**Частное образовательное учреждение высшего образования**

**«Международный Институт Дизайна и Сервиса»**

**(ЧОУВО МИДиС)**

Кафедра математики и информатики

**КУРСОВАЯ РАБОТА**

По дисциплине

«Объектно-ориентированный анализ и программирование на языке С#»

На тему: «Информационная система автосалона»

Направление: 09.03.03 Прикладная Информатика

Выполнил работу:

A.Я. Гарток

Группа ПИ-226

Проверил работу:

С.С. Чеботарев

Челябинск 2023

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc166559291)

[ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc166559292)

[1.1 Описание предметной области 4](#_Toc166559293)

[1.2 Анализ существующих решений 4](#_Toc166559294)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ 9](#_Toc166559295)

[2.1 Обоснование выбора СУБД 9](#_Toc166559296)

[СУБД - Microsoft SQL Server + MS SSMS 10](#_Toc166559297)

[2.2 Разработка инфологической модели 10](#_Toc166559298)

[2.3 Нормализация базы данных 11](#_Toc166559299)

[2.4 Разработка даталогической модели базы данных 12](#_Toc166559300)

[2.5 Инициализация базы данных 13](#_Toc166559301)

[2.6 Заполнение таблиц данными 15](#_Toc166559302)

[ГЛАВА 3. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ 17](#_Toc166559303)

[3.1 Обоснование выбора среды разработки 17](#_Toc166559304)

[3.2 Проектирование приложения 19](#_Toc166559305)

[3.3 Разработка приложения 24](#_Toc166559306)

[ЗАКЛЮЧЕНИЕ 34](#_Toc166559307)

[СПИСОК ИСПОЛЬЗУЕМЫХ РЕСУРСОВ 35](#_Toc166559308)

# **ВВЕДЕНИЕ**

Информационная система автосалона — это специализированная система, предназначенная для автоматизации и управления различными аспектами деятельности автосалона. Она объединяет в себе программное обеспечение, данные и процессы, необходимые для эффективного функционирования бизнеса. Цель курсовой работы заключается в приобретении навыков проектирования и разработки базы данных, а также управления данными в базе с использованием программного обеспечения.

Объектом исследования курсовой работы является база данных.

Предметом исследования является система управления автосалоном

Основные задачи работы:

1. Исследование предметной области.
2. Разработка модели системы управления автосалоном
3. Выбор СУБД и среды программирования.
4. Проектирование и создание базы данных.
5. Проектирование и создание приложения для управления базой

данных.

# **ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

## **1.1 Описание предметной области**

Предметная область автосалона включает в себя информацию об автомобилях (уникальный идентификационный номер, марка авто, модель авто, фото, идентификационный номер конфигурации авто)

У каждого автомобиля есть конфигурация его составных частей (уникальный идентификационный номер, тип двигателя, тип колес, цвет авто)

Важной частью предметной области являются клиенты автосалона, автосалон ведет учет пользователей. У каждого клиента существует личный кабинет с информацией о своем профиле, а также с информацией о своем автомобиле.

Клиент автосалона имеет возможность регистрации и авторизации в приложении, а также оформить заказ на понравившийся автомобиль, используя приложение.

Пользователи приложения, имеющие статус «admin», имеют возможность редактировать список пользователей, обрабатывать заказы пользователей, а также редактировать справочные таблицы.

## **1.2 Анализ существующих решений**

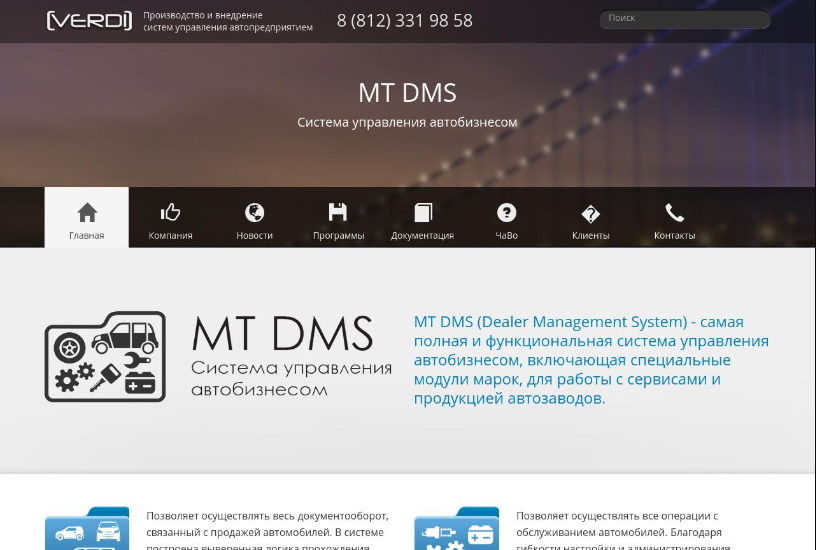
Анализ существующих программных решений, которые выполняют функции схожие с системой управления автосалоном.

Перечень схожих решений:

**«MT DMS»** -это комплексная система управления автодилером, формирует единое информационное пространство на предприятии. MT DMS спроектирована, как универсальное решение, «закрывающее» большинство актуальных потребностей автодилеров, как отдельных, так и в составе дилерской группы.

Основные возможности MT DMS включают:

* MT DMS позволяет руководителю предприятия быть в курсе оперативной деятельности предприятия в любой момент времени, благодаря гибко настраиваемым отчетам. Нет необходимости запрашивать информацию от руководителей направлений и терять время, ожидая ответ.
* Перед обновлением есть возможность предварительно оценить нововведения, не затрагивая рабочую базу данных. Посмотреть на новые функции в тестовом режиме.
* Отчёты в MT DMS интерактивны. Это означает, что сотрудник может не просто просмотреть итоговые данные и распечатать их, но и сделать развёрнутую детализацию по конкретным позициям.
* Программисты создали дружественный пользователю конструктор печатных форм для создания отчётов. С ним возможно автоматическое заполнение практически любых полей из таблиц базы данных.
* Существует возможность отправлять сообщения электронной почты в формате HTML.

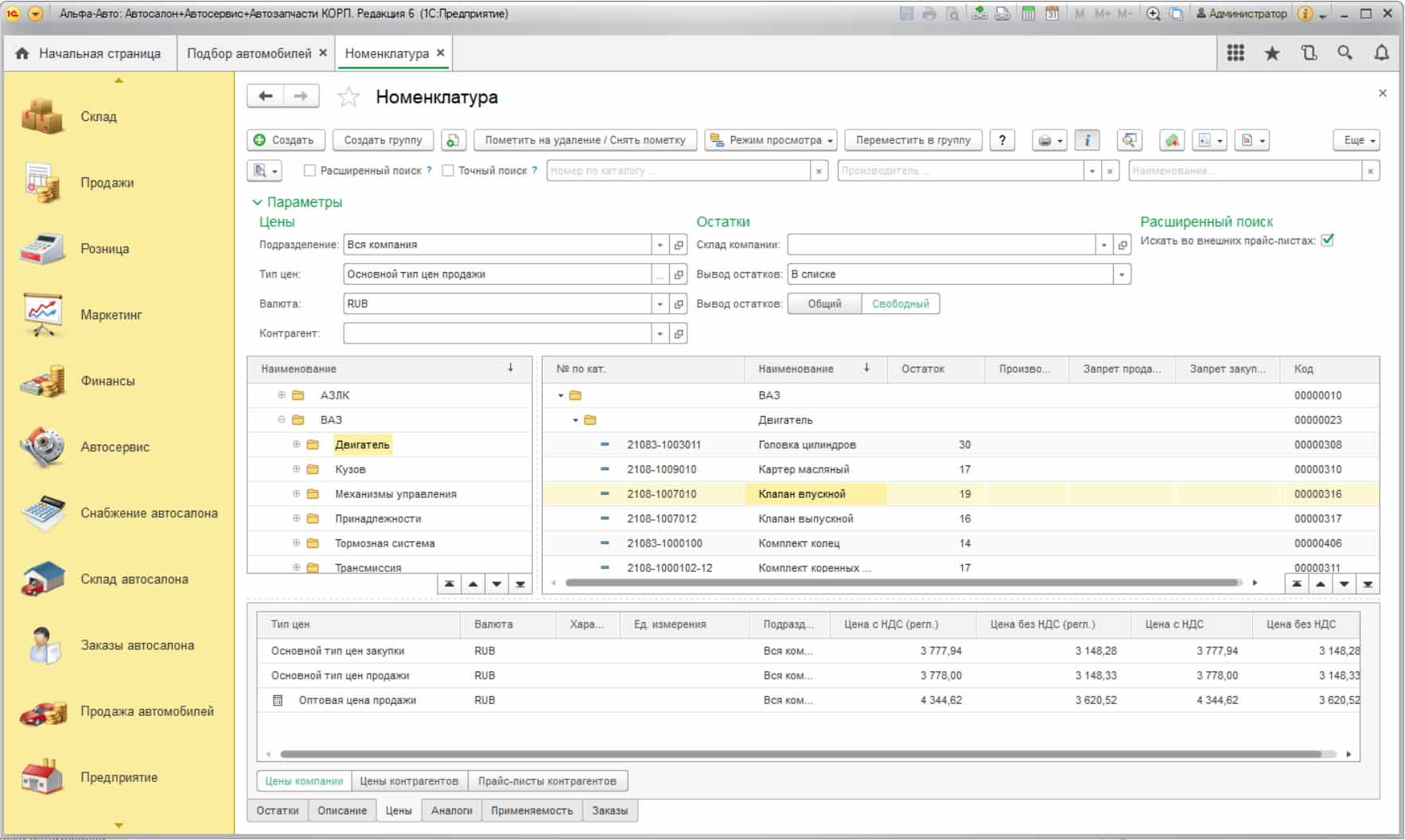


*Рис. 1. MT DMS*

**«1C Альфа-Авто»** - комплексное программное решение на базе 1С, специально разработанное для автосалонов и автосервисов. Программа отличается высоким уровнем функциональности и гибкими настройками под потребности автомобильного бизнеса. Основными функциями «Avaya IP Office» являются: - Управление звонками: IP Office позволяет управлять звонками и маршрутизацией вызовов в соответствии с заданными правилами. Это включает автоматическое распределение вызовов, управление очередями вызовов и перенаправление вызовов.

Основные возможности программы 1C Альфа-Авто:

* Ведение списка аналогов и замен: Альфа-авто позволяет вести список аналогов и замен запасных частей с возможностью быстрого и удобного доступа к данным об их текущих остатках. Это способствует более эффективному управлению запасами и быстрой подборке необходимых деталей для ремонта.
* Заполнение ячеек хранения запчастей: система предоставляет удобный интерфейс для заполнения данных о расположении запчастей на складе. Это позволяет быстрее и проще находить нужные запчасти при выполнении заказов.
* Подключение внешних прайс-листов: система имеет возможность подключения внешних прайс-листов разных форматов (txt, ods, xls, dbf). Это позволяет быстро и удобно обновлять данные о ценах и наличии запчастей у поставщиков без необходимости ручного ввода информации.
* Оформление заказов покупателей: система Альфа-авто позволяет с минимальными усилиями оформлять заказы покупателей с резервированием имеющихся в наличии позиций и заказом отсутствующих.
* Уведомление менеджеров о поступлении товаров: система автоматически уведомляет менеджеров о поступлении товаров по заказам покупателей, что существенно ускоряет процесс выполнения заказов и минимизирует недостачу запчастей.



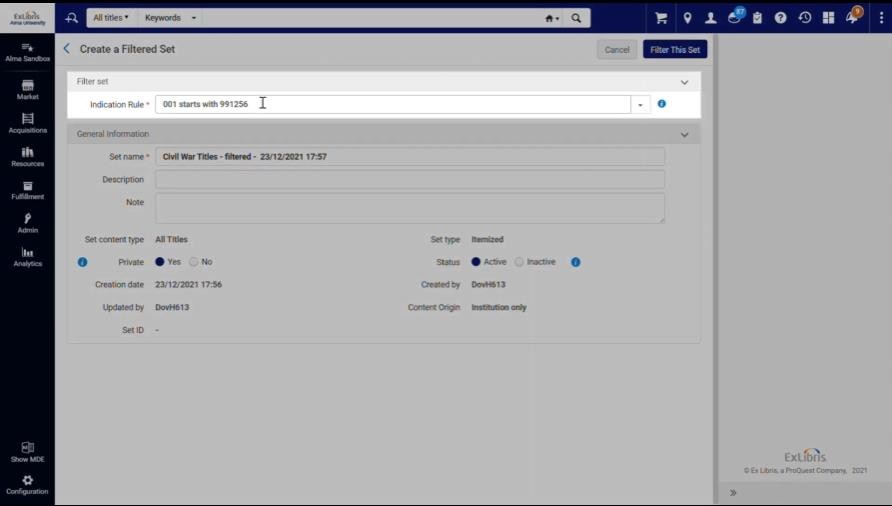
*Рис. 2. 1C Альфа-Авто*

Несмотря на то, что в России на сегодняшний день существует множество программ, ориентированных на автомобильный бизнес, Альфа-Авто продолжает оставаться лидером на рынке.

**«АвтоДилер»** - Онлайн программа для автосервисов. (Склад, проценка, продажа запчастей, CRM, запись на ремонт).

Основные возможности программы АвтоДилер:

* Программа позволяет настраивать распределение товаров по разным складам и стеллажам, оформлять приходные и расходные накладные, размещать заказ клиентов и заказы поставщику.
* Программа автоматически фиксирует обращения клиентов и создает на их основе сделки. Сделки помещаются на канбан, откуда их берут в работу сотрудники.
* Есть история ремонтов и раздел рекомендованных товаров и услуг. Запись клиентов на ремонт возможна в программе и через мобильное приложение.
* Есть интеграция с кассовыми сервисами Эвотор, Штрих-М и Атол, мессенджерами Telegram и WhatsApp, операторами ip-телефонии. Также есть возможность настроить выгрузку данных в 1С:Бухгалтерия.



*Рис. 3. АвтоДилер*

# **ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ**

## **2.1 Обоснование выбора СУБД**

Система управления базой данных (СУБД) является критическим компонентом информационной системы. Она необходима для разработки программы на алгоритмическом языке. Основные функции СУБД включают:

* управление данными во внешней памяти (на дисках);
* управление данными в оперативной памяти;
* журнализация изменений и восстановление базы данных после сбоев;

Критерии выбора СУБД:

* Тип данных: необходимо определить, какие типы данных будут храниться в базе данных и выбрать СУБД, поддерживающую эти типы данных.
* Масштаб проекта: в зависимости от размеров проекта необходимо выбрать СУБД, способную эффективно обрабатывать большие объемы данных.
* Скорость обработки данных: если требуется быстрая обработка и доступ к данным, следует выбрать СУБД с высокой производительностью.
* Гибкость: СУБД должна быть гибкой и легко настраиваемой, чтобы соответствовать требованиям проекта.
* Стоимость: необходимо учитывать бюджет проекта при выборе СУБД, чтобы избежать лишних затрат.

## **СУБД - Microsoft SQL Server + MS SSMS**

Так как приложение будет разработано на платформе WPF (Windows Presentation Foundation ) с целью упрощения процесса разработки и оптимизации работы приложения для его разработки была выбрана СУБД Microsoft SQL Server в связке с MS SSMS.

**Microsoft SQL Server** — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

**Microsoft SQL Server Management Studio** (MS SSMS) — утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Удобный графический интерфейс упрощает взаимодействие с СУБД, что ускоряет процесс разработки приложения.

## **2.2 Разработка инфологической модели**

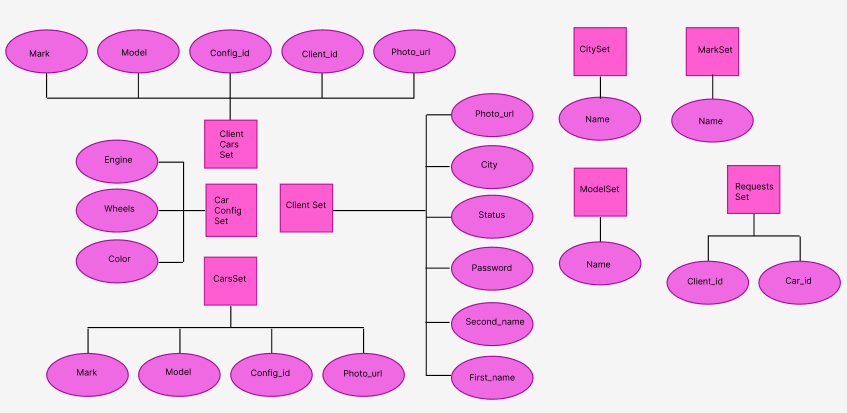
Инфологическая модель – это потоки информации, сущности и связи данной области. В такой модели указываются связи между сущностями данной предметной области.

Сущность – представляет тип объектов, которые должны храниться в базе данных.

Связь – это ассоциирование нескольких сущностей с целью отыскания одних из них по значениям других.

Атрибут – это имя или структура поля записи. Атрибут характеризует размер или тип информации, содержащейся в поле. Его наименование должно быть уникальным для конкретного типа сущностей

Первичный ключ - это специальное поле в таблице, которое однозначно идентифицирует каждую запись или строку в базе данных.



*Рис. 4. Концептуальная модель*

## **2.3 Нормализация базы данных**

Нормализация — это метод проектирования реляционных БД, который помогает правильно структурировать таблицы данных. Каждый последующий шаг разбивает таблицу на более лёгкую в управлении информацию. Этот процесс направлен на создание системы с чётким представлением информации и взаимосвязей, без избыточности и потери данных.

Избыточность данных в БД относится к нежелательным явлениям, поскольку ведет к увеличению объема памяти, необходимого для физического хранения отношений, например, если потребуется изменить, удалить или добавить определенные данные то взаимодействовать придется с более чем одной таблицей.

**Первая нормальная форма**

* Отсутствие упорядочивания строк сверху вниз.
* Отсутствие упорядочивания столбцов слева направо.
* Отсутствие повторяющихся строк.

**Вторая нормальная форма**

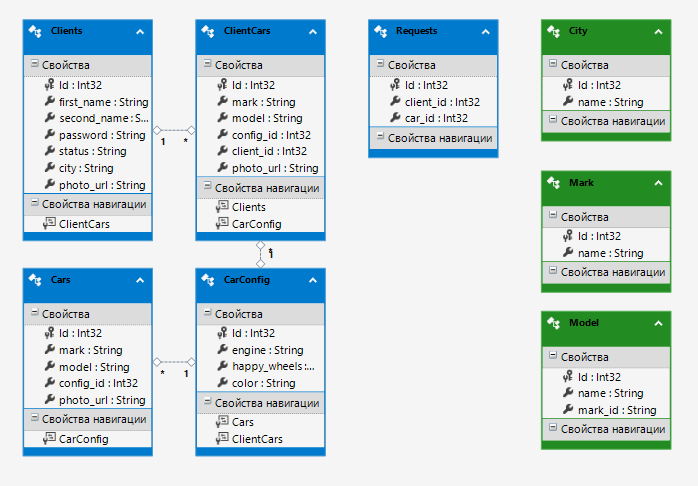
* Таблица находится в первой нормальный форм.
* Все неключевые атрибуты зависят от полного первичного ключа таблицы.

**Третья нормальная форма**

* Таблица находится во второй нормальной форме.
* Неключевые атрибуты не зависят друг от друга.
* Неключевые атрибуты не зависят от части первичного ключа.

## **2.4 Разработка даталогической модели базы данных**

Даталогическая модель – это модель логического уровня системы, представляющая собой отображение логических связей между элементами базы данных в среде конкретной СУБД. Конечным результатом даталогического проектирования является описание логической структуры базы данных на языке описания данных, поддерживаемом конкретной СУБД.



*Рис. 5. Даталогическая модель*

## **2.5 Инициализация базы данных**

Таблицы:

Создание таблицы «ClientsSet»

CREATE TABLE [dbo].[ClientsSet] (

[Id] int IDENTITY(1,1) NOT NULL,

[first\_name] nvarchar(max) NOT NULL,

[second\_name] nvarchar(max) NOT NULL,

[password] nvarchar(max) NOT NULL,

[status] nvarchar(max) NOT NULL,

[city] nvarchar(max) NOT NULL,

[photo\_url] nvarchar(max) NOT NULL

);

GO

Создание таблицы «CarsSet»

CREATE TABLE [dbo].[CarsSet] (

[Id] int IDENTITY(1,1) NOT NULL,

[mark] nvarchar(max) NOT NULL,

[model] nvarchar(max) NOT NULL,

[config\_id] int NOT NULL,

[photo\_url] nvarchar(max) NOT NULL,

[CarConfig\_Id] int NOT NULL

);

GO

Создание таблицы «CarConfigSet»

CREATE TABLE [dbo].[CarConfigSet] (

[Id] int IDENTITY(1,1) NOT NULL,

[engine] nvarchar(max) NOT NULL,

[happy\_wheels] nvarchar(max) NOT NULL,

[color] nvarchar(max) NOT NULL

);

GO

Создание таблицы «RequestsSet»

CREATE TABLE [dbo].[RequestsSet] (

[Id] int IDENTITY(1,1) NOT NULL,

[client\_id] int NOT NULL,

[car\_id] int NOT NULL

);

GO

Создание таблицы «ClientCarsSet»

CREATE TABLE [dbo].[ClientCarsSet] (

[Id] int IDENTITY(1,1) NOT NULL,

[mark] nvarchar(max) NOT NULL,

[model] nvarchar(max) NOT NULL,

[config\_id] int NOT NULL,

[client\_id] int NOT NULL,

[photo\_url] nvarchar(max) NOT NULL,

[Clients\_Id] int NOT NULL,

[CarConfig\_Id] int NOT NULL

);

GO

Создание таблицы «CitySet»

CREATE TABLE [dbo].[CitySet] (

[Id] int IDENTITY(1,1) NOT NULL,

[name] nvarchar(max) NOT NULL

);

GO

Создание таблицы «MarkSet»

CREATE TABLE [dbo].[MarkSet] (

[Id] int IDENTITY(1,1) NOT NULL,

[name] nvarchar(max) NOT NULL

);

GO

Создание таблицы «ModelSet»

CREATE TABLE [dbo].[ModelSet] (

[Id] int IDENTITY(1,1) NOT NULL,

[name] nvarchar(max) NOT NULL,

[mark\_id] nvarchar(max) NOT NULL

);

GO

## **2.6 Заполнение таблиц данными**

Таблицы:

Запалнение таблицы «MarkSet»

INSERT INTO [MarkSet] ([name]);

VALUES(N' ASA');

INSERT INTO [MarkSet] ([name]);

VALUES(N' Jacer');

INSERT INTO [MarkSet] ([name]);

VALUES(N' Bisu');

INSERT INTO [MarkSet] ([name]);

VALUES(N' BMW');

Запалнение таблицы «ModelSet»

INSERT INTO [ModelSet] ([name], [mark\_id]);

VALUES(N'A2', 1);

INSERT INTO [ModelSet] ([name], [mark\_id]);

VALUES(N'A5', 1);

INSERT INTO [ModelSet] ([name], [mark\_id]);

VALUES(N' Concur V2', 2);

INSERT INTO [ModelSet] ([name], [mark\_id]);

VALUES(N' Bouble Bb', 2);

INSERT INTO [ModelSet] ([name], [mark\_id]);

VALUES(N' Virtchezz', 3);

INSERT INTO [ModelSet] ([name], [mark\_id]);

VALUES(N' Povorocz', 3);

INSERT INTO [ModelSet] ([name], [mark\_id]);

VALUES(N' Rashah', 4);

INSERT INTO [ModelSet] ([name], [mark\_id]);

VALUES(N' Shrimp V8', 4);

Запалнение таблицы «CitySet»

INSERT INTO [CitySet] ([name])

VALUES(N' Chelyabinck ');

INSERT INTO [CitySet] ([name]);

VALUES(N' Moscow ');

INSERT INTO [CitySet] ([name]);

VALUES(N' Omsk ');

**2.7 Дополнительные инструменты**

**Figma** — онлайн-редактор, в котором удобно проектировать интерфейсы, создавать макеты сайтов, мобильных приложений, презентации, иллюстрации, логотипы и анимацию.

Сервис доступен по подписке, предусмотрен бесплатный тарифный план для одного пользователя. Имеются офлайн-версии для Windows, macOS. Реализована интеграция с корпоративным мессенджером Slack и инструментом прототипирования Framer.

Используется как для создания упрощённых прототипов интерфейсов, так и для детальной проработки дизайна интерфейсов мобильных приложений, веб-сайтов, корпоративных порталов.

# **ГЛАВА 3. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ**

## **3.1 Обоснование выбора среды разработки**

**Microsoft Visual Studio 2022** — это мощное средство разработки, которое позволяет выполнять весь цикл разработки в одном месте. Данная среда разработки предназначена для записи, редактирования, отладки и сборки кода, а затем и для развёртывания приложения.

Данная версия более быстрая, более производительная и упрощенная, предназначенная для учащихся, а также пользователей, которые создают решения промышленного масштаба.

При помощи данной среды разработки можно создавать игры, консольные приложения и приложения с графическим интерфейсом. Примером приложения с графическим интерфейсом могут послужить WPF, UWP, Windows Forms. Также Microsoft Visual Studio подходит для разработки веб-приложений и веб-служб для платформ: Windows Mobile, Windows CE, Windows, NET Core, .NET, MAUI, Xbox.

Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

Обширное сообщество и поддержка: Visual Studio имеет огромное сообщество разработчиков, которые готовы поделиться своим опытом, решить проблемы и предложить лучшие практики, что делает работу в Visual Studio 2022 еще более продуктивной.

Связанные сервисы и инструменты: Visual Studio 2022 интегрируется с другими сервисами и инструментами Microsoft, такими как Azure DevOps и GitHub, что упрощает процесс разработки и развертывания приложений.

**Windows Presentation Foundation (WPF)** — это фреймворк для создания графических пользовательских интерфейсов в приложениях, работающих под управлением операционной системы Windows. WPF обеспечивает разработчикам широкие возможности для создания современных и стильных интерфейсов, которые могут включать в себя трехмерную графику, анимацию, эффекты и другие элементы. Он основан на языке разметки XAML (Extensible Application Markup Language), который позволяет создавать интерфейсы декларативно.

WPF также предлагает мощные средства для управления данными, взаимодействия с пользователем, мультимедийных возможностей, а также поддерживает различные типы ввода, такие как сенсорные экраны и стилусы. Благодаря своей гибкости и масштабируемости, WPF является популярным выбором для создания различных типов приложений, от настольных приложений до веб-приложений.

Графической технологией, лежащей в основе WPF, является DirectX, в отличие от Windows Forms, где используется GDI/GDI+. Производительность WPF выше, чем у GDI+ за счёт использования аппаратного ускорения графики через DirectX. Также существует урезанная версия CLR, называющаяся WPF/E, она же известна как Silverlight.

**ADO.NET Entity Framework (EF)** — это набор технологий в ADO.NET, которые поддерживают разработку приложений программного обеспечения, ориентированных на данные. Архитекторам и разработчикам приложений, ориентированных на обработку данных, приходится учитывать необходимость достижения двух совершенно различных целей. Они должны моделировать сущности, связи и логику решаемых бизнес-задач, а также работать с ядрами СУБД, используемыми для сохранения и получения данных.

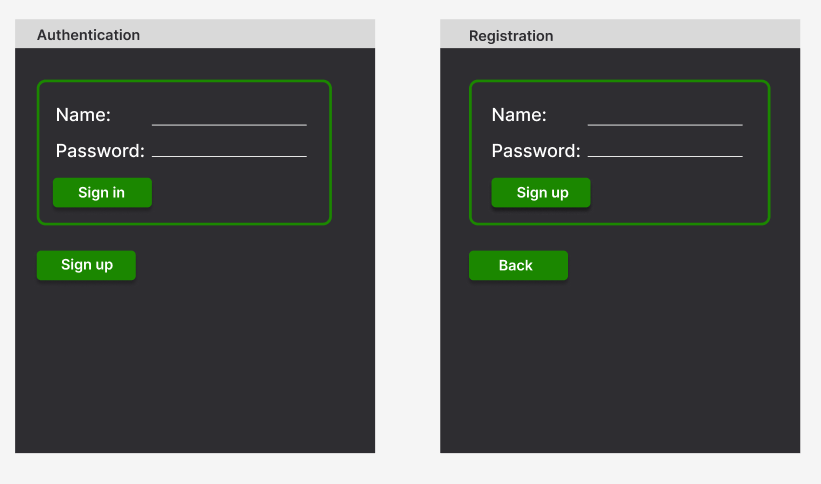
Учитывая все вышесказанное, я выбрал данный инструмент для реализации приложения.

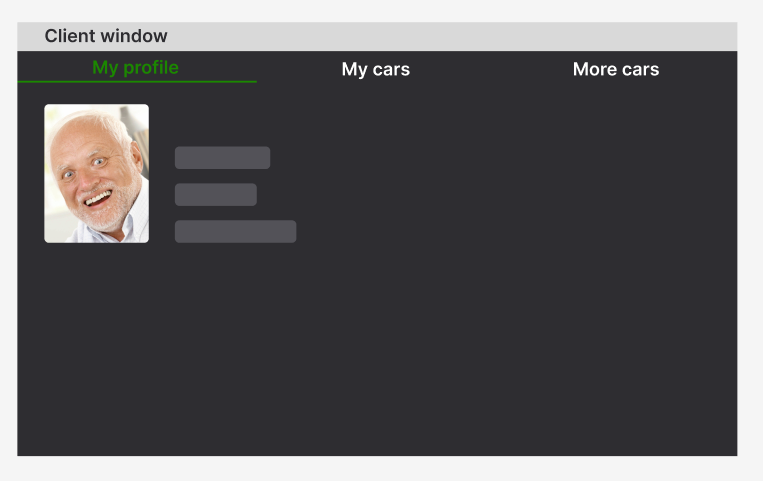
В качестве технологии реализации была выбрана платформа “WPF” в cвязке с пакетом “Entity framework”

## **3.2 Проектирование приложения**

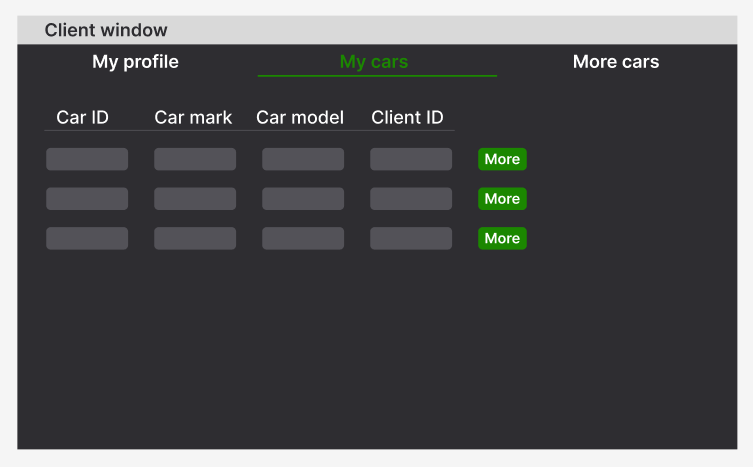
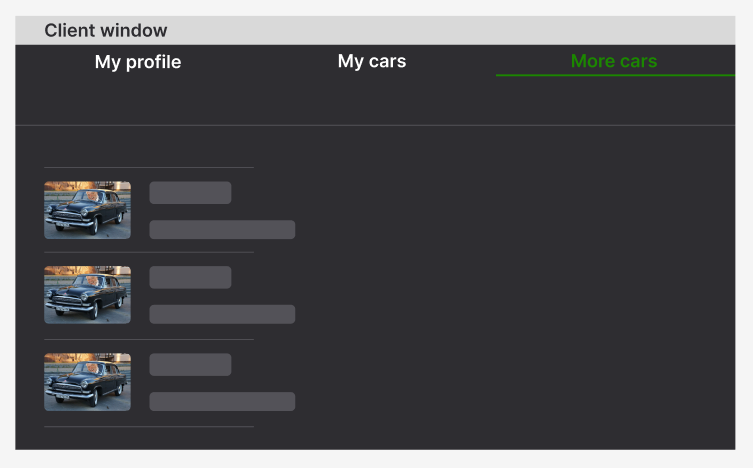
Для разработки макета приложения и проработки его структуры было выбрано приложение «figma».

Окно авторизации – это первое окно доступное пользователю, после авторизации пользователь будет переведен на страницу клиента или на панель администратора в зависимости от статуса пользователя. Если пользователь не зарегистрирован в системе, то он может перейти на форму регистрации для дальнейшей регистрации.

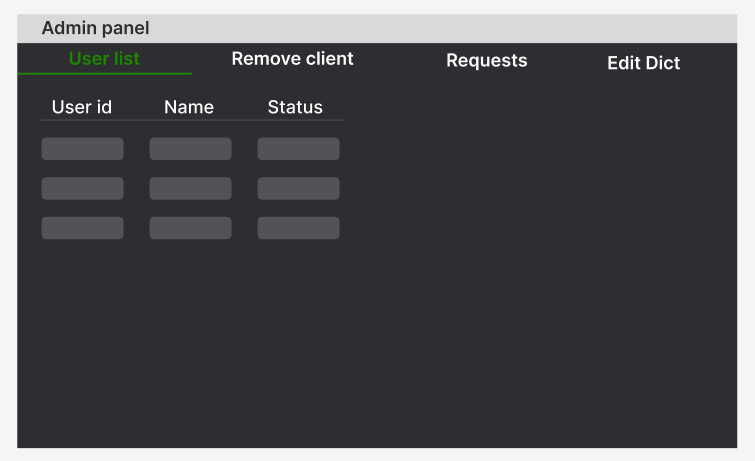
*Рис. 6. Формы авторизации и регистрации *

На странице клиента, клиенту доступна информация о его профиле, информация о его автомобилях, а также возможность заказать новый автомобиль.

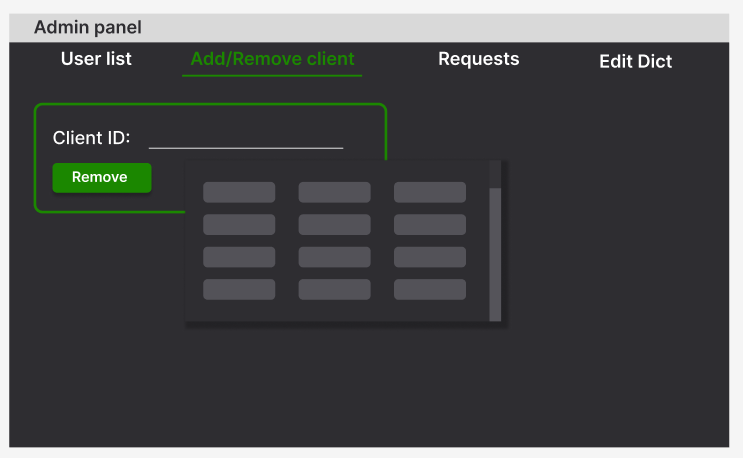
*Рис. 7. Профиль пользователя*

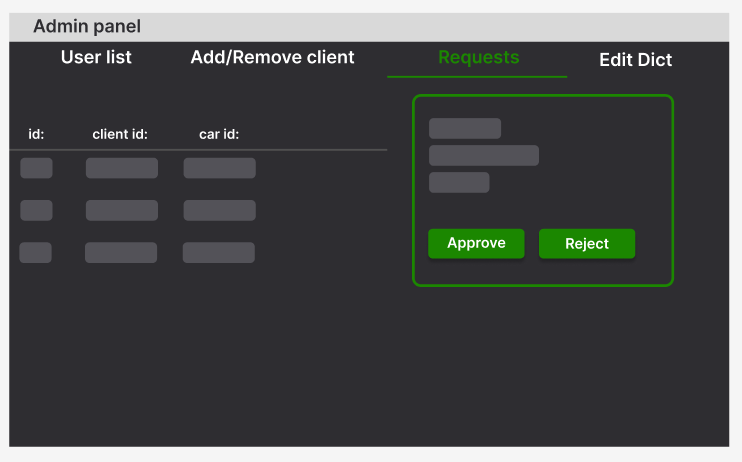
*Рис. 8. Авто пользователя*

*Рис. 9. Доступные авто*

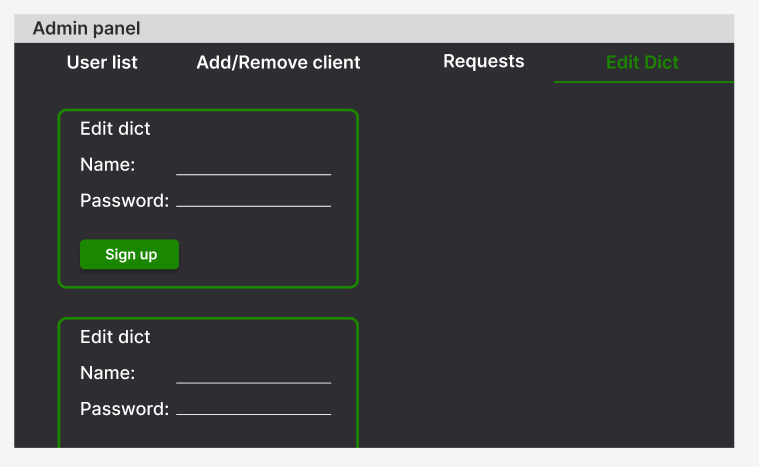
При авторизации в роли администратора пользователю будет доступны четыре вкладки. На первой вкладке расположен список клиентов зарегистрированных в системе, на второй вкладке расположена форма для удаления клиента из системы, на третьей вкладке расположены запросы пользователей на приобретение автомобилей, у администратора есть возможность подтвердить или отклонить запрос, на последней вкладке администратору будут доступны формы редактирования справочных таблиц.

*Рис. 10. Список пользователей*



*Рис. 11. Удаление клиента*

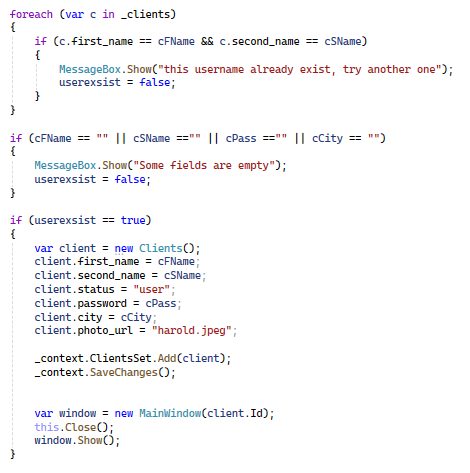
*Рис. 12. Запросы пользователей*



*Рис. 13. Редактирование справочных таблиц*

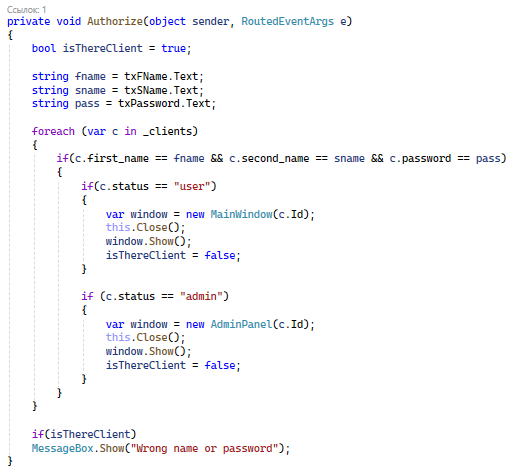
## **3.3 Разработка приложения**

Добавление новых пользователей в базу данных происходит при их регистрации. Данные из формы регистрации записываются локальные переменные метода добавления нового пользователя. Далее производится проверка на наличие данного пользователя в системе, а также проверка на наполненность полей, в случае если пользователя в системе нет и все поля заполнены, новый пользователь будет зарегистрирован. Добавление всех новых записей в базу данных происходит схожим путем.



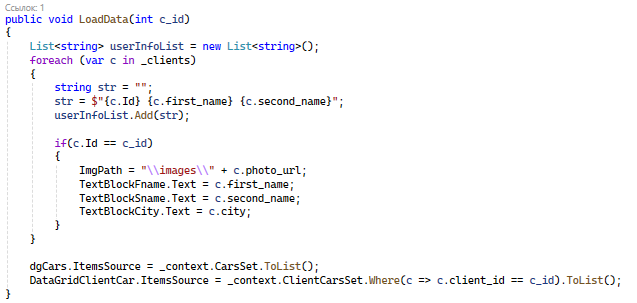
*Рис. 14. Регистрация пользователя*

Авторизация реализована схожим с регистрацией образом. В локальные переменные метода авторизации записываются данные из формы авторизации, далее происходит проверка наличия данного пользователя в системе и сверка его пароля. В зависимости от статуса пользователя он будет переправлен либо на страницу пользователя, либо на панель администратора.



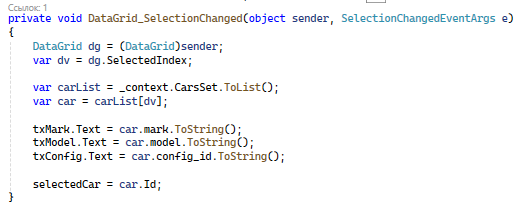
*Рис. 15. Метод авторизации*

При объявлении окна пользователя в методе инициализации вызывается метод подгрузки данных, который обновляет все таблицы и основные текстовые поля для поддержания их в актуальном состоянии.



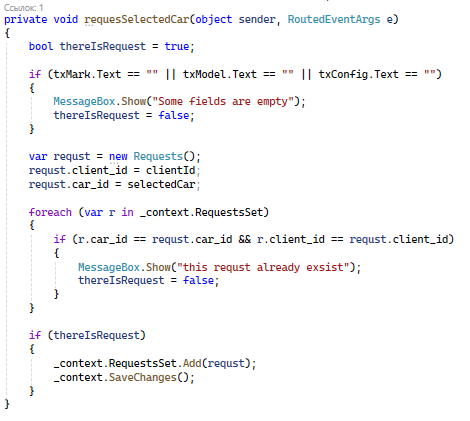
*Рис. 16. Метод подгрузки данных*

При клике на пункты таблицы во вкладке для заказа новых автомобилей вызывается метод обновления полей с информацией об автомобиле



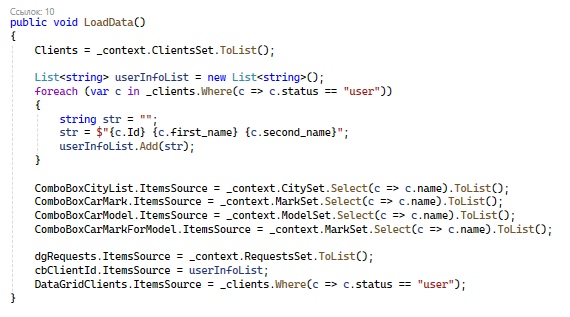
*Рис. 17. Метод обновления полей*

Метод заказа автомобиля реализован следующим образом:



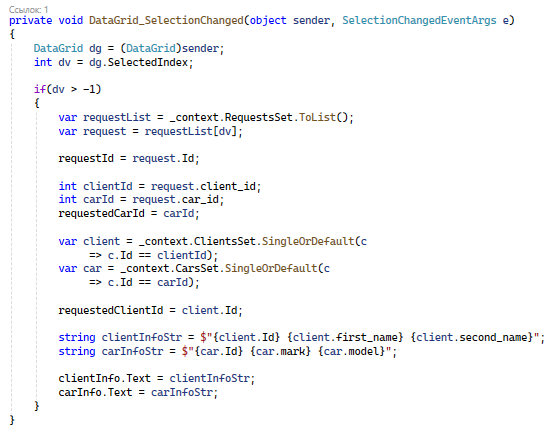
*Рис. 18. Метод заказа автомобиля*

При объявлении панели администратора в методе инициализации вызывается метод подгрузки данных, который обновляет все таблицы и основные текстовые поля для поддержания их в актуальном состоянии.

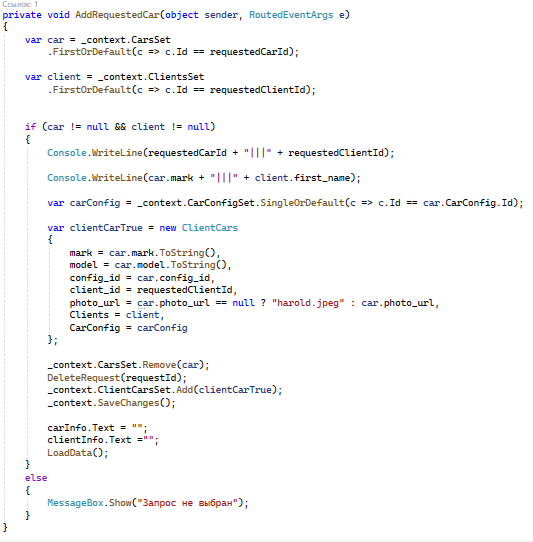


*Рис. 19. Метод подгрузки данных (панель администратора)*

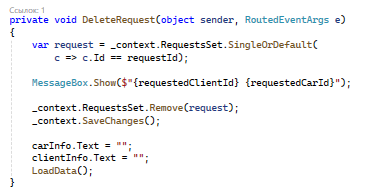
При клике на пункты таблицы во вкладке заказы, вызывается метод обновления полей с информацией о заказе, после чего администратор может подтвердить или отклонить запрос.



*Рис. 20. Метод обновления полей (панель администратора)*

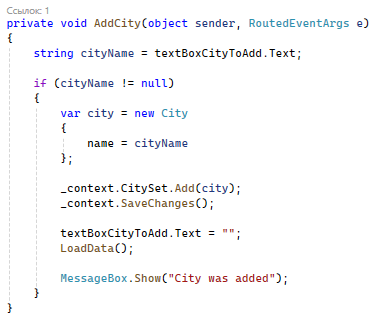


*Рис. 21. Метод подтверждения запроса*



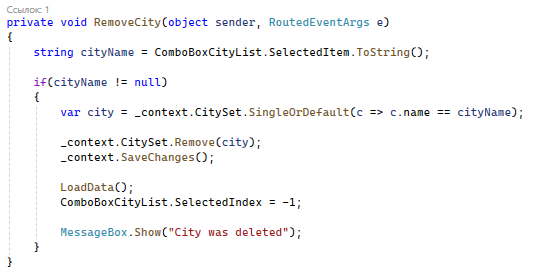
*Рис. 22. Метод отклонения запроса*

Добавление новых городов в таблицу реализовано следующим образом: в локальную переменную метода добавления города записывается значение из текстового поля, после производится проверка на наполненность переменной, в случае если переменная не пуста и не равна null создается экземпляр класса City и добавляется в базу данных. После добавления строки в базу данных текстовое поле обнуляется.



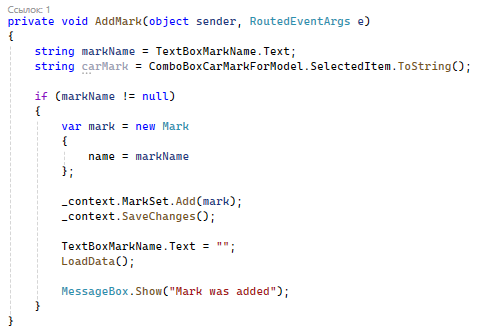
*Рис. 23. Метод добавления города*

Удаление города из таблицы реализовано следующим образом: значение из комбо-бокса записывается в локальную переменную метода удаления города, далее в случае если переменная не пуста и не равна null в таблице находится запись со значением равным локальной переменной, после чего запись удаляется. После удаления записи вызывается метод подгрузки данных, а активное значение комбо-бокса сбрасывается на нулевое.



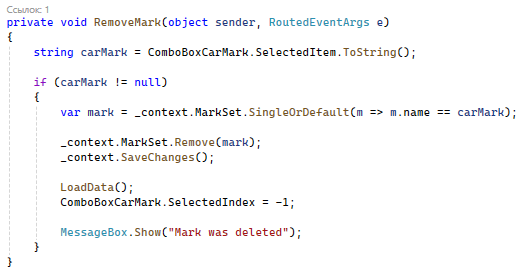
*Рис. 24. Метод удаление города*

Добавление новых марок автомобилей в таблицу реализовано следующим образом: в локальную переменную метода добавления марок авто записывается значение из текстового поля, после производится проверка на наполненность переменной, в случае если переменная не пуста и не равна null создается экземпляр класса Mark и добавляется в базу данных. После добавления строки в базу данных текстовое поле обнуляется.



*Рис. 25. Метод добавления марки авто*

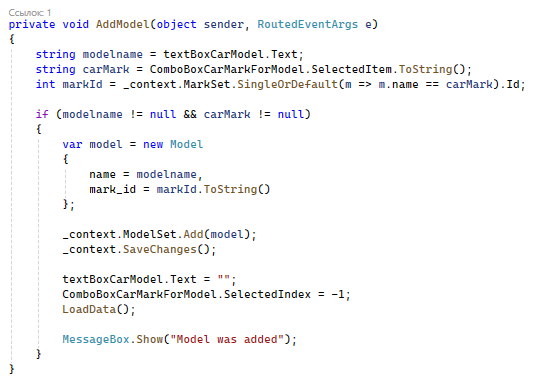
Удаление автомобильной марки из таблицы реализовано следующим образом: значение из комбо-бокса записывается в локальную переменную метода удаления марки, далее в случае если переменная не пуста и не равна null в таблице находится запись со значением равным локальной переменной, после чего запись удаляется. После удаления записи вызывается метод подгрузки данных, а активное значение комбо-бокса сбрасывается на нулевое.



*Рис. 26. Метод удаления марки авто*

Добавление новых моделей автомобилей в таблицу реализовано следующим образом: в локальную переменную modelname записывается значение из текстового поля, в переменную carMark записывается значение из комбо-бокса, после чего в переменную markId записывается индекс марки авто значение которой равно переменной carMark. В случае если переменные modelname и carMark не пусты и не равны null создаётся экземпляр класса Model со значениями name и mark\_id соответствующими modelname и carMark и добавляется в базу данных.

После добавления новой записи вызывается метод подгрузки данных, текстовое поле принимает значение пустой строки, а значение комбо-бокса сбрасывается на нулевое.



*Рис. 27. Метод добавления модели авто*

# **ЗАКЛЮЧЕНИЕ**

Рассмотрим результаты проведенной работы, целью которого являлось создание базы данных и приложения для реализации управления этой базой.

Результатами проведенной работы, целью которой являлось проектирование и создание базы данных, а также приложения для управления данными.

Перечень выполненных работ:

• Исследование предметной области

• Разработка модели системы управления автосалоном

• Проектирование и создание реляционной базы данных

• Проектирование и создание приложения для управления базой данных

В результате выполненных работ удалось освоить принципы построения структуры базы данных и научиться использовать различные инструменты проектирования и программирования. Были освоены новые методы взаимодействия с данными.

В результате всех усилий функционал приложения считается завершенным.

# **СПИСОК ИСПОЛЬЗУЕМЫХ РЕСУРСОВ**

1. Базы данных: учебно-методическое пособие / Г. И. Ревунков, Н. А. Ковалева, Е. Ю. Силантьева [и др.]. —Москва: МГТУ им. Н.Э. Баумана, 2020. —28 с. —ISBN 978-5-7038-5381-8 «MySQL по максимуму» Бэрон Шварц, Вадим Ткаченко, Петр Зайцев.
2. Бычков, М. И. Работа с данными в ADO.NET: учебное пособие / М. И. Бычков. —Новосибирск: НГТУ, 2021. —144 с. —ISBN 978-5-7782-4503-7.
3. Базы данных: учебное пособие / составители Т. Ж. Базаржапова [и др.]. —Улан-Удэ: Бурятская ГСХА им. В.Р. Филиппова, 2022. —84 с.
4. Прокушев, Я. Е. Базы данных: учебное пособие / Я. Е. Прокушев. —2-е изд., доп. —Санкт-Петербург: Интермедия, 2022. —264 с. —ISBN 978-5-4383-0250-6.
5. Алёшкин, А. В. Основы информатики: алгоритм выполнения заданий лабораторных работ в приложении Windows Forms (MS Visual Studio) : учебно-методическое пособие / А. В. Алёшкин, В. М. Шишкин, О. Ю. Медведев. —Киров: ВятГУ, 2016. —63 с.
6. Ильин, И. В. Базы данных : учебное пособие / И. В. Ильин, О. Ю.Ильяшенко. — Санкт-Петербург : СПбГПУ, 2020. — 96 с. — ISBN 978-5-7422-7101-7.— Текст: электронный// Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/192881 (датаобращения: 20.06.2023). — Режим доступа: для авториз. пользователей.
7. Васильева, М. А. Фильтрация набора данных : учебно-методическоепособие / М. А. Васильева, О. А. Тимофеева, К. М. Филипченко. —Москва : РУТ (МИИТ), 2020. — 31 с.— Текст: электронный// Лань :электронно-библиотечнаясистема.—URL:https://e.lanbook.com/book/175827 (дата обращения: 20.06.2023). — Режим доступа: для авториз. пользователей.
8. SQL Задачи и решения [Электронный ресурс] – Режим доступа: <https://fast-project.ru/Moiseenko_SQL.pdf>.
9. Stack Overflow [Электронный ресурс] – Режим доступа: <https://stackoverflow.com>.
10. RU-CENTER. Что такое СУБД? [Электронный ресурс] – Режим доступа: https://www.nic.ru/help/chto-takoe-subd\_8580.html.