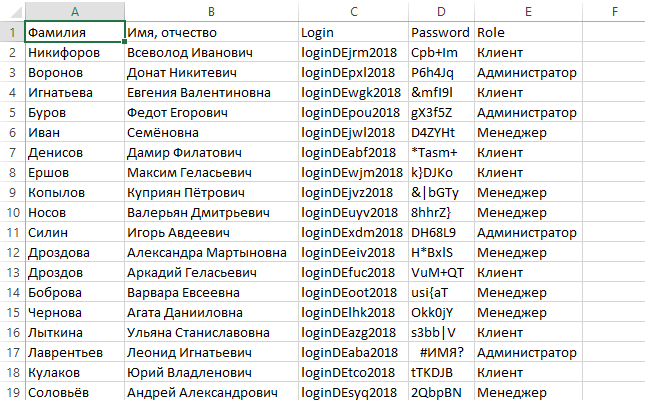


Рисунок 6 – Импорт данных в БД

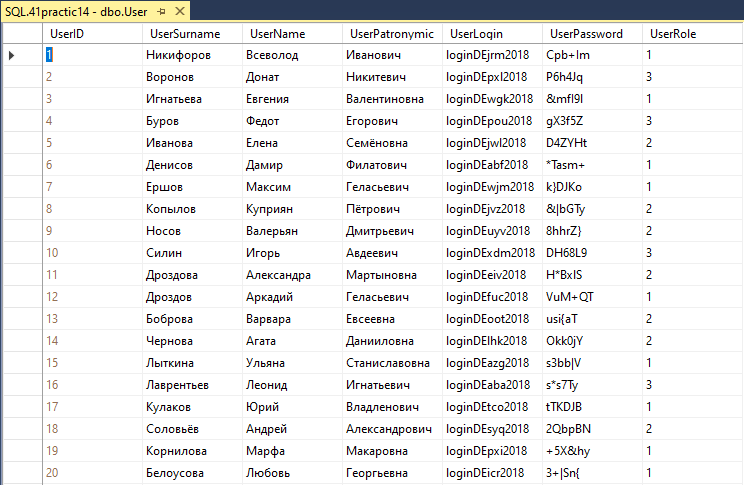


Рисунок 7 – Таблица «User»

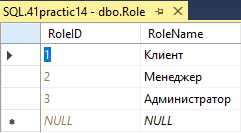


Рисунок 8 – Таблица «Role»

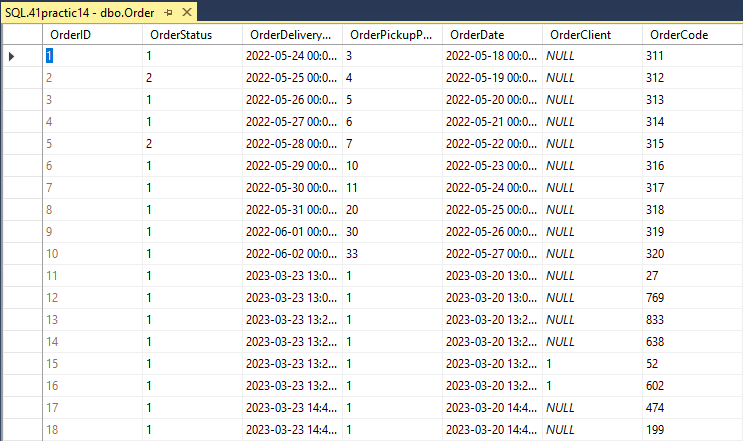


Рисунок 9 – Таблица «Order»

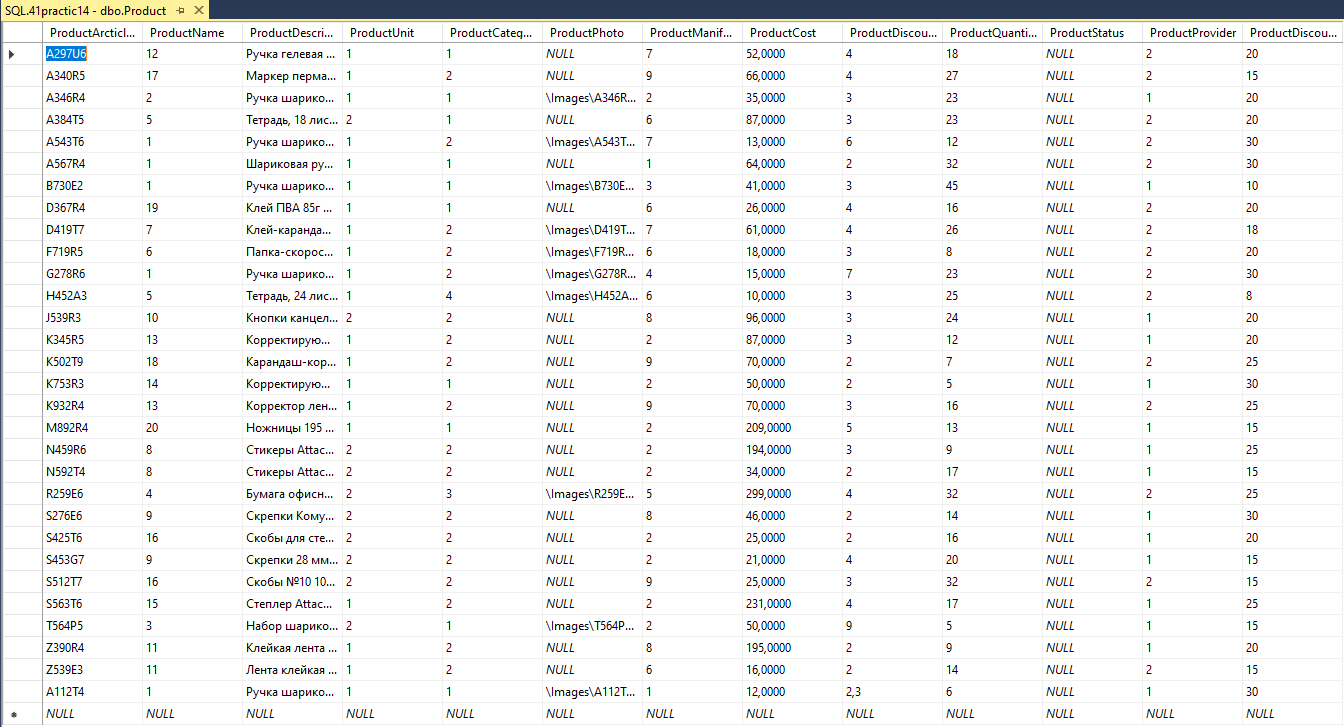
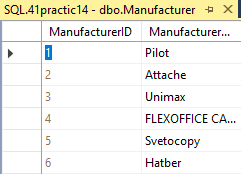


Рисунок 10 – Таблица «Product»



Рисунок 11 – Таблица «NameProduct»



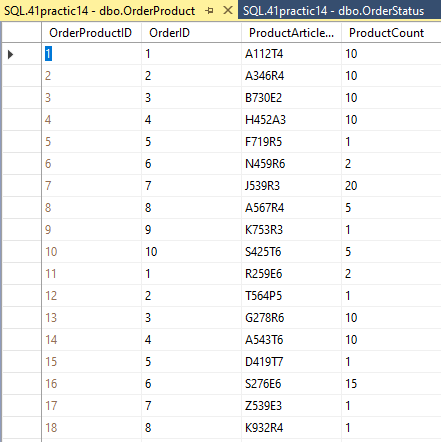


Рисунок 12 – Таблица «OrderProduct»

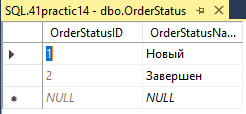


Рисунок 13 – Таблица «OrderStatus»

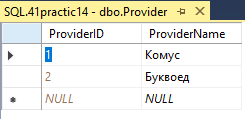


Рисунок 14 – Таблица «Provider»

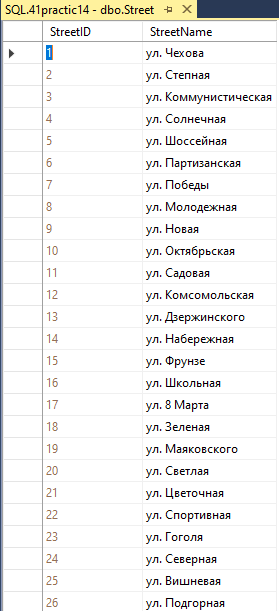


Рисунок 15 – Таблица «Street»

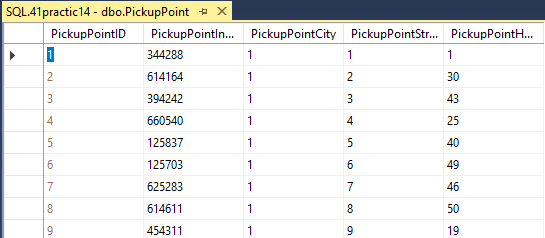


Рисунок 16 – Таблица «PickupPoint»

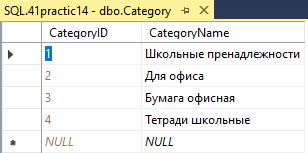


Рисунок 17 – Таблица «Category»

1. Разработка WPF-приложений

Разработать WPF-приложений для всех пользователей. Для этого нужно сначала создать проект и написать коды. Перед разработкой необходимо подключить к базу данных в WPF.

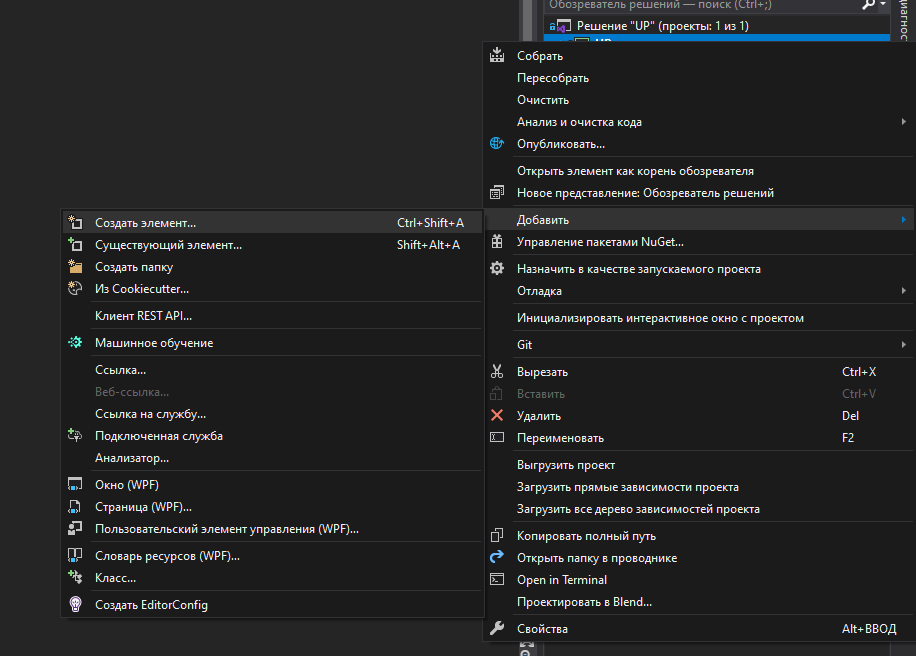


Рисунок 18 – Кнопки «Добавить» и «Список элемнтов»

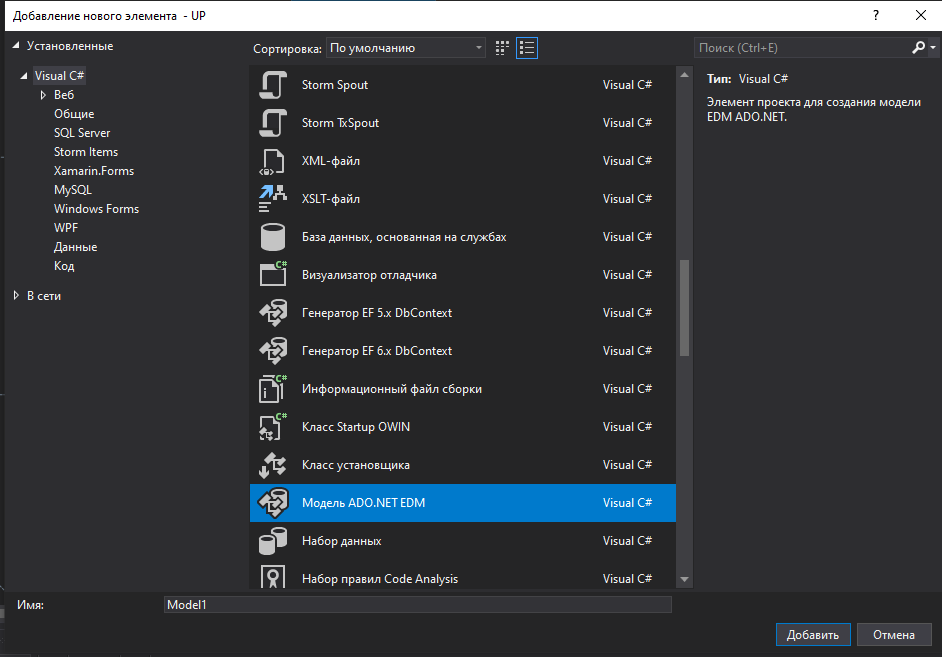


Рисунок 18 – Выбор «МОДЕЛЬ ADO.NET EDM»

Добавить конструктор EF из базы данных:

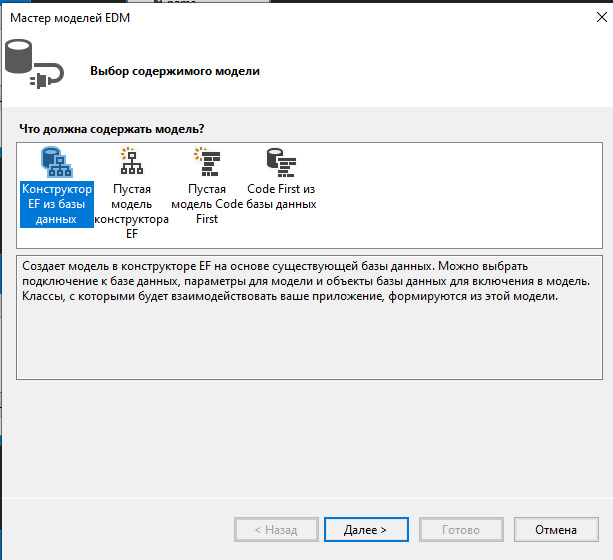


Рисунок 19 – Visual Studio 2022 WPF C# подключение к БД

Введите имя сервера свой колледж:

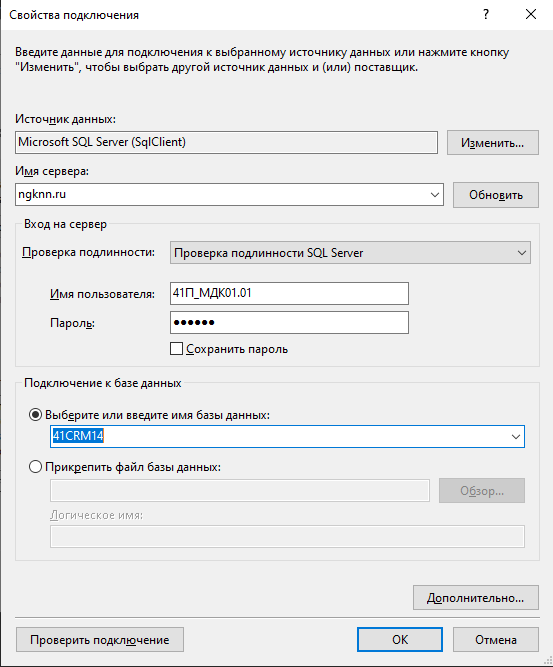


Рисунок 20 – Visual Studio 2022 WPF C# подключение к БД

После нажмите ОК.

У нас Visual Studio 2022 появились в БД:

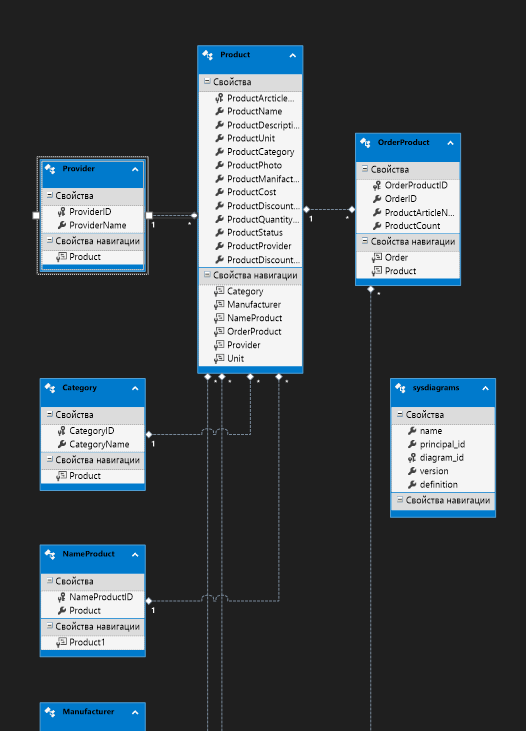


Рисунок 21 – Диаграмма в WPF

После этого можно разработать и написать коды. Затем нажать на кнопку «Пуск», чтобы запустить проект. Весь проект нужно выполнить по руководству по стилю (см в рис. 22).

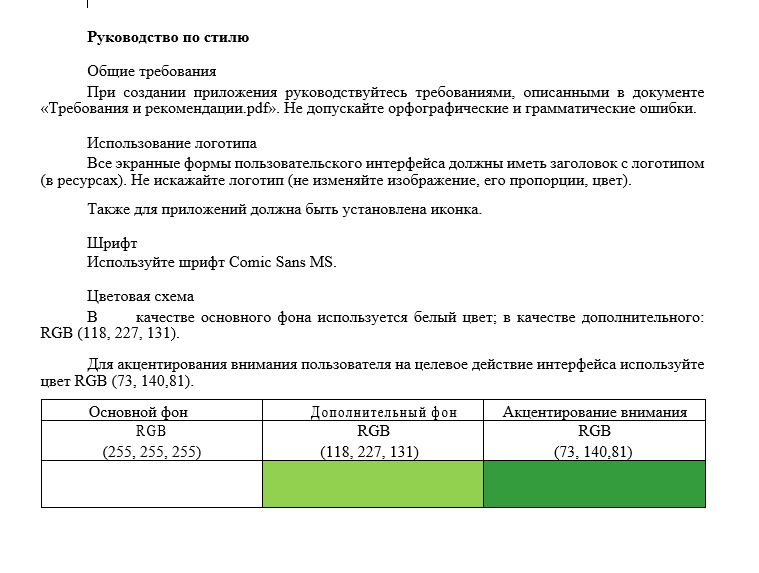


Рисунок 22 – Руководство по стилю

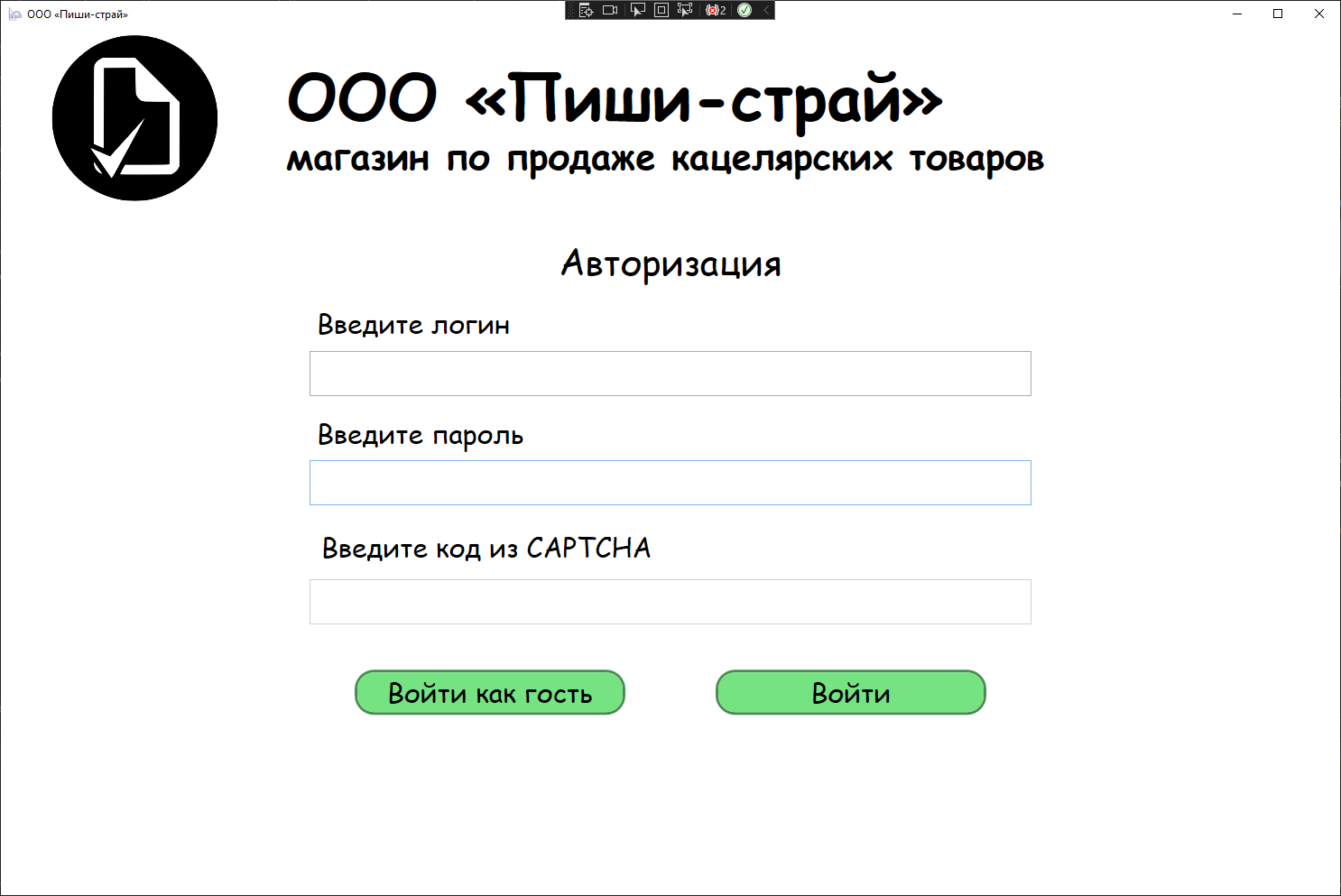


Рисунок 23 – Форма «Авторизация»

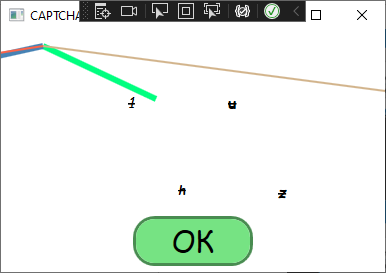


Рисунок 24 – Форма «CAPTHA»

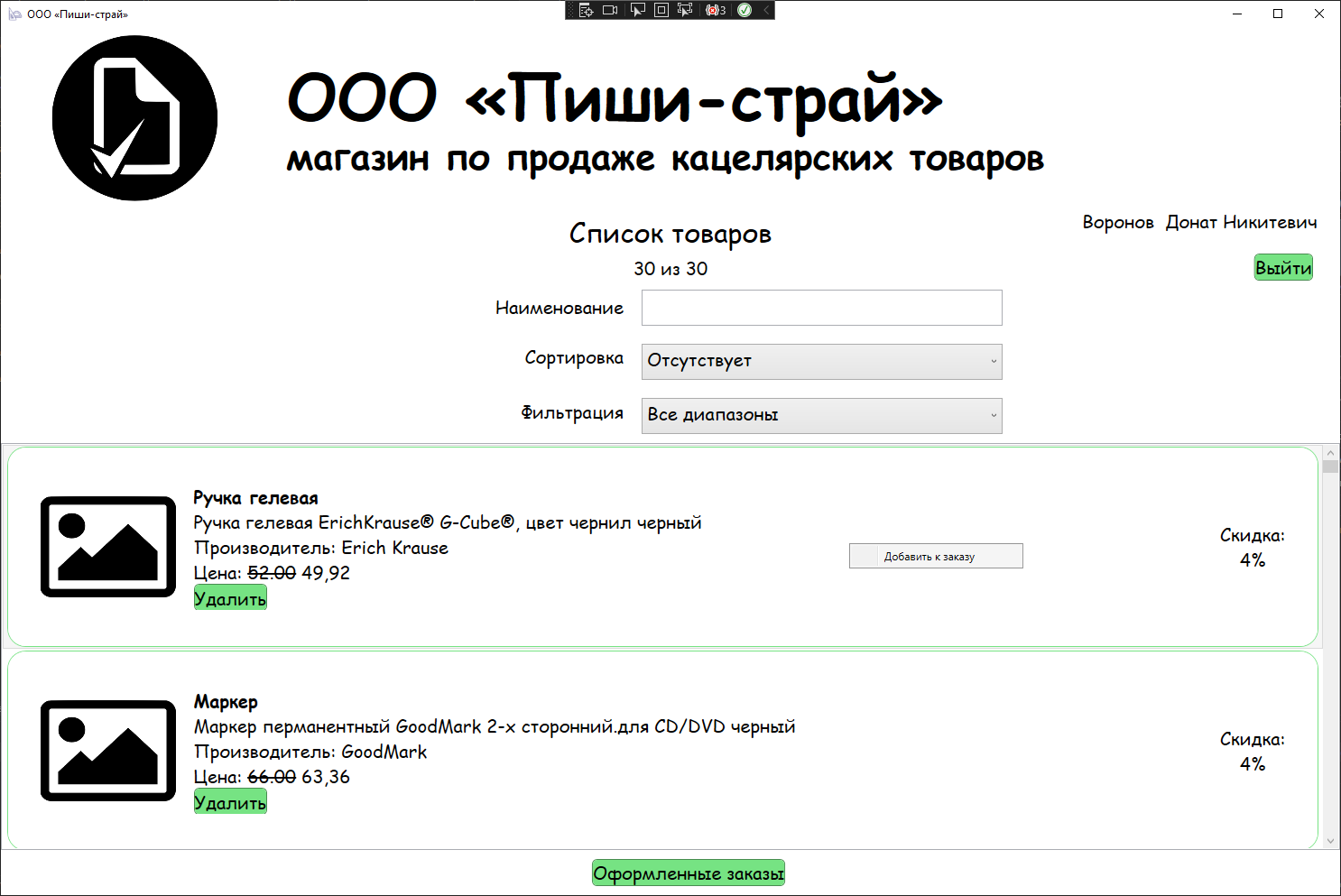


Рисунок 25 – Форма «Список товаров»

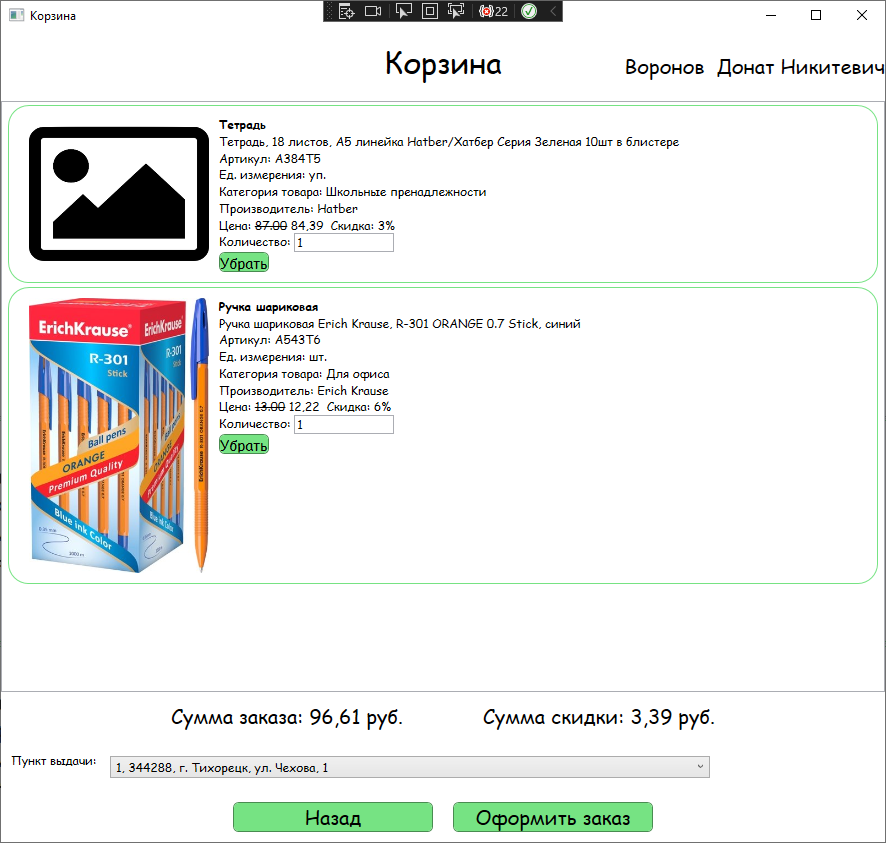


Рисунок 25 – Форма «Корзина»

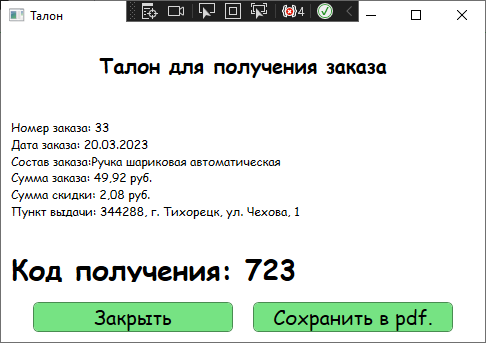


Рисунок 26 – Форма «Талон»

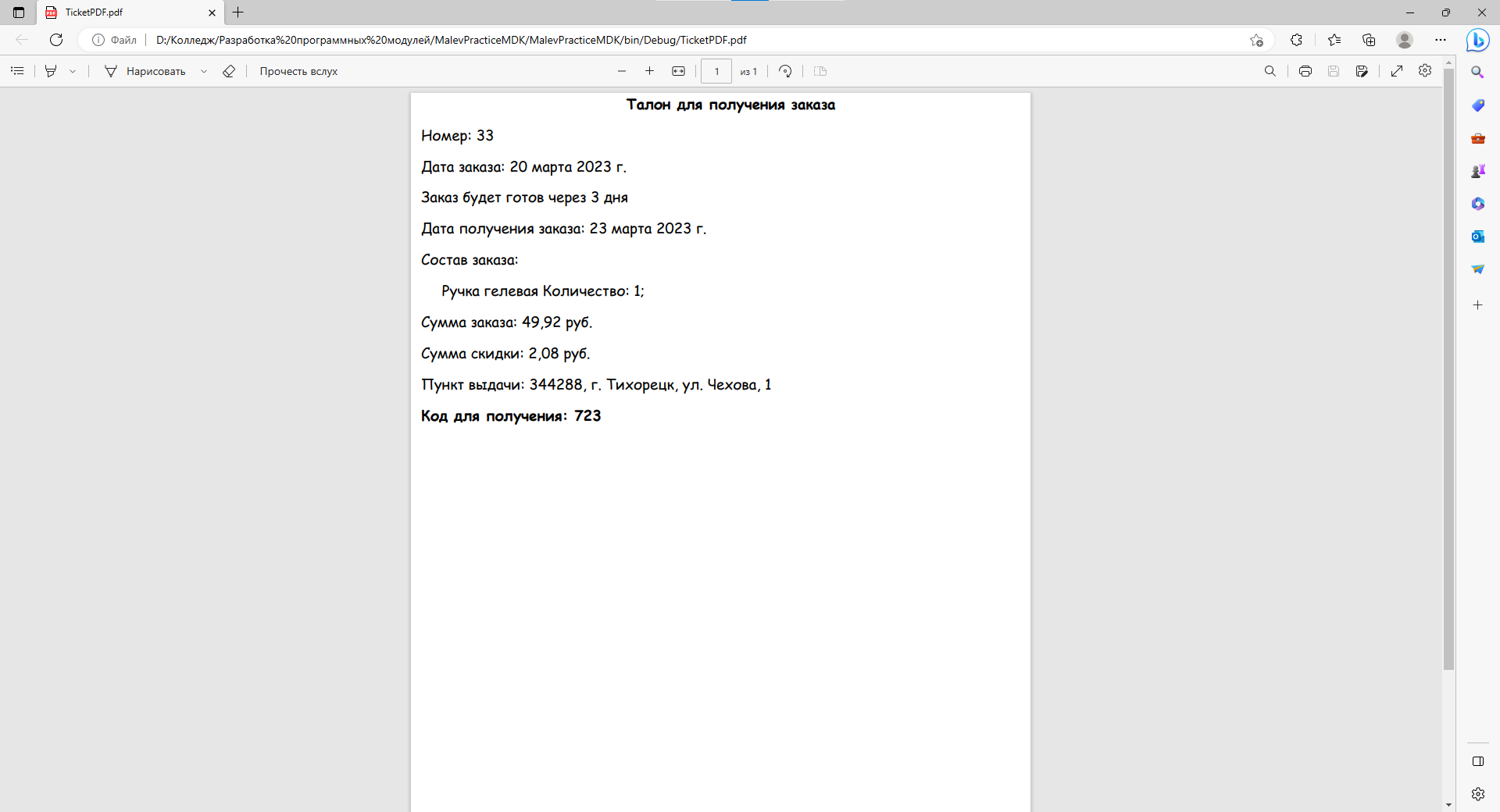


Рисунок 27 – Форма «Талон в формате PDF»



Рисунок 27 – Форма «Оформленные заказы»

# МДК 01.02 Поддержка и тестирование программных модулей

## **Сессия 1**

### **Создание базы данных**

Создать таблицы баз данных для этого проекта.

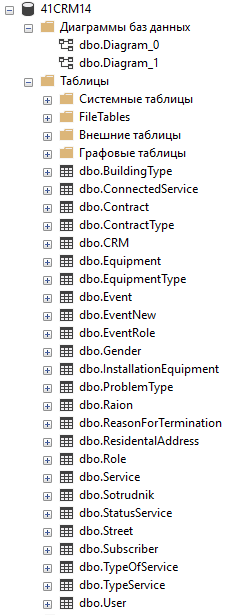


Рисунок 28 – «Создание базы данных»

Нужно создать диаграмму баз данных для Авторизации.

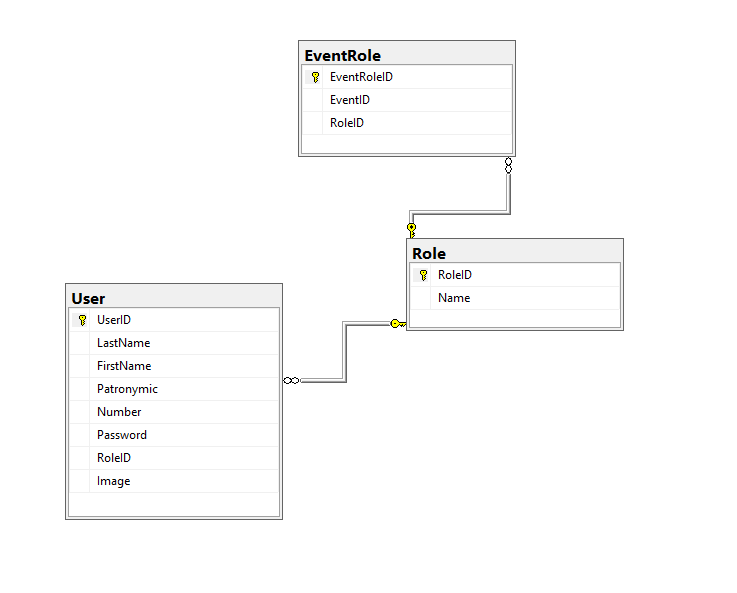
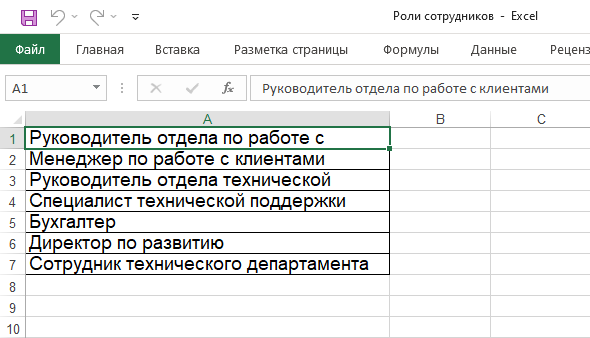


Рисунок 29 – «Диаграмма баз данных»

### **Создание импорт данных**

Заказчик системы предоставил файлы с данными (с пометкой import в ресурсах) для переноса в новую систему. Подготовьте данные файлов для импорта и загрузите в разработанную базу данных. Папка вот это нужно брать:

Рисунок 30 – Импорт данных «Роли сотрудников» в базу данных

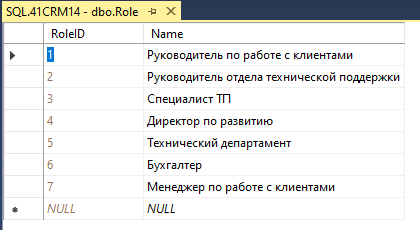


Рисунок 31 – «Таблица Role»

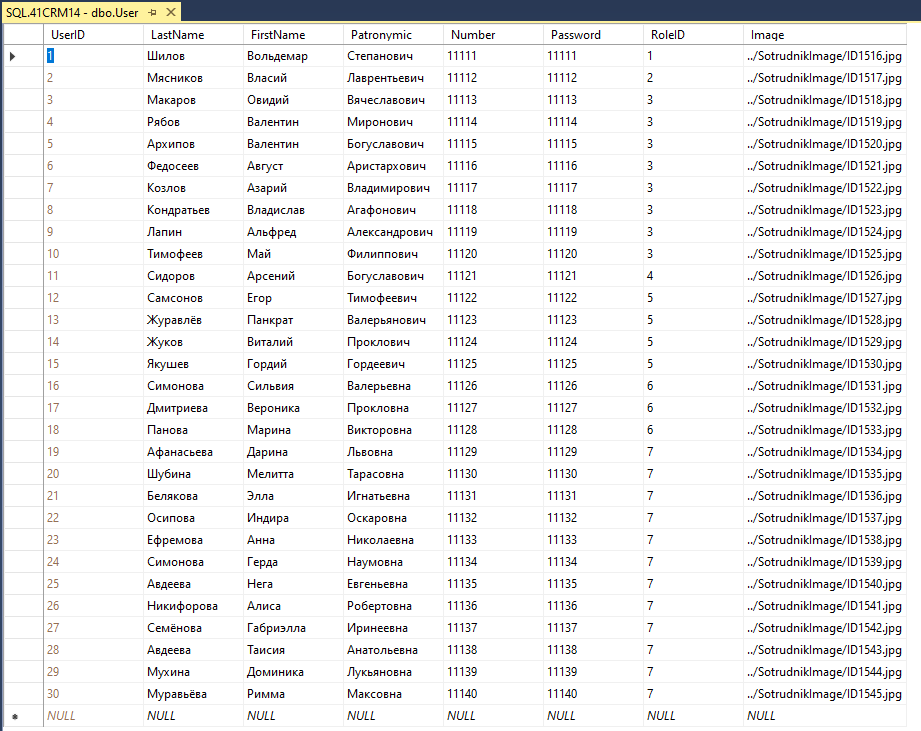


Рисунок 32 – «Таблица User»

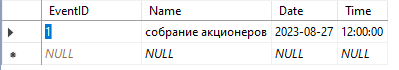


Рисунок 33 – «Таблица Event»

### **Разработка WPF-приложений**

Подключение к базе данных “Добавить -> Создать элемент…”:

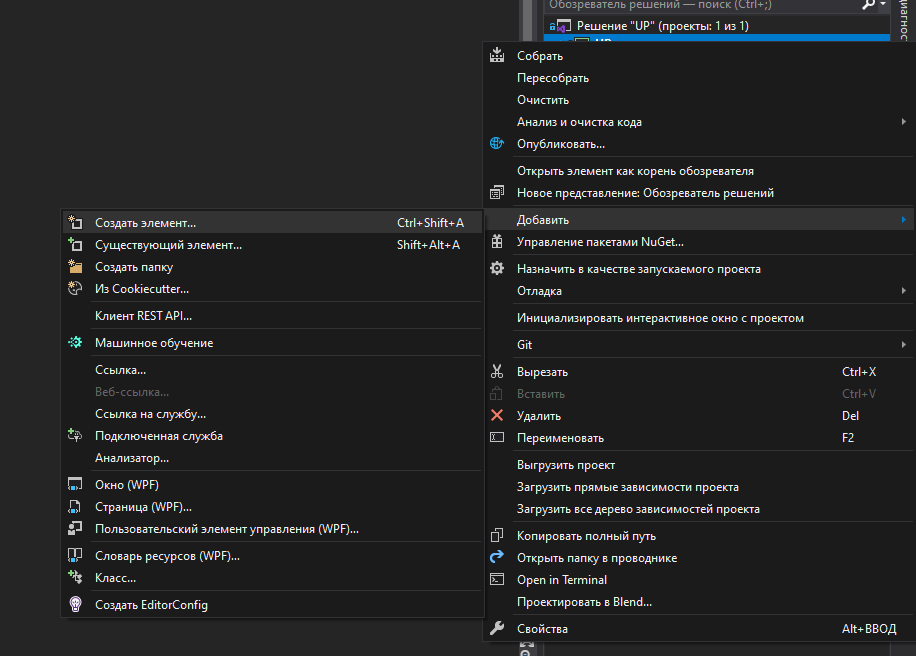


Рисунок 34 – «Visual Studio 2022 WPF C# подключение к БД»

Вот это Модель ADO.NET EDM добавить:

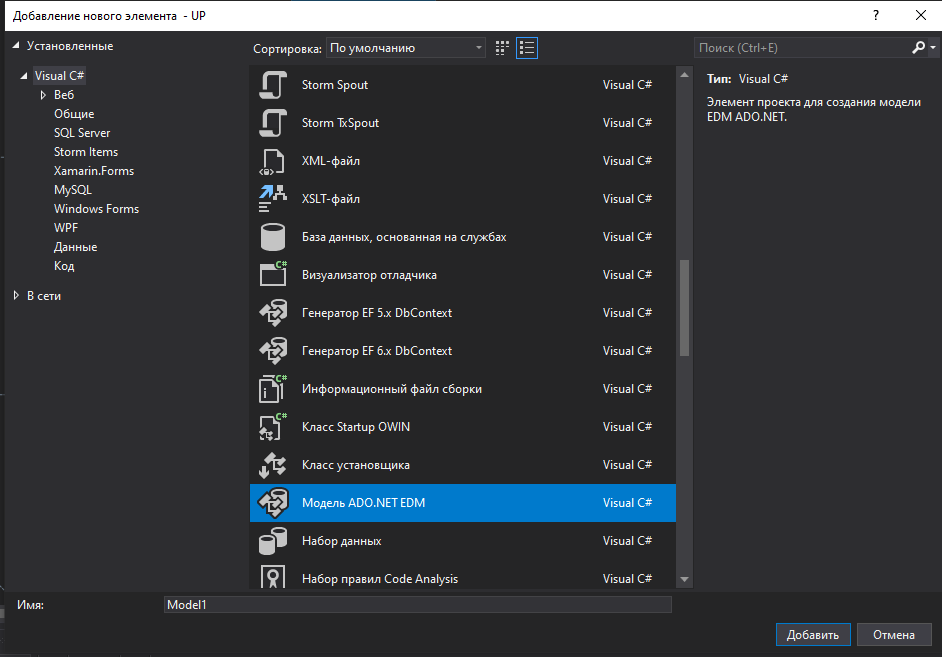


Рисунок 35 – «Visual Studio 2022 WPF C# подключение к БД»

Добавить конструктор EF из базы данных:

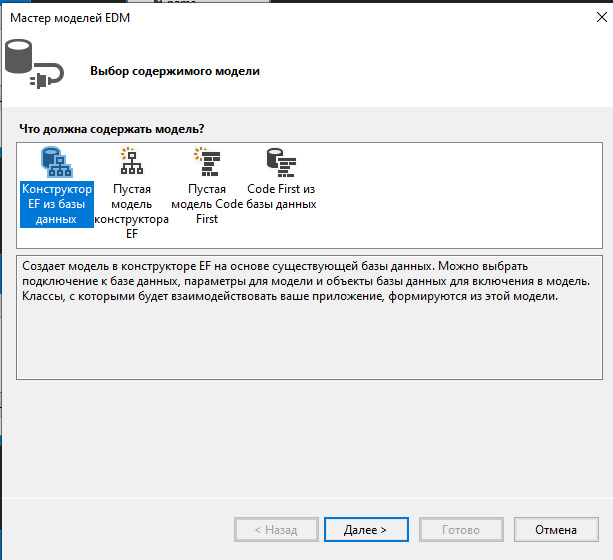


Рисунок 36 – «Visual Studio 2022 WPF C# подключение к БД»

Введите имя сервера свой колледж:

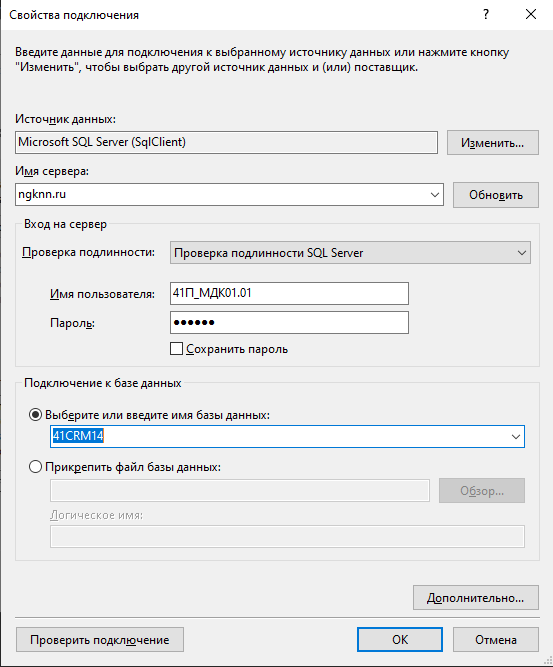


Рисунок 37 – «Visual Studio 2022 WPF C# подключение к БД»

После нажмите ОК.

У нас Visual Studio 2022 появились в БД:

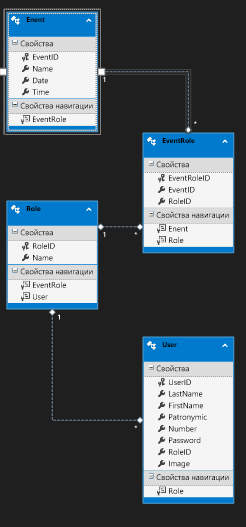


Рисунок 38 – «Visual Studio 2022 WPF C# база данных»

Запустить проект, проверить работоспособность. Так сначала нужно вводить номер пользователя. Сразу написать пароль и код невозможно. Когда вводил номер, нужно нажать на кнопку Enter на клавиатуре.

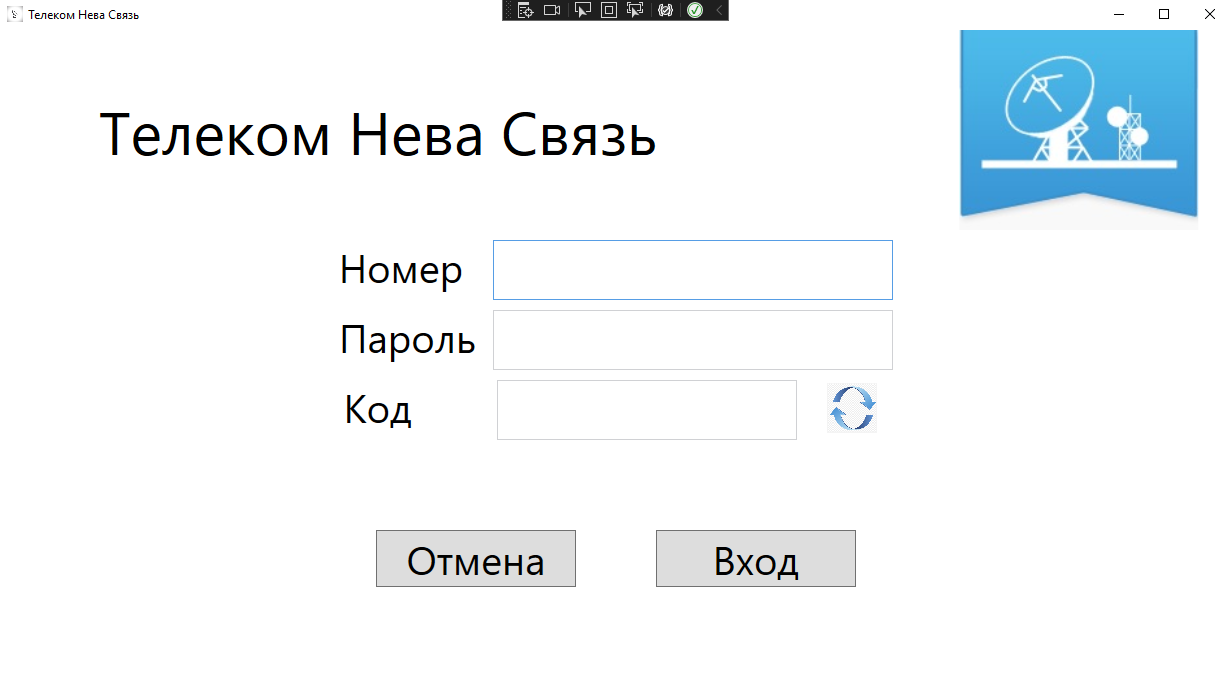


Рисунок 39 – «Форма проекта»

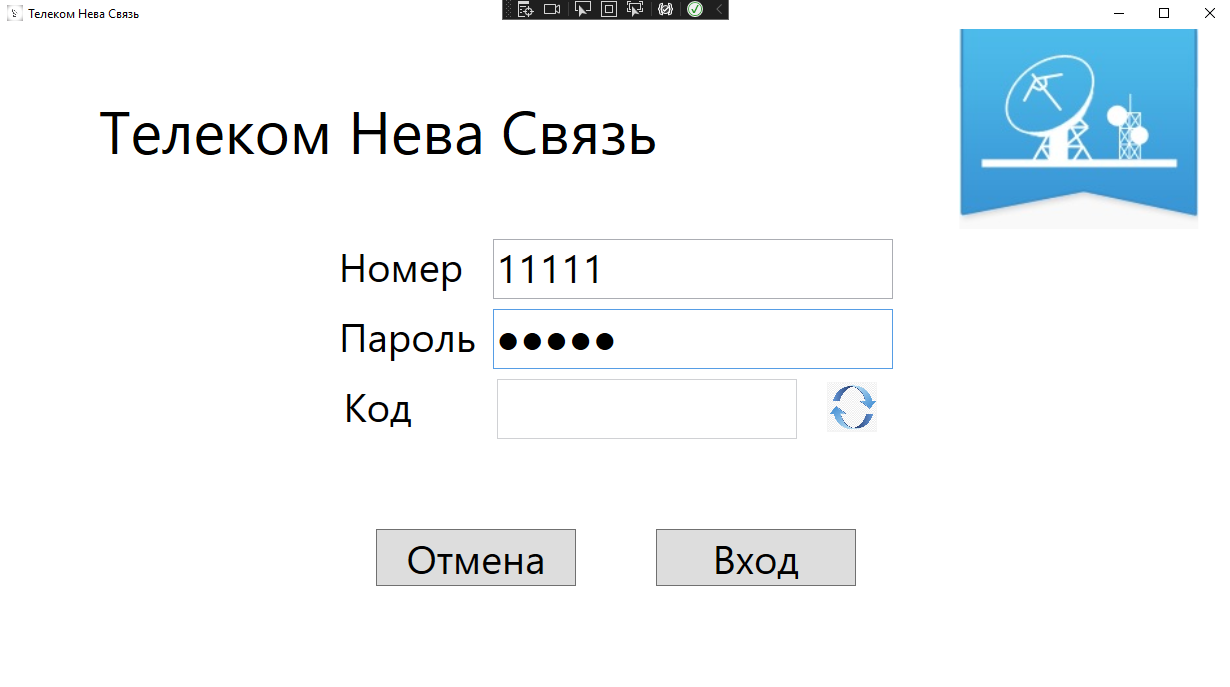


Рисунок 40 – «Форма проекта (продолжение)»

После ввода номера пишем пароль для входа. И снова нажать на кнопку Enter на клавиатуре. И выводится сообщение о сгенерированном коде для ввода.

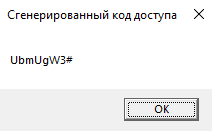


Рисунок 41 – «Сообщение о сгенерированном коде для ввода»

И после него пишем код для входа. Если неверно вводится код, тогда нужно нажать на кнопку «Обновить» в виде значка. А при кнопке «Отмена» - сбросить все данные для ввода.

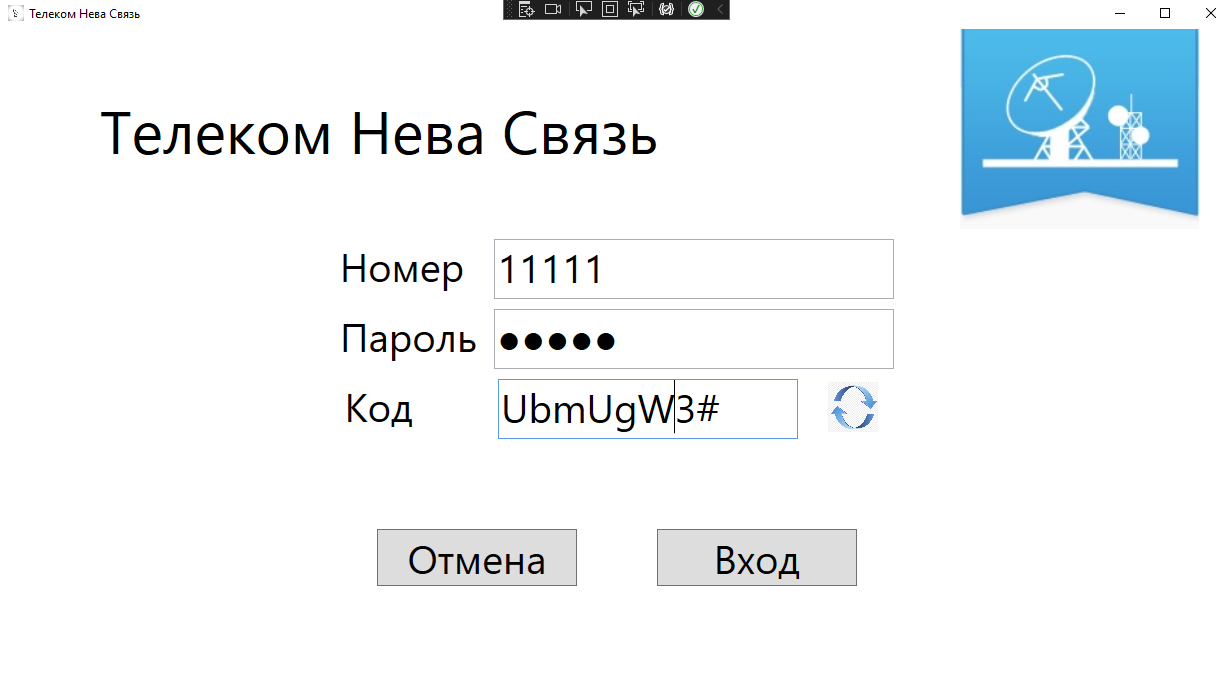


Рисунок 42 – «Ввод кода в форме проекта для входа пользователя»

Потом нажать на кнопку Вход. После входа выводится сообщение об успешном входе пользователя.

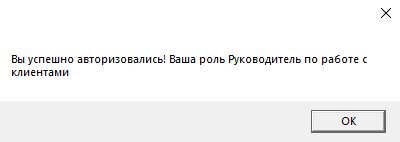


Рисунок 43 – «Сообщение об успешном входе пользователя»

## **Сессия 2**

### **Создание базы данных**

Создать таблицы баз данных для этого проекта.

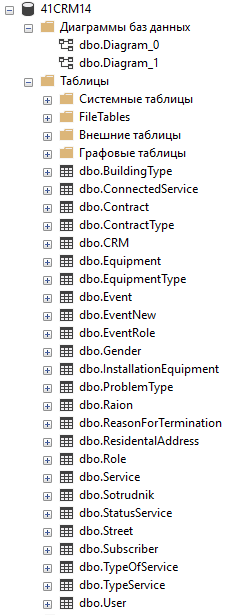


Рисунок 44 – «Создание базы данных»

Нужно создать диаграмму баз данных для сотрудника.

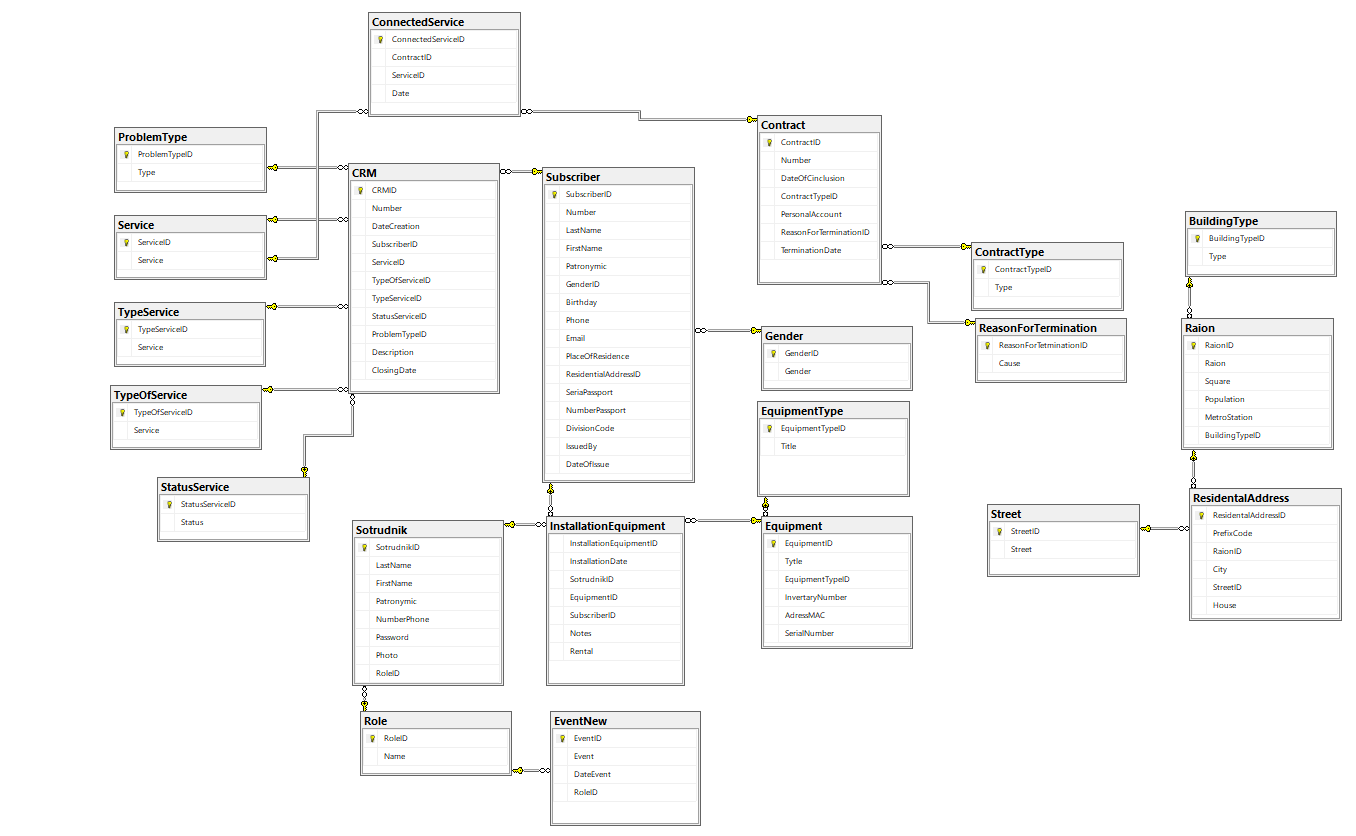


Рисунок 45 – «Диаграмма баз данных»

### **Создание импорт данных**

Заказчик системы предоставил файлы с данными (с пометкой import в ресурсах) для переноса в новую систему. Подготовьте данные файлов для импорта и загрузите в разработанную базу данных. Папка вот это нужно брать:

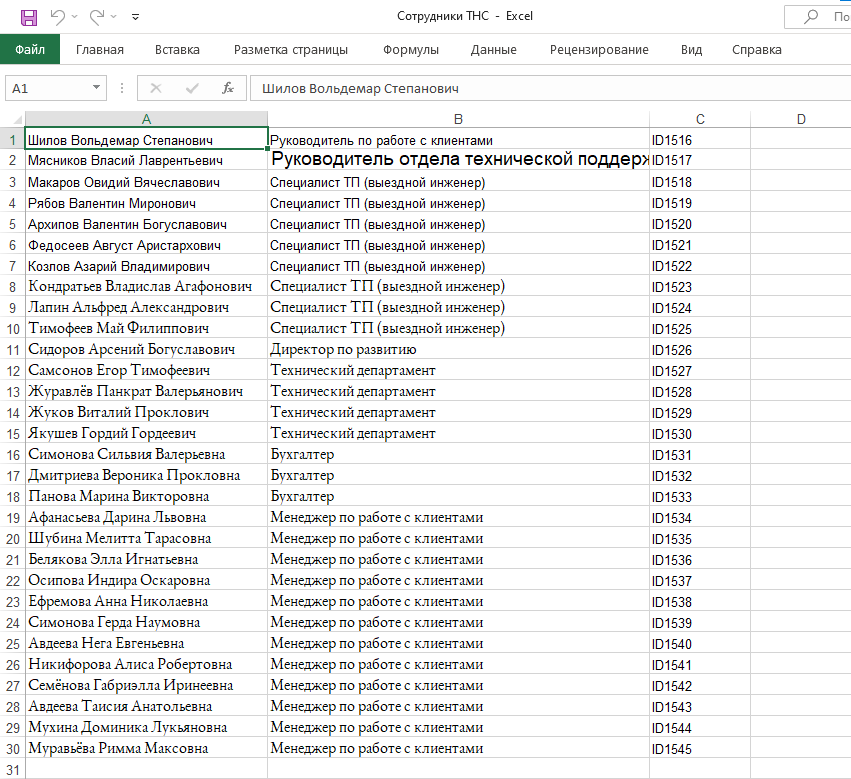


Рисунок 46 – «Импорт данных «Сотрудники ТНС» в базу данных»

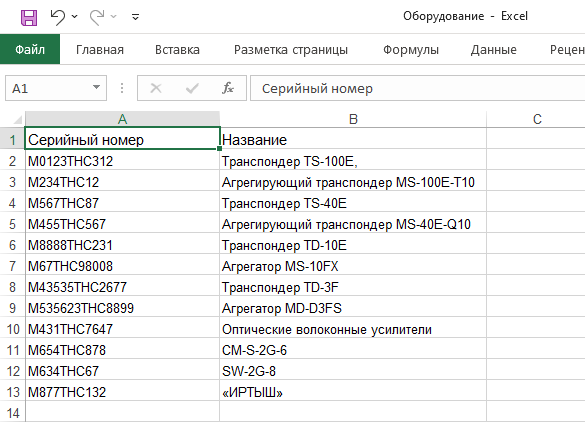


Рисунок 47 – «Импорт данных «Сотрудники ТНС» в базу данных»

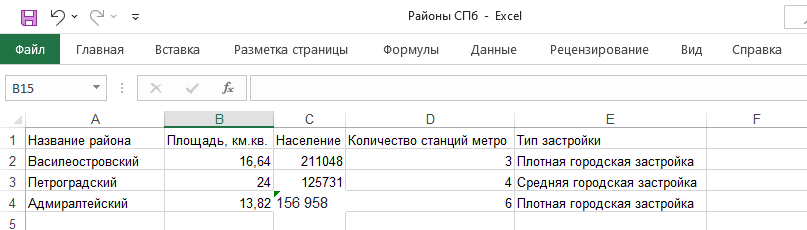


Рисунок 48 – «Импорт данных «Районы СПб» в базу данных»

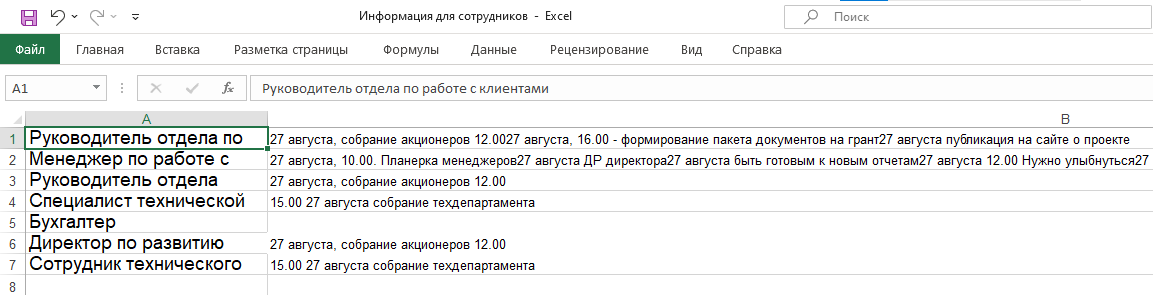


Рисунок 49 – «Импорт данных «Информация для сотрудников» в базу данных»

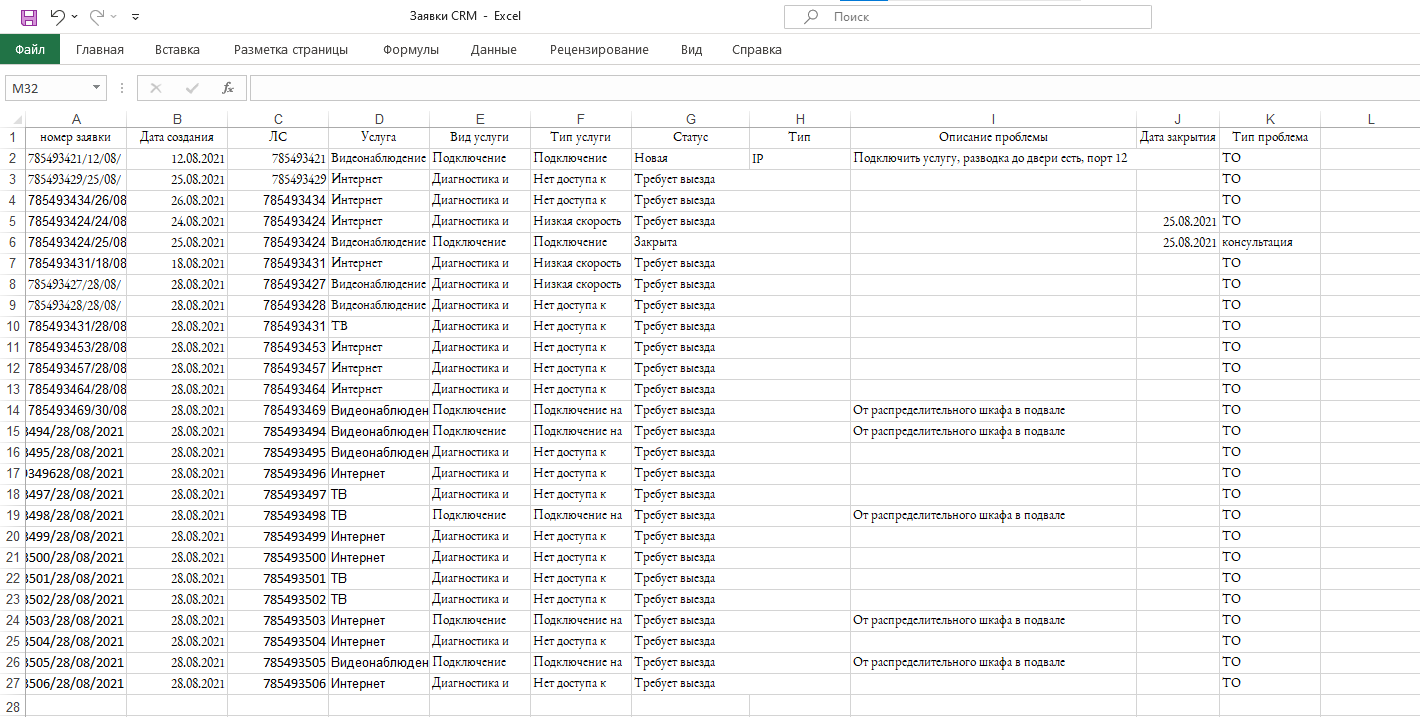


Рисунок 50 – «Импорт данных «Заявки CRM» в базу данных»

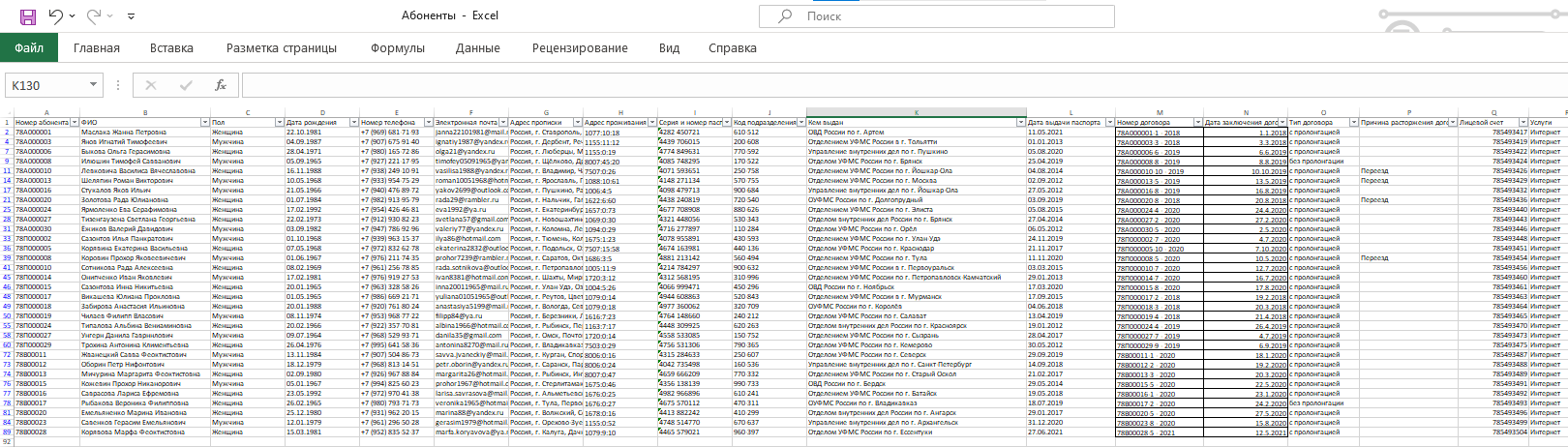


Рисунок 51 – «Импорт данных «Абоненты» в базу данных»

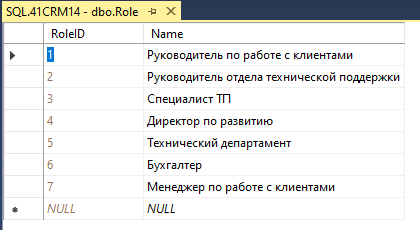


Рисунок 52 – «Таблица Role»

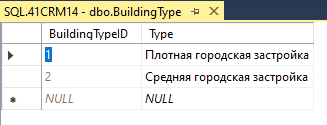


Рисунок 53 – «Таблица BuildingType»

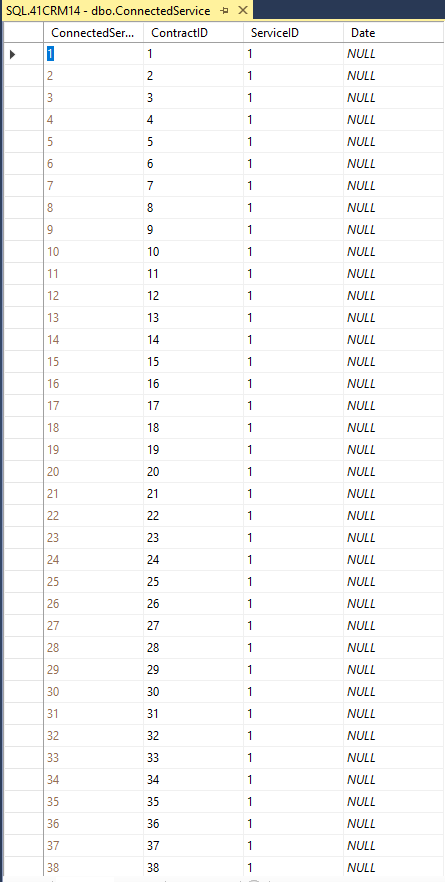


Рисунок 54 – «Таблица ConnectedService»



Рисунок 55 – «Таблица Contract»

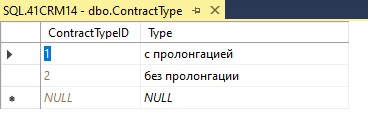


Рисунок 56 – «Таблица ContractType»

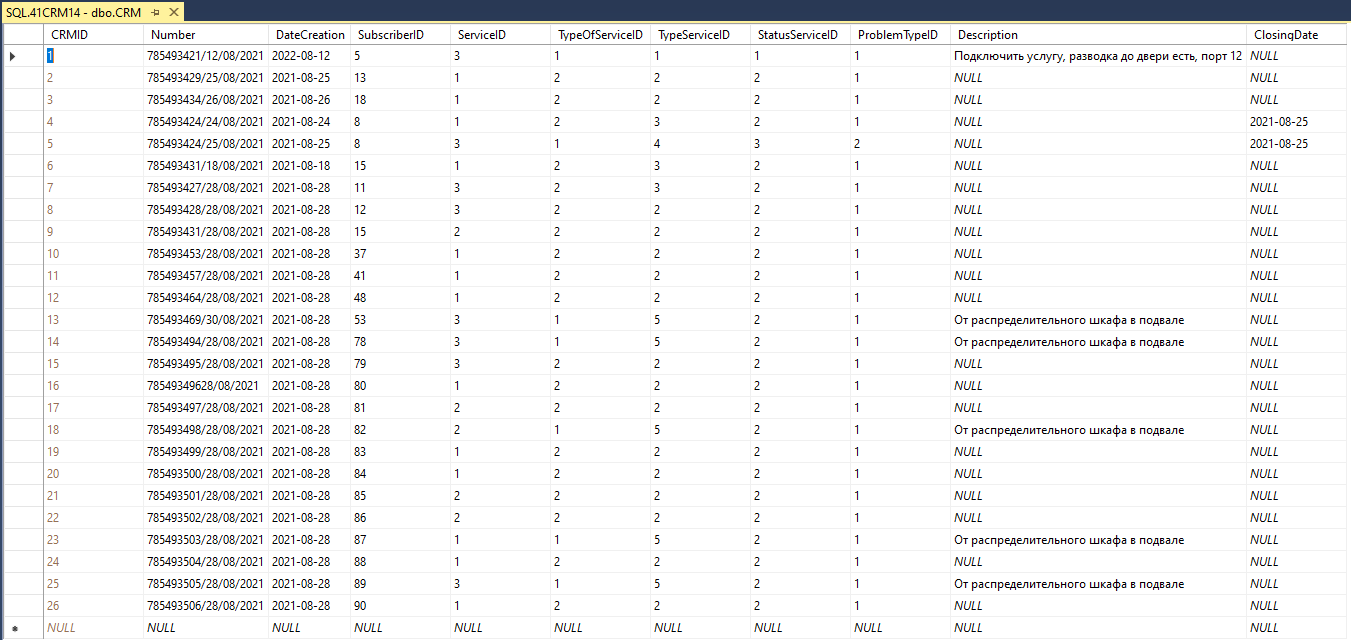


Рисунок 57 – «Таблица CRM»

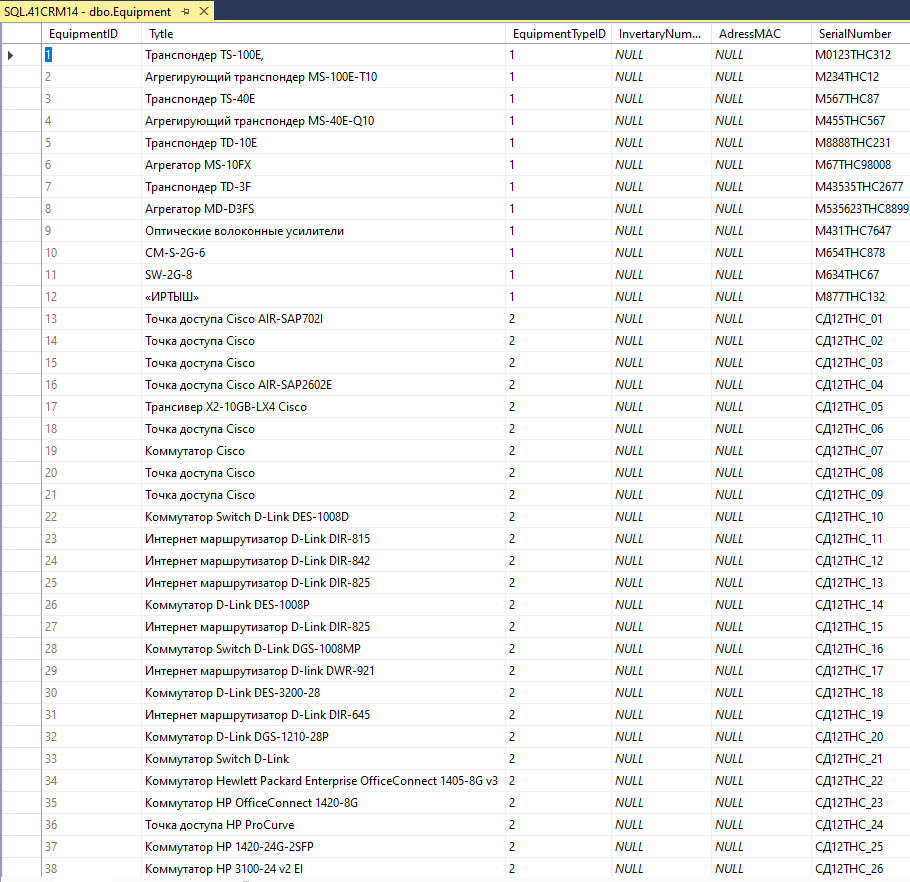


Рисунок 36 – «Таблица Equipment»

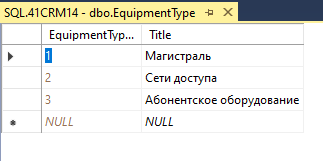


Рисунок 58 – «Таблица EquipmentType»

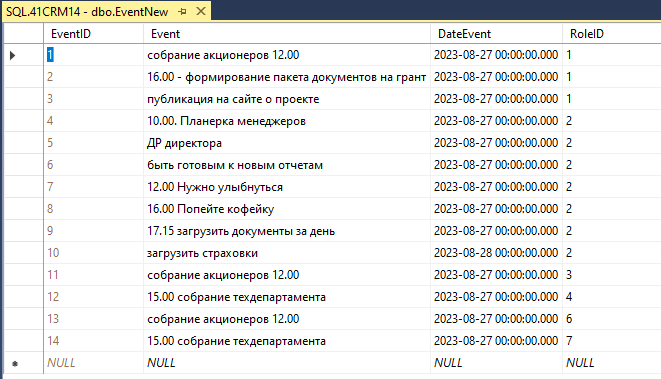


Рисунок 59 – «Таблица EventNew»

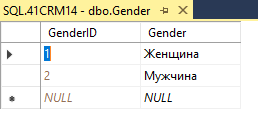


Рисунок 60 – «Таблица Gender»

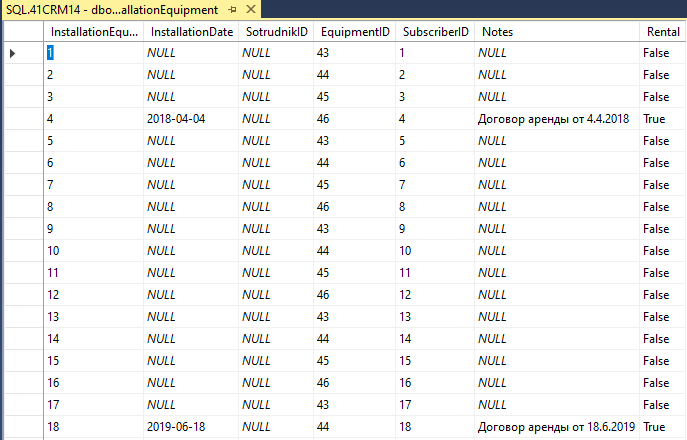


Рисунок 61 – «Таблица InstallationEquipment»

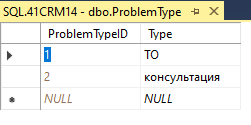


Рисунок 62 – «Таблица ProblemType»

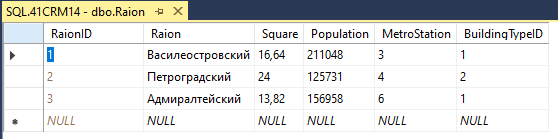


Рисунок 63 – «Таблица Raion»

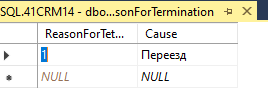


Рисунок 64 – «Таблица ReasonForTermination»

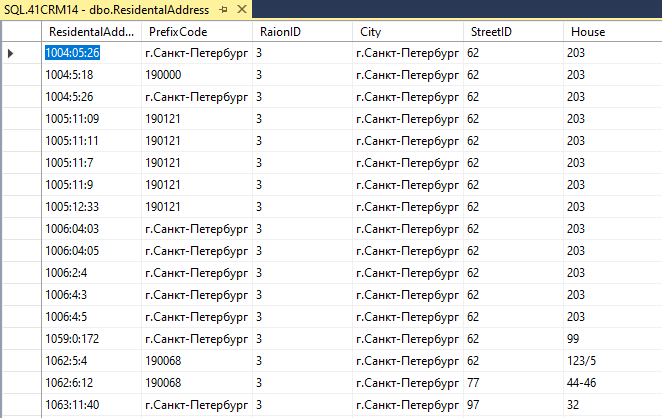


Рисунок 65 – «Таблица ResidentalAddress»

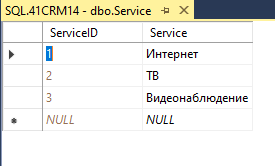


Рисунок 66 – «Таблица Service»

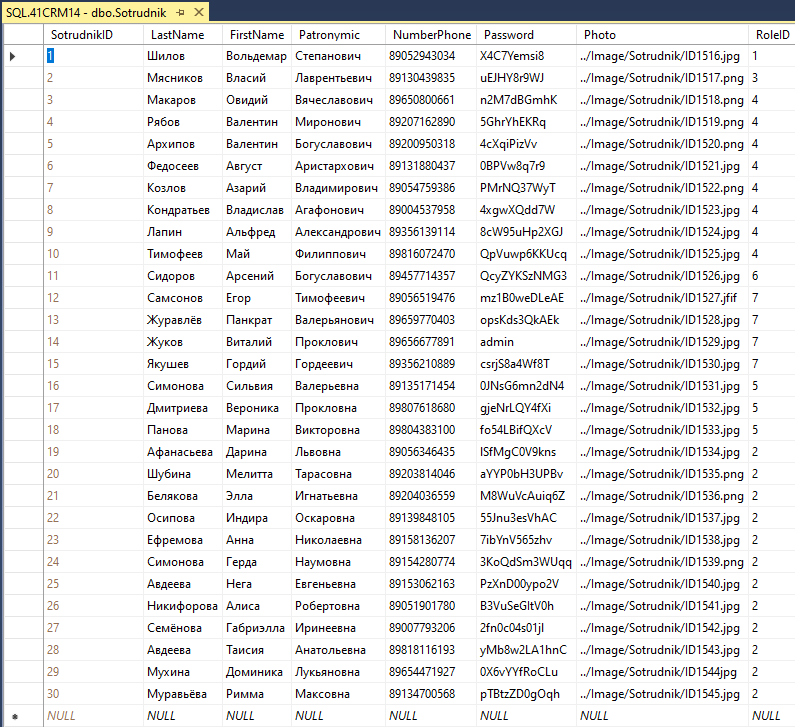


Рисунок 67 – «Таблица Sotrudnik»

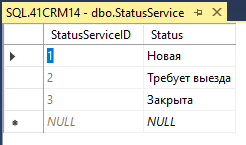


Рисунок 68 – «Таблица StatusService»

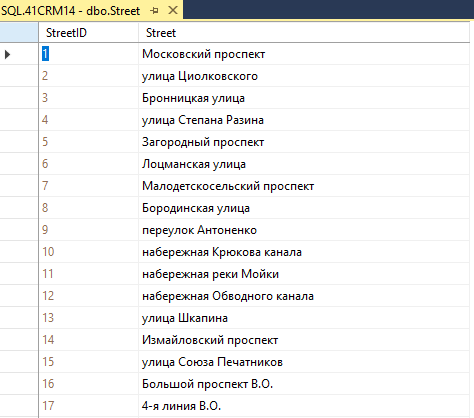


Рисунок 69 – «Таблица Street»

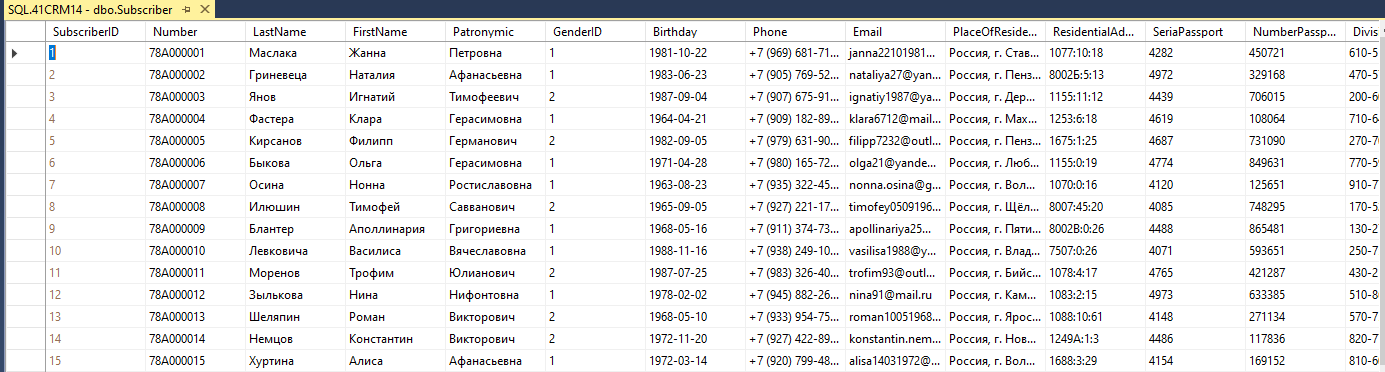


Рисунок 70 – «Таблица Subscriber»

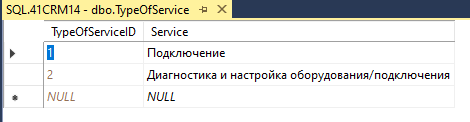


Рисунок 71 – «Таблица TypeOfService»

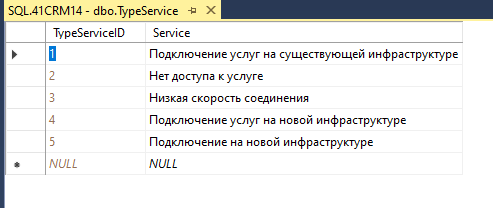


Рисунок 72 – «Таблица TypeService»

### **Разработка WPF-приложений**

Подключение к базе данных “Добавить -> Создать элемент…”:

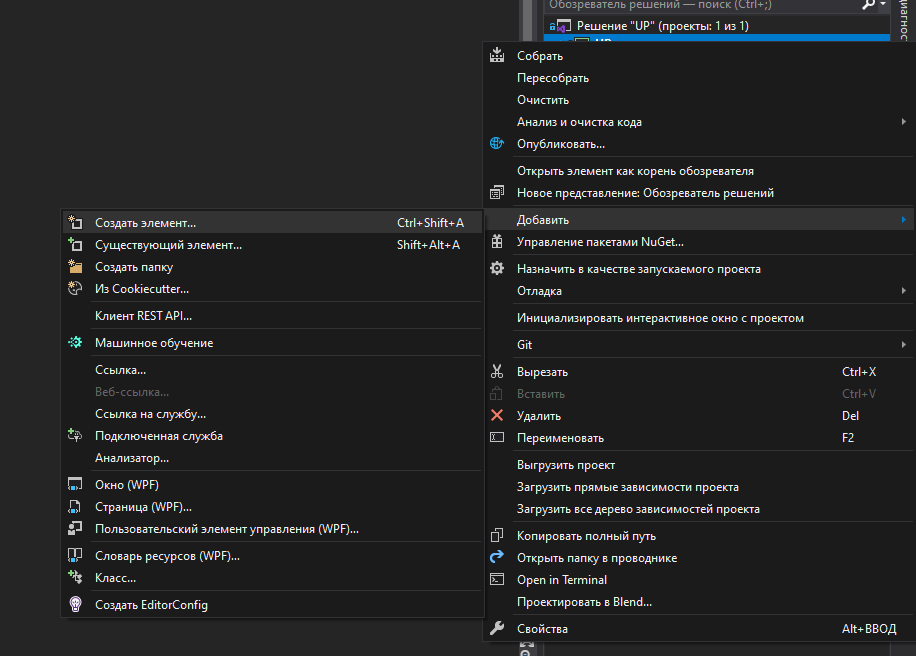


Рисунок 73 – «Visual Studio 2022 WPF C# подключение к БД»

Вот это Модель ADO.NET EDM добавить:

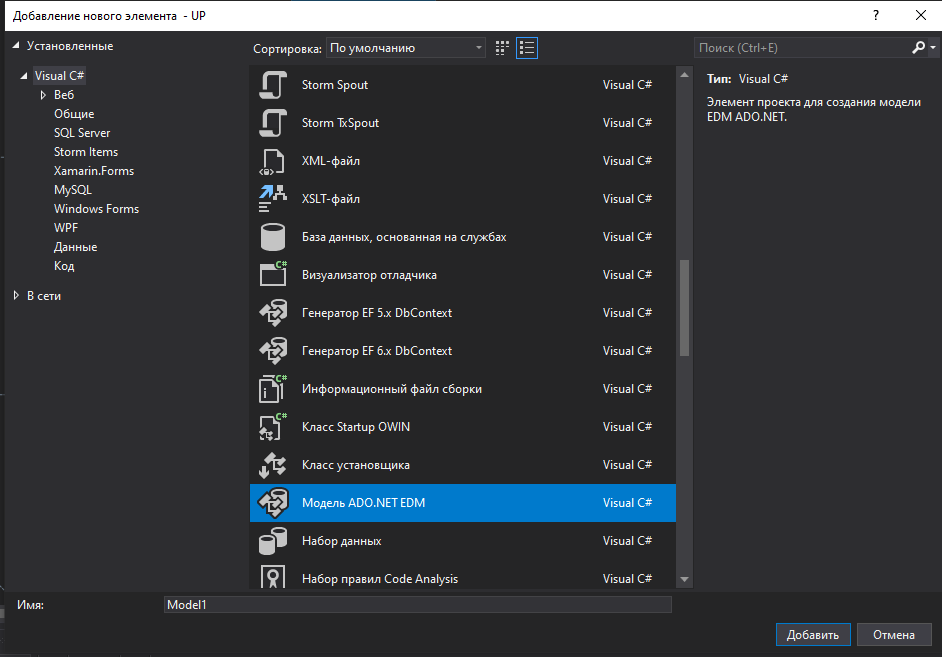


Рисунок 74 – «Visual Studio 2022 WPF C# подключение к БД»

Добавить конструктор EF из базы данных:

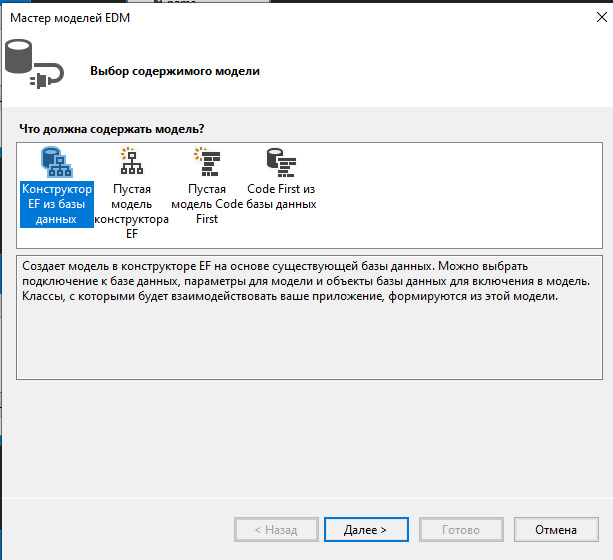


Рисунок 75 – «Visual Studio 2022 WPF C# подключение к БД»

Введите имя сервера своего колледжа:

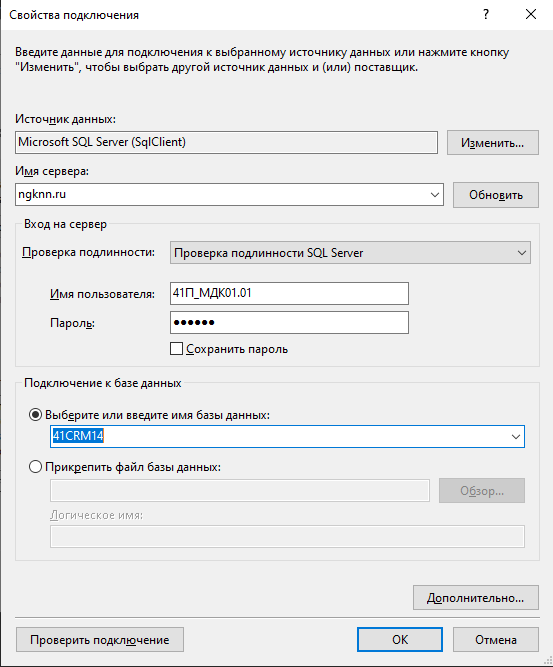


Рисунок 76 – «Visual Studio 2022 WPF C# подключение к БД»

После нажмите ОК.

У нас Visual Studio 2022 появились в БД:

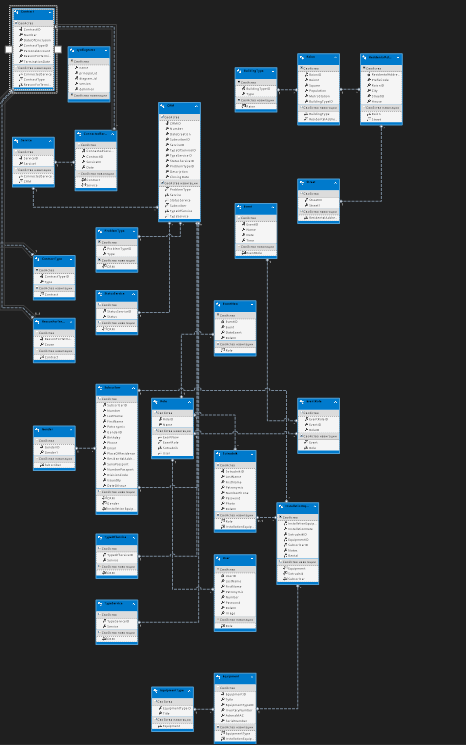


Рисунок 77 – «Visual Studio 2022 WPF C# база данных»

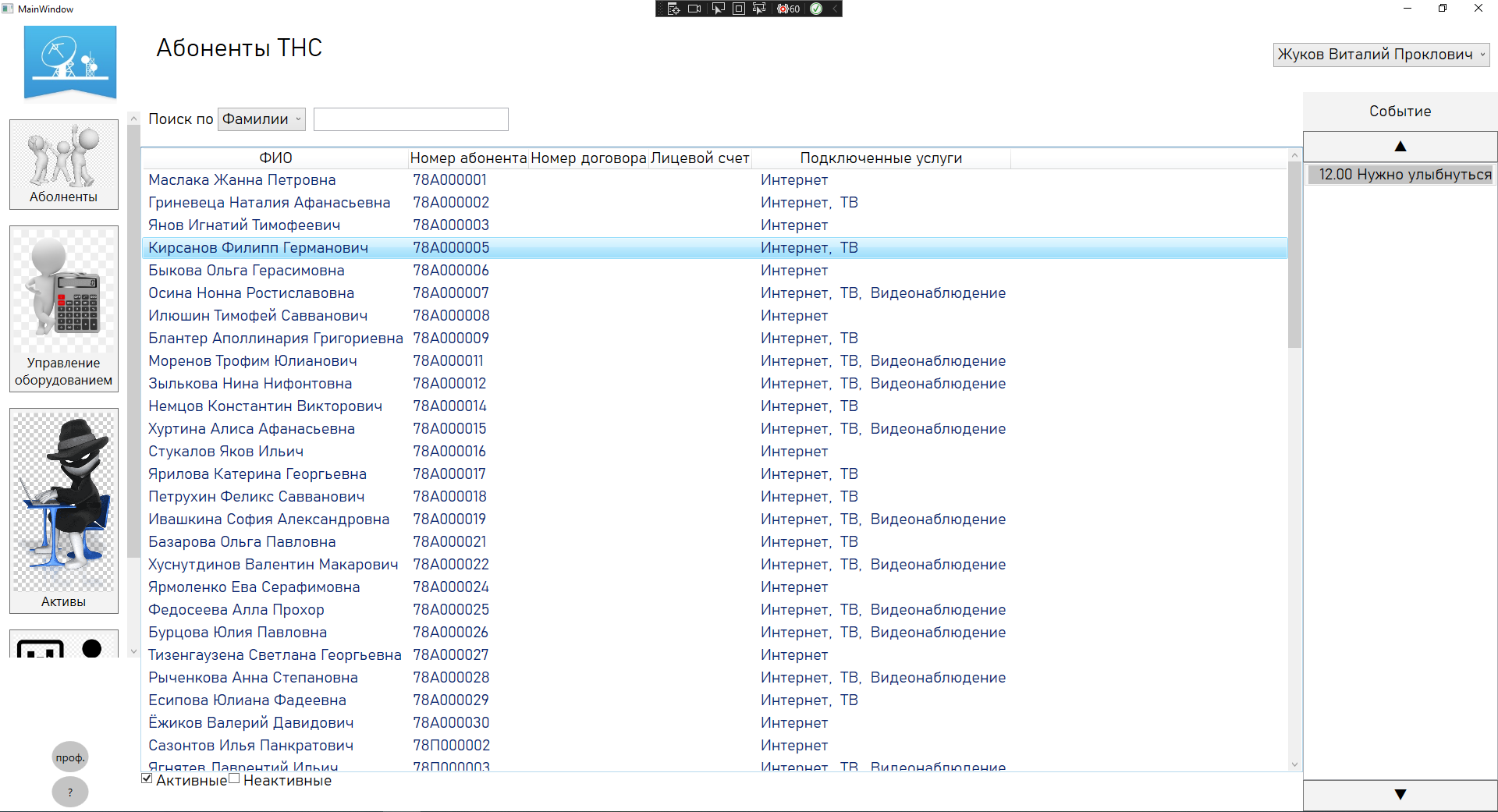


Рисунок 78 – «Форма проекта»

# МДК 01.03 Разработка мобильных приложений

Нашей задачей является разработать приложение для смартфона. Дизайн

приложения находится в Figma (рис. 79)

Нам необходимо следовать дизайну, а также логике переходов, которая

реализована в прототипе (её можно посмотреть, нажав в левом side-блоке на

“prototype”). То есть все экраны должны соответствовать макету (за исключением экранов-заглушек). Текст должен быть вставлен в виде текстовых блоков, поля для ввода - полями для ввода. В прототипе присутствуют экраны-заглушки. Переход на них тоже нужно реализовать. Добавим иконку к приложению и используйте предоставленный шрифт.



Рисунок 79 – Дизайн приложения

**Onboarding**

У пользователя должна быть возможность перейти на окно входа и экранзаглушку - регистрацию.

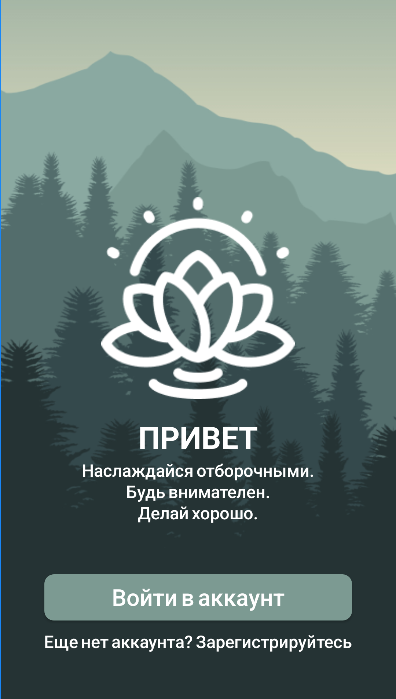


Рисунок 80 – Onboarding

**Login**

На данном экране пользователь должен иметь возможность ввести email и

пароль к своей учетной записи и получить токен авторизации. Перед тем, как отправить данные на сервер для получения токена, необходимо проверить заполненность полей, а также наличие символа @ в поле для Email.

Данные авторизации должны сохраняться на устройстве и после

перезагрузки приложения, то есть пользователь при повторном запуске будет

пропускать экраны onboarding и login.

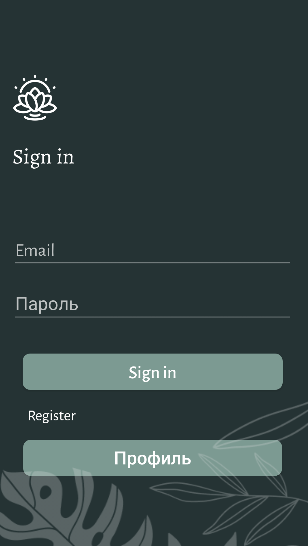


Рисунок 81 - Login

**Main**

На главном экране необходимо выводить полученную информацию о текущем пользователе (изображение профиля и имя). На экране должен выводиться список ощущений, получаемый с севера. Список должен листаться по горизонтали. Список ощущений должен быть отсортирован по параметру “position” по возрастанию. На экране должны выводиться блоки с короткими цитатами, получаемые с сервера. В случае, если список цитат не помещается на экран, то необходимо позволить пользователю листать экран полностью (за исключением нижнего таббара).

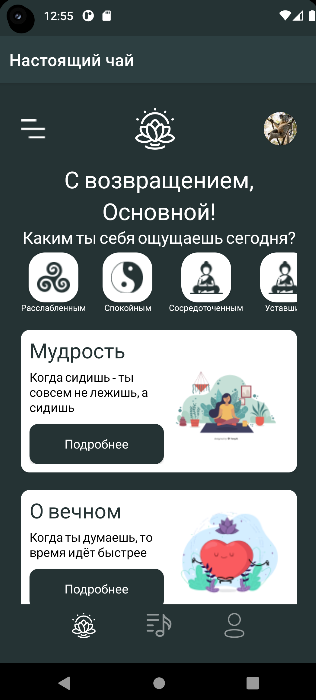


Рисунок 82 – Main

**Profile**

На экране профиля добавьте кнопку “выход”, после которой приложение открывает окно авторизации. В случае, если пользователь вышел из профиля, то в окне Login в поле Email должен отображаться Email пользователя, который был последний раз авторизован в системе. Данный Email должен сохраняться и после перезагрузки приложения. На экране должны быть отображены: имя и фотография пользователя. В нижней части экрана должна быть показана галерея фотографий, которые были загружены из стандартной галереи приложения. На каждой фотографии должно отображаться время её загрузки. Самым последним элементом всегда должна быть кнопка добавления фотографии, при нажатии на которую открывается выбор фотографий из стандартной галереи. Фотографии должны сохраняться в памяти приложения, чтобы они остались в приложении в случае удалениях их из стандартной галереи. Фотографии должны сохранятся и после перезагрузки приложения. После нажатия на фотографию, должен открываться экран Photo.

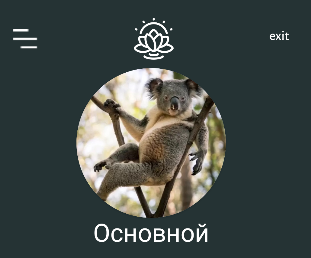


Рисунок 83 – Profile

# Заключение

Таким образом, были достигнуты цели и задачи учебной практики.

Цель учебной заключалась в практическом применении теоретических знаний, полученных студентами. Учебная практика предназначалась для ознакомления в производственных условиях с технологическими процессами на предприятии, а также для подготовки студентов к дальнейшему изучению дисциплин учебного плана специальности.

Задачи учебной практики:

1. Проанализированы предметная область разрабатываемой программы;
2. Разработка программного модуля Visual Studio 2022 WPF на языке C#;
3. Поддержка и тестирование программных модулей
4. Разработка мобильных приложений

# Список литератур

1. <https://metanit.com/java/android/12.3.php>
2. <https://developer.alexanderklimov.ru/android/views/imageview.php>
3. <https://coderlessons.com/articles/mobilnaia-razrabotka-articles/dobavlenie-zhestov-smakhivaniia-v-recyclerviews>
4. <https://metanit.com/sharp/wpf/>

# Приложения

Приложение А

|  |  |
| --- | --- |
|  | **Тестовый документ** |

## **Аннотация теста**

|  |  |
| --- | --- |
| **Название проекта** | MalevPracticeMDK |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Антон |
| **Дата(ы) теста** | **21.03.2023** |

## **Расшифровка тестовых информационных полей:**

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | Название тестируемого проекта |
| **Рабочая версия** | Версия проекта/программного обеспечения (первый тест считается 1.0). |
| **Имя тестирующего** | Имя того, кто проводил тесты |
| **Дата(ы) теста** | Дата(ы) проведения тестов – это один или несколько дней. Если тесты проводились в более протяженный период времени, нужно отметить отдельную дату для каждого теста. |
| **Тестовый пример #** | Уникальный ID для каждого тестового примера. Следуйте некоторым конвенциям, чтобы указать типы тестов. Например,‘TC\_UI\_1′ означает‘user interface test case #1′ ( ТС\_ПИ\_1: тестовый случай пользовательского интерфейса#1) |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Насколько важен каждый тест. Приоритет тестирования для бизнес-правил и функциональных тестовых случаев может быть средним или высоким, в то время как незначительные случаи пользовательского интерфейса могут иметь низкий приоритет. |
| **Заголовок/название теста** | Название тестового случая. Например, Подтвердите страницу авторизации с действительным именем пользователя и паролем. |
| **Краткое изложение теста** | Описание того, что должен достичь тест. |
| **Этапы теста** | Перечислите все этапы теста подробно. Запишите этапы теста в том порядке, в котором они должны быть реализованы. Предоставьте как можно больше подробностей и разъяснений. Пронумерованный список – хорошая идея. |
| **Тестовые данные** | Перечислите/опишите все тестовые данные, используемые для данного тестового случая. Так, фактические используемые входные данные можно отслеживать по результатам тестирования. Например, Имя пользователя и пароль для подтверждения входа. |
| **Ожидаемый результат** | Каким должен быть вывод системы после выполнения теста? Подробно опишите ожидаемый результат, включая все сообщения/ошибки, которые должны отображаться на экране. |
| **Фактический результат** | Каким должен быть фактический результат после выполнения теста? Опишите любое релевантное поведение системы после выполнения теста. |
| **Предварительное условие** | Любые предварительные условия, которые должны быть выполнены до выполнения теста. Перечислите все предварительные условия для выполнения этого тестового случая. |
| **Постусловие** | Каким должно быть состояние системы после выполнения теста? |
| **Статус**  *(Зачет/Незачет)* | Если фактический результат не соответствует ожидаемому результату, отметьте тест как неудачный. В ином случае обновление пройдено. |
| **Примечания/комментарии** | Используйте эту область для любых дополнительных заметок/комментариев/вопросов. Эта область предназначена для поддержки вышеуказанных полей (например, если есть некоторые особые условия, которые не могут быть описаны в любом из вышеуказанных полей, или если есть вопросы, связанные с ожидаемыми или фактическими результатами). |

## Тестовый пример #1:

|  |  |
| --- | --- |
| **Тестовый пример #** | ТС\_ФД\_1 (тестовый случай функционирования добавления) |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Добавление товара в корзину |
| **Краткое изложение теста** | Данный тестовый пример отражает основную суть добавления товара в корзину в режиме администратора |
| **Этапы теста** | 1. Необходимо запустить программу 2. Авторизоваться под учетной записью администратора 3. Нажать на ПКМ и выбрать «Добавить товар» |
| **Тестовые данные** |  |
| **Ожидаемый результат** | После нажатия на ПКМ «Добавить в корзину», товар добавится в корзину |
| **Фактический результат** | Товар добавлен в корзину |
| **Статус** | Зачет |
| **Предварительное условие** | 1. Компьютер с ОС 2. Установленная VS 3. Периферийные устройства |
| **Постусловие** | Товар добавлен в корзину. Исключения не найдены. Программа функционирует |
| **Примечания/комментарии** |  |

## Тестовый пример #2:

|  |  |
| --- | --- |
| **Тестовый пример #** | ТС\_ФУ\_2 (тестовый случай функционирования удаления) |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Удаление товара из списка товаров |
| **Краткое изложение теста** | Данный тестовый пример должен отразить функциональную способность системы удаления товара в режиме администратора |
| **Этапы теста** | 1. Необходимо запустить программу 2. Авторизоваться под учетной записью администратора 3. Нажать на кнопку «Удалить» 4. Нажать на кнопку «Да» в сообщении «Вы хотите удалить элемент?» |
| **Тестовые данные** |  |
| **Ожидаемый результат** | Товар удален |
| **Фактический результат** | Товар не добавлися |
| **Статус** | Зачет |
| **Предварительное условие** | 1. Компьютер с ОС 2. Установленная VS 3. Периферийные устройства |
| **Постусловие** | Товар удалился. Вывелось сообщение «Удалено» |
| **Примечания/комментарии** |  |

## Тестовый пример #3:

|  |  |
| --- | --- |
| Тестовый пример **#** | ТС\_ФУ\_3 (тестовый случай функционирования удаления) |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Удаление товара, при этом добавлен товар в корзину |
| **Краткое изложение теста** | Данный тестовый пример должен отразить функциональную способность системы удаления товара при этом добавлен товар в корзину в режиме администратора |
| **Этапы теста** | 1. Необходимо запустить программу 2. Авторизоваться под учетной записью администратора 3. Нажать на кнопку «Удалить» 4. Нажать на кнопку «Да» в сообщении «Вы хотите удалить элемент?» |
| **Тестовые данные** |  |
| **Ожидаемый результат** | Товар не удален, так как товар находится в корзине |
| **Фактический результат** | Товар не удален |
| **Статус** | Зачет |
| **Предварительное условие** | 1. Компьютер с ОС  2. Установленная VS  3. Периферийные устройства |
| **Постусловие** | Товар не удалился. Вывелось сообщение об ошибках удаления |
| **Примечания/комментарии** |  |

## Тестовый пример #4:

|  |  |
| --- | --- |
| Тестовый пример **#** | ТС\_ФВ\_4 (тестовый случай функционирования входа) |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Вход с неправильными данными |
| **Краткое изложение теста** | Тестирование отражает функциональную способность системы заблокироания входа |
| **Этапы теста** | 1. Необходимо запустить программу 2. Вводить поля неправильными данными 3. Нажать на кнопку «Вход» |
| **Тестовые данные** |  |
| **Ожидаемый результат** | После нажатия на кнопку «Входа», вывелось сообщение о неправильном вводе данных. После этого появится ввод поля из CAPTCHA |
| **Фактический результат** | Вход не удался. Появится ввод поля из CAPTCHA |
| **Статус** | Зачет |
| **Предварительное условие** | 1. Компьютер с ОС  2. Установленная VS  3. Периферийные устройства |
| **Постусловие** | После нажатия на кнопку «Вход». Вывелось сообщение о неправильном вводе данных. Появится ввод поля из CAPTCHA |
| **Примечания/комментарии** |  |

## Test case #5:

|  |  |
| --- | --- |
| **Test Case #** | ТС\_ФВ\_5 (тестовый случай функционирования ввода) |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Вход с правильными данными, но с неправильного кода из CAPTCHA |
| **Краткое изложение теста** | Тестирование отражает функциональную способность системы заблокироания входа на 10 секунд после неправильного ввода кода из CAPTCHA |
| **Этапы теста** | 1. Необходимо запустить программу 2. Вводить поля неправильным кодом 3. Нажать на кнопку «Вход» |
| **Тестовые данные** | После нажатия на кнопку «Входа», вывелось сообщение о неправильном вводе данных и о заблокировании на 10 сек. Система заблокирована на 10 сек. |
| **Ожидаемый результат** | Откроется одно модальное окно с добавлением товара. Открытие второго окна будет невозможным |
| **Фактический результат** | Открылось одно модальное окно с добавлением товара. Второе окно не открылось |
| **Статус** | Зачет |
| **Предварительное условие** | 1. Компьютер с ОС  2. Установленная VS  3. Периферийные устройства |
| **Постусловие** | После нажатия на кнопку «Входа», вывелось сообщение о неправильном вводе данных и о заблокировании на 10 сек. Система заблокирована на 10 сек. |
| **Примечания/комментарии** |  |