

Министерство образования и науки Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет

Имени Александра Григорьевича и Николая Григорьевича  
Столетовых»

(МИ ВлГУ)

Факультет \_\_\_\_\_ ИТР \_\_\_\_\_

Кафедра: \_\_\_\_\_ ПИН \_\_\_\_\_

# Курсовая работа

По: \_\_\_\_\_ Системы управления базами данных \_\_\_\_\_

Тема: Информационная система «Автосалон» \_\_\_\_\_

Руководитель

\_\_\_\_\_

(Оценка)

\_\_\_\_\_ Колпаков А.А. \_\_\_\_\_

(фамилия, инициалы)

\_\_\_\_\_

(подпись)

(дата)

Члены комиссии

Студент \_\_\_\_\_ ПИН-122 \_\_\_\_\_

(группа)

\_\_\_\_\_

(Подпись Ф.И.О.)

\_\_\_\_\_ Павлов Д.В. \_\_\_\_\_

(фамилия, инициалы)

\_\_\_\_\_

(Подпись Ф.И.О.)

\_\_\_\_\_

(подпись)

(дата)

Муром 2024

Целью курсовой работы является разработка приложения, которое будет систематизировать данные, упрощать работу с ними, а также позволит легко получить доступ ко всем нужным таблицам. В работе были использованы теоретические методы в виде изучения научной литературы по созданию концептуальной модели, составлению баз данных и работы с ними в среде PostgreSQL, а также приведение базы данных к трем нормальным формам.

Данная работа содержит анализ технического задания, разработку и создание базы данных, описание доменов, руководство для пользователя и программиста, тестирование приложения и выводы. Также приведены итоги работы, в виде готового приложения. Присутствует список используемой литературы.

Приложение написано на одном языке - русском языке. Эффективность приложения заключается в минимальном количестве ввода информации от пользователя, что сократит количество ошибок ввода информации. Приложение имеет интуитивно понятный интерфейс, который будет понятен всем пользователем. Также приложение не требует специальных системных ресурсов, таких как установка специальных драйверов.

The goal of the course work is to develop an application that will systematize data, simplify working with it, and also allow you to easily access all the necessary tables. We used theoretical methods in the form of studying the scientific literature on creating a conceptual model, compiling databases and working with them in the IBExpert environment, as well as reducing the database to three normal forms.

This work contains an analysis of the technical task, database development and creation, domain description, user and programmer's guide, application testing, and conclusions. The results of the work are also presented in the form of a ready-made application. There is a list of used literature.

The application is written in one language - the Russian language. The effectiveness of the application is to minimize the amount of information input from the user, which will reduce the number of input errors. The app has an intuitive interface that will be clear to all users. Also, the application does not require special system resources, such as installing special drive

Содержание	
Введение .....	5
1 Анализ технического задания .....	7
2 Разработка моделей данных .....	8
2.1 Описание предметной области .....	8
2.2 Описание структуры таблиц .....	10
2.3 Нормализация отношений.....	16
3 Разработка и реализация АИС .....	17
3.1 Интерфейс программного обеспечения.....	17
3.2 Руководство программиста .....	18
3.3 Руководство пользователя.....	19
5 Заключение .....	25
6 Список литературы .....	26
Приложение А .....	27
Приложение Б .....	29
Приложение В.....	30

					МИВУ 09.03.04-13.000ПЗ							
Изм.	Лист	№ докум.	Подп.	Дата	Курсовая работа по теме:  Информационная система  «Автосалон»			Лит.		Лист	Листов	
Студент		Павлов Д.В.							У		4	34
Руков.		Колпаков А.А.						МИ ВлГУ  ПИН-122				
Конс												
Н.контр.												
Утв.												

## Введение

Тематика данной курсовой работы связана с разработкой приложения, для систематизации данных и упрощения работы с ними с помощью создание базы данных в среде программирования Visual Studio и PostgreSQL.

Систематизации данных – это одна из важных проблем, которая не утратила своего значения, поскольку в жизни приходится работать с огромным количеством информации, которую надо обработать за самое короткое время. Получить и работать с информацией о каком-то определенном объекте, или их множестве удобнее, когда вся она представлена в виде таблиц, где с помощью запросов и команд можно будет работать с ней и находить нужную на определенный момент информацию. Поэтому создание приложения для систематизации данных позволит пользователю без больших знаний искать нужную ему информацию за короткое время, добавлять её автоматически, что значительно облегчит работу пользователю.

Базы данных – это совокупность массивов и файлов данных, организованная по определённым правилам, предусматривающим стандартные принципы описания, хранения и обработки данных независимо от их вида. Базы данных могут применяться, как и в учебных заведениях, так и в банках, магазинах, на любых предприятиях. Базы данных – это информация, представленная в электронном виде, где можно осуществить быстрый доступ к информации, а также осуществлять такие действия как объединение, удаление и изменение информации, что позволяет сэкономить время и силы.

Целью курсовой работы является создание интуитивно понятного в использовании приложения с использованием базы данных для пользователя, с возможность внесения, удаления, изменения информации в ней.

Для осуществления обозначенной цели служат следующие задачи:

- Изучение учебной, научной, справочной литературы по теме исследования

					МИВУ 09.03.04-13.000ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

- Написание определенного алгоритма для создания приложения
- Разработка приложения
- Разработка читабельного и интуитивно понятного интерфейса программы
- Тестирование приложения, при введении случайных данных, даже некорректных

					МИВУ 09.03.04-13.000ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

## 1 Анализ технического задания

Согласно техническому заданию на курсовую работу необходимо разработать приложение с использованием базы данных. Приложение должно иметь такие возможности, как редактирование, изменение и удаление информации в таблице в соответствии с правами пользователя. Также удобного поиска и вывода нужной информации пользователю необходимо разработать удобный и понятный интерфейс, позволяющий минимизировать и исключить ошибки при введении информации от пользователя.

Основные возможности программы: занесение необходимой информации в программу, проверка адекватности внесенных данных, вывод результатов на экран, простой и удобный интерфейс для работы пользователя

Требования к программе:

1) Стабильность. Программа должна быть устойчива, даже при введении некорректных данных от пользователя;

2) Целостность базы данных;

3) Быстрый поиск и получение информации по запросам пользователей;

4) Простота обновления данных;

С учетом требований к приложению, база данных должна включать несколько таблиц, которые необходимо связать между собой для обеспечения целостности системы. В рамках данной задачи можно выбрать PostgreSQL, так как это бесплатная объектно-реляционная СУБД. Она реализует реляционную модель данных и поддерживает стандартный язык запросов SQL.

Для создания программы, взаимодействующей с базой данных, необходимо применять объектно-ориентированный подход. Для разработки клиентской части приложения выбран язык C#, поскольку он обладает строгой типизацией, что позволяет предотвратить множество ошибок на этапе компиляции. Кроме того, его динамически типизированные возможности упрощают процесс поиска ошибок в коде, что делает разработку более удобной и эффективной.

					МИВУ 09.03.04-13.000ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

## 2 Разработка моделей данных

### 2. 1 Описание предметной области

Информационная система «Автосалон» содержит автомобили, которые покупатель может изучить по следующим критериям. Информация об авто включает марку, модель, тип, объем, мощность двигателя, привод(передний, задний, полный), тип КПП, год выпуска, пробег автомобиля и цена. Также автомобиль имеет статус: продано, когда автомобиль уже куплен и статус продается, когда авто можно купить.

Чтобы увидеть список доступных для покупки машин, пользователь должен пройти авторизацию, войти в приложение со своим логином и паролем. Или зарегистрироваться, если аккаунта еще нет.

После авторизации пользователь также может просматривать каталог автомобилей и посмотреть авто в наличии у себя.

Для разработки базы данных требуется осуществить следующие задачи:

- 1) Разработать логическое проектирование БД;
- 2) Осуществить физическое проектирование БД;

Основной задачей для первого пункта будет создание ER-модели, определение объектов взаимодействия – сущности, описание их свойств и определение отношений между объектами.

В данной диаграмме будет выделено десять сущностей.

Первая сущность называется «Марка». Эта сущность хранит информацию о марке машины

Второй сущностью является «Модель». Данная сущность содержит информацию о модели машины и ее марки

Третья сущность «Привод». В этой сущности находится информация о приводе машины

Четвертая сущность «Автомобиль». В этой сущности находится информация о характеристиках машины, о ее пробеге, цене, статусе и изображения самого авто

					МИВУ 09.03.04-13.000ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		



Пятой сущностью является «Характеристика». В этой сущности находится информация о модели машины, двигателе, приводе и КПП. А также о годе выпуска машины.

Шестая сущность «КПП». Здесь находится тип коробки передач

Седьмая сущность «Двигатель». Здесь хранятся данные о типе, объёме и мощности двигателя

Восьмая сущность «Продажа». Здесь хранятся данные о проданных автомобилях

Девятой сущностью является «Сотрудник». Данная сущность хранит данные о сотрудниках автосалона

Девятой сущностью является «Покупатель». Данная сущность хранит данные о покупателях

В результате определения сущностей, их атрибутов и установление связей, была разработана диаграмма, изображенная на рисунке 1.

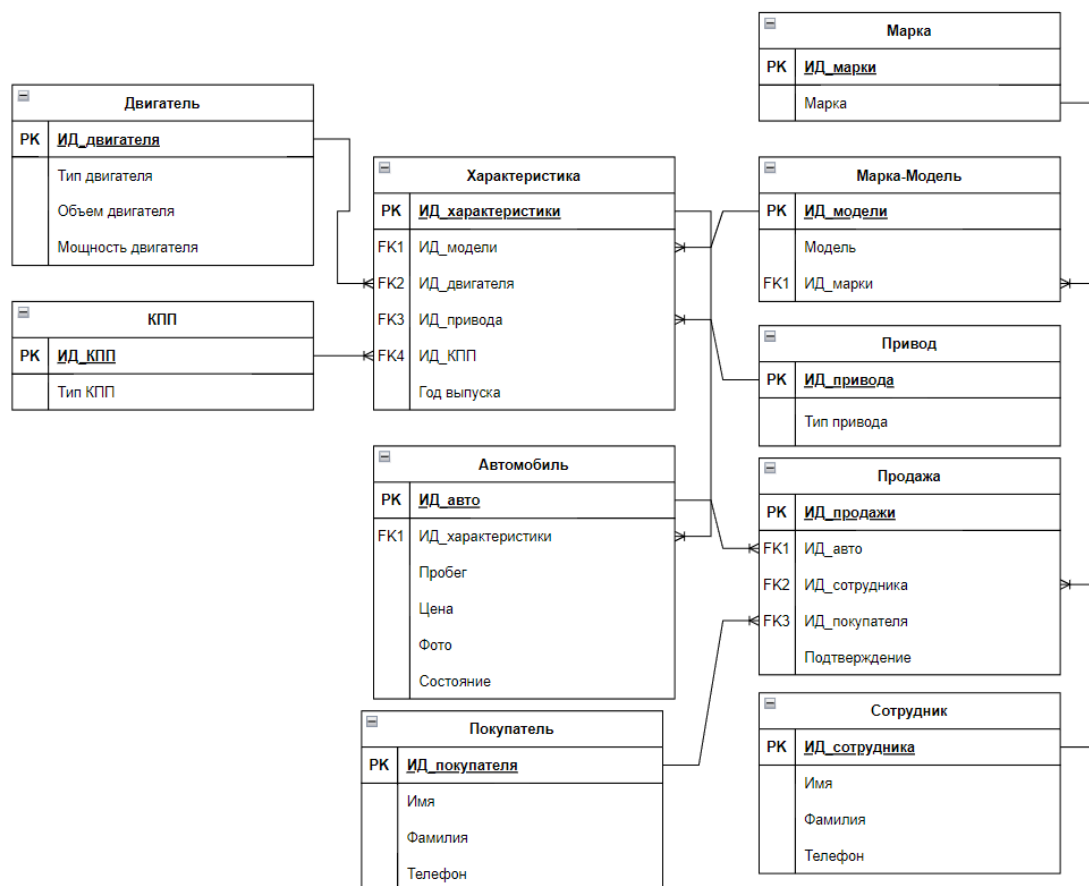


Рисунок 1– ER-модель базы данных «Автосалон»

## 2.2 Описание структуры таблиц

Для создания физической модели данных будут использованы средства PostgreSQL. На этом этапе описаны типы данных, которые используются в базе данных, имена таблиц и т.д. А также описание отношений и доменов.

Таблица «Двигатель» будет содержать следующие поля:

ИД\_двигателя (ID) = { натуральное число, не пустое }

Тип двигателя (TYPE) = { символьный массив длиной 255 символов, не пустое };

Объем двигателя (CAPACITY) = { вещественное число, не пустое };

Мощность двигателя (POWER) = { натуральное число, не пустое };

Описание атрибутов:

Атрибут «ИД\_двигателя» содержит уникальный идентификатор двигателя

Атрибут «Тип двигателя» содержит информацию о типе двигателя(бензин, дизель, электрический)

Атрибут «Объем двигателя» содержит объем двигателя

Атрибут «Мощность двигателя» содержит мощность двигателя

В данной таблице идет следующие разграничение целостности данных:  
Поле ID является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Таблица «КПП» будет содержать следующие поля:

ИД\_КПП (ID) = { натуральное число, не пустое };

Тип КПП (TYPE) = { символьный массив длиной 255 символов, не пустое};

Описание атрибутов:

Атрибут «ИД\_КПП» содержит уникальный идентификатор КПП

Атрибут «Тип КПП» содержит тип коробки передач(механика, робот, автомат, полуавтомат)

В данной таблице идет следующие разграничение целостности данных:  
Поле ID является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Таблица «Характеристика» будет содержать следующие поля:

ИД\_Характеристики (ID) = { натуральное число, не пустое };

ИД\_Модели (MODEL\_ID) = { натуральное число, не пустое };

ИД\_Двигателя (ENGINE\_ID) = { натуральное число, не пустое };

ИД\_Привода (DRIVE\_ID) = { натуральное число, не пустое };

ИД\_КПП (TRANSMISSION\_ID) = { натуральное число, не пустое }

Год выпуска (YEAR) = { натуральное число, не пустое }

Описание атрибутов:

Атрибут «ИД\_Характеристики» содержит уникальный идентификатор характеристики

Атрибут «ИД\_Модели» содержит идентификатор модели

Атрибут «ИД\_Двигателя» содержит идентификатор двигателя

Атрибут «ИД\_Привода» содержит идентификатор привода

Атрибут «ИД\_КПП» содержит идентификатор КПП

Атрибут «Год выпуска» содержит год выпуска машины

В данной таблице идет следующие разграничение целостности данных:  
Поле ID является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Внешними ключами являются поля MODEL\_ID, ENGINE\_ID, DRIVE\_ID и TRANSMISSION\_ID, которые ссылаются на таблицы «Модель», «Двигатель», «Привод» и «КПП» соответственно.

Таблица «Автомобиль» будет содержать следующие поля:

ИД\_авто (ID) = { натуральное число, не пустое };

ИД\_характеристики (CHARACTERISTIC\_ID) = { натуральное число, не пустое };

Пробег (MILEAGE) = { натуральное число, не пустое };

Цена (PRICE) = { вещественное число, не пустое };

					МИВУ 09.03.04-13.000ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

Фото (PHOTO) = { массив двоичных данных, не пустое };

Состояние (STATE) = { символьный массив длиной 255 символов, не пустое};

Описание атрибутов:

Атрибут «ИД\_ авто» содержит уникальный идентификатор автомобиля

Атрибут «ИД\_характеристики» содержит идентификатор характеристики

Атрибут «Пробег» содержит информацию о пробеге

Атрибут «Цена» содержит информацию о цене

Атрибут «Фото» содержит фото автомобиля

Атрибут «Состояние» содержит информацию о состоянии авто(продан, продается)

В данной таблице идет следующие разграничение целостности данных:  
Поле ID является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Внешним ключом является поле CHARACTERISTIC\_ID, которое ссылается на таблицу «Характеристика».

Таблица «Марка» будет содержать следующие поля:

ИД\_марки (ID) = { натуральное число, не пустое};

Марка (NAME) = {символьный массив длиной 255 символов, не пустое};

Описание атрибутов:

Атрибут «ИД\_марки» содержит уникальный идентификатор марки

Атрибут “ Марка ” содержит название марки

В данной таблице идет следующие разграничение целостности данных:  
Поле ID является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Таблица «Марка-Модель» будет содержать следующие поля:

ИД\_модели (ID) = { натуральное число, не пустое }

Модель (NAME) = {символьный массив длиной 255 символов, не пустое};

					МИВУ 09.03.04-13.000ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

ИД\_марки(BRAND\_ID) = { натуральное число, не пустое };

Описание атрибутов:

Атрибут «ИД\_модели» содержит уникальный идентификатор модели

Атрибут «Модель» содержит название модели

Атрибут «ИД\_марки» содержит идентификатор марки

В данной таблице идет следующие разграничение целостности данных:

Поле ID является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Внешним ключом является поле BRAND\_ID, которое ссылается на таблицу «Марка».

Таблица «Привод» будет содержать следующие поля:

ИД привода (ID) = { натуральное число, не пустое }

Тип привода (TYPE) = { символьный массив длиной 255 символов, не пустое};

Описание атрибутов:

Атрибут «ИД привода» содержит уникальный идентификатор привода

Атрибут «Тип привода» содержит тип привода (передний, задний, полный)

В данной таблице идет следующие разграничение целостности данных:

Поле ID является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Таблица «Продажа» будет содержать следующие поля:

ИД продажи (ID) = { натуральное число, не пустое }

ИД авто (AUTO\_ID) = { натуральное число, не пустое };

ИД сотрудника (STAFF\_ID) = { натуральное число, не пустое };

ИД покупателя (CUSTOMER\_ID) = { натуральное число, не пустое };

Подтверждение (STAFFCONFIRM) = { логический тип данных. Не пустое };

Описание атрибутов:

Атрибут «ИД продажи» содержит уникальный идентификатор продажи

					МИВУ 09.03.04-13.000ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

Атрибут «ИД авто» содержит идентификатор авто

Атрибут «ИД сотрудника» содержит идентификатор сотрудника

Атрибут «ИД покупателя» содержит идентификатор покупателя

Атрибут «Подтверждение» содержит информацию о подтверждении покупки

В данной таблице идет следующие разграничение целостности данных:  
Поле ID является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Внешними ключами являются поля AUTO\_ID, STAFF\_ID и CUSTOMER\_ID, которые ссылаются на таблицы «Автомобиль», «Сотрудник» и «Покупатель» соответственно.

Таблица «Сотрудник» будет содержать следующие поля:

ИД сотрудника (ID) = { натуральное число, не пустое }

Имя (FIRST\_NAME) = { символьный массив длиной 255 символов, не пустое};

Фамилия (LAST\_NAME) = { символьный массив длиной 255 символов, не пустое};

Телефон (PHONE) = { символьный массив длиной 255 символов, не пустое};

Описание атрибутов:

Атрибут «ИД сотрудника» содержит уникальный идентификатор сотрудника

Атрибут «Имя» содержит имя сотрудника

Атрибут «Фамилия» содержит фамилию сотрудника

Атрибут «Телефон» содержит номер телефона

В данной таблице идет следующие разграничение целостности данных:  
Поле ID является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Таблица «Покупатель» будет содержать следующие поля:

ИД покупателя (ID) = { натуральное число, не пустое }

					МИВУ 09.03.04-13.000ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		



## 2.3 Нормализация отношений.

При приведении таблиц к нормальным формам, необходимо знать основные правила каждой формы.

1 нормальная форма: необходимость неделимости значения в каждом поле строки. Каждая строка должна быть уникально и не зависеть от других строк. Должен быть определен первичный ключ

2 нормальная форма: условие этой формы - является отсутствие зависимости неключевых полей от части составного ключа. Все таблицы должны находиться в первой нормальной форме

3 нормальная форма: условием третьей нормальной формы является то, что не первичные атрибуты не должны быть зависимы друг от друга.

На примере таблицы «Автомобиль» докажем нахождения таблиц в нормальных формах:

Отношение Автомобиль (ИД\_авто, ИД\_характеристики, пробег, цена, фото, состояние) находится в первой нормальной форме, т.к. все атрибуты данного отношения не являются составными, не зависят друг от друга. Ключом отношения будет являться атрибут ИД авто, т.к. данный атрибут является уникальным для каждого автомобиля. Также данное отношение находится во второй нормальной форме, т.к. все атрибуты полностью зависят от первичного ключа, ключ не является составным. Отношение находится в третьей нормальной форме, т.к. каждый атрибут данной таблицы не зависит друг от друга.

Все эти условия соблюдаются и для других таблиц.



### 3 Разработка и реализация АИС

#### 3.1 Интерфейс программного обеспечения

В данном пункте приведена информация о реализации приложения для информационной системы «Автосалон», разрабатываемой в данной курсовой работе.

Приложение для информационной системы «Автосалон» используются следующие компоненты: Button, Label, TabControl, DataGridView, RadioButton, Combobox, CheckBox.

Компонент Button - кнопка, с помощью которой пользователь взаимодействует с окнами и редактирует информацию.

Компонент Label представляет описание некоторых объектов, для помощи пользователю.

С помощью компонента dataGridView происходит вывод информации, которая содержится в базе данных

Компонент Combobox отвечает за выбор пользователя.

Компонент CheckBox отвечает за выгрузку информации.

Компонент TabControl отвечает за перемещение между вкладками программы в одно форме.

Приложение «Автосалон» состоит из 6 окон. Главным окном является 1 окно, в котором от выбора пользователя зависит следующее открытие окон(2 или 3). Данные окна связаны с 1 и без выбора и подтверждения пользователя не смогут открыться.

Первое окно, которое появится перед пользователем – окно авторизации.(Приложение В, Рисунок В.1) Состоит из полей для ввода логина и пароля и двух кнопок – кнопка входа и кнопка авторизации.

Если человек заходит как пользователь, то открывается окно 2. В данном окне есть три вкладки для пользователя. В первой вкладке пользователь может изучить машины, представленные для продажи. Во второй увидеть список тех же машин, представленных в табличном варианте – для тех, кто знает, что

хочет купить. И третья вкладка – мои авто. Здесь пользователь может посмотреть свои купленные машины.

Если человек заходит под логином и паролем администратора, то ему откроется третье окно, которое содержит компонент dataGridView для вывода таблиц, соответствующие кнопки для управления информацией в таблицах и выпадающий список с названиями таблиц.(Приложение В, Рисунок В.2) Администратор может удалить, изменить или обновить информацию в базе данных. Для добавления данных, необходимо нажать кнопку «добавить», где будет представлено еще одно окно, где администратор может внести нужную информацию и подтвердить операцию. Для удаления информации, администратору необходимо выделить нужную строку в базе данных и нажать кнопку «удалить». Для изменения информации есть кнопка «изменить».

### 3.2 Руководство программиста

Для подключения к базе данных используется класс NpgsqlConnection для установления соединения с сервером PostgreSQL. Подключение к базе данных сделано в отдельном методе ConnectToDB().

```
public void ConnectToDB()  
{  
    connectionString = "Server=localhost;Port=5432;User  
Id=postgres;Password=postgres;Database=AutoMarketDB;";  
    conn = new NpgsqlConnection(connectionString);  
    conn.Open();  
}
```

Для вывода информации из таблиц используется следующая конструкция: SELECT <Столбцы> FROM <Таблица>;

Например, для вывода определенной марки из таблицы соответствует следующий запрос:

```
command = $"SELECT id FROM brand where name = '{bran}'";
```

Для изменения и удаления данных используются следующие конструкции соответственно:

```

command = $"update sale set staff_id = {StaffID}, staffconfirm =
'{staffconfirm}' where id = {id}";
command = $"delete from sale where id = {id}";
cmd = new NpgsqlCommand(command, conn);

```

Для выполнения всех команд используется метод `ExecuteNonQuery` - это метод объекта `Command`, который выполняет команды, не возвращающие результирующие наборы. Для возвращения результатов используется метод `ExecuteReader` - метод класса `SqlCommand`, который выполняет SQL-выражение и возвращает строки из таблицы. Подходит для SQL-выражения `SELECT`.

### 3.3 Руководство пользователя

Чтобы запустить программу, необходимо открыть файл `AutoMarket.exe`. После запуска приложения откроется окно авторизации, где будут два поля ввода – логин и пароль, а также две кнопки войти и зарегистрироваться. После введения данных и нажатия кнопки войти откроется окно программы с данными. А кнопка зарегистрироваться позволяет создать свой профиль для входа в систему. Если пользователь, не зарегистрированный в системе, попытается войти в приложение, ему будет показано окно с предупреждением о невозможности входа без учетного профиля.

После успешного входа в приложение пользователь может просматривать представленные автомобили двумя разными способами. На главном окне находятся три вкладки, которые позволяют перемещаться между разными окнами.

В первой вкладке представлено фото автомобиля, его характеристики, а также цена. Между автомобилями можно переключаться, нажав соответствующие кнопки в виде стрелочек. Если автомобиль заинтересовал, то есть кнопка «Заказать», которая позволит отправить запрос сотруднику

автосалона. Если автомобиль уже оформлен другим человеком, то будет соответствующее предупреждение. Если автомобиль еще не оформлен, то для покупки его необходимо дождаться подтверждения сотрудника автосалона.

Во второй вкладке находится такой же список автомобилей, но уже в табличном представлении. Из этой вкладки также можно заказать автомобиль, выбрав его из списка и нажав кнопку «Заказать».

Третья вкладка позволяет просмотреть ранее купленные пользователем автомобили.

#### 4 Тестирование АИС

Приложение, которое будет удовлетворять всем условиям проверки, стабильно работать при введении разных данных от пользователя, а так же защищать информацию, доступную только администратору – является целью данной курсовой работы.

Чтобы проверить приложение на его стабильность, было проведено тестирование по следующим пунктам:

- 1) Проверка правильности введенных данных во время авторизации
- 2) Проверка правильности добавления данных в таблицу от лица администратора
- 3) Проверка правильности изменения данных в таблице от лица администратора
- 4) Проверка правильности удаления данных из таблицы от лица администратора
- 5) Проверка повторного заказа авто
- 6) Проверка входа в систему под неизвестным пользователем

Для проверки первого условия необходимо внести некорректные данные в окно авторизации(Рисунок 4). И после введения данных нажать кнопку «войти».

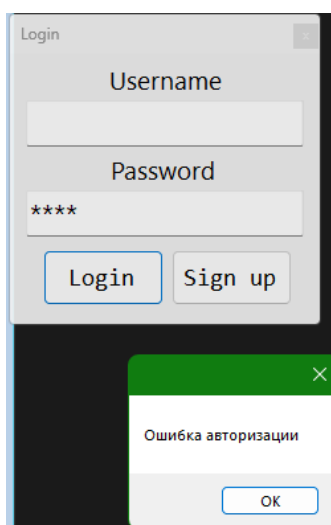


Рисунок 4 – Ошибка при вводе «пробелов»

В ходе данной проверки, можно увидеть, что при введении пустых данных, форма не будет открываться.

Проверка условия 2. Чтобы проверить данное условие, необходимо нажать на кнопку «Insert auto», внести данные и подтвердить их (Рисунок 5).

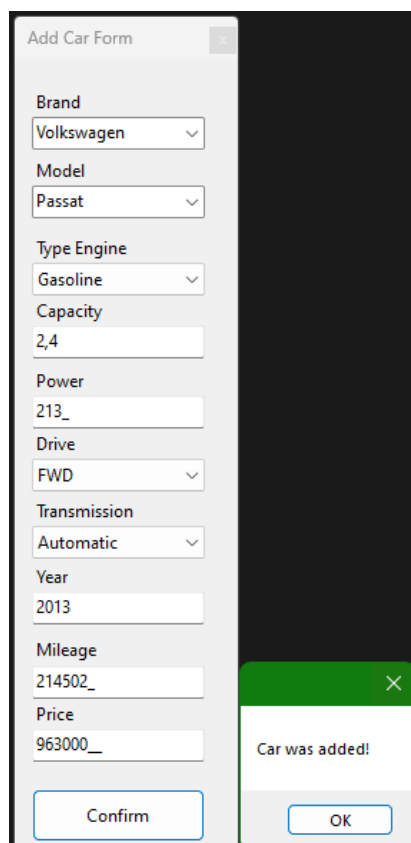
The screenshot shows a mobile application interface titled "Add Car Form". It contains several input fields: "Brand" (dropdown menu with "Volkswagen" selected), "Model" (dropdown menu with "Passat" selected), "Type Engine" (dropdown menu with "Gasoline" selected), "Capacity" (text input with "2,4"), "Power" (text input with "213\_"), "Drive" (dropdown menu with "FWD" selected), "Transmission" (dropdown menu with "Automatic" selected), "Year" (text input with "2013"), "Mileage" (text input with "214502\_"), and "Price" (text input with "963000\_"). At the bottom of the form is a blue "Confirm" button. To the right of the form, a green notification box with a white "X" icon displays the message "Car was added!". Below the notification box is a white "OK" button.

Рисунок 5 – Скриншот выполнение программы при добавлении новых данных

В ходе этой проверки, новые данные были добавлены успешно. Ошибок выявлено не было.

Чтобы проверить третье условие, нужно выбрать данные и нажать кнопку «Update sale». Потом подтвердить данные.

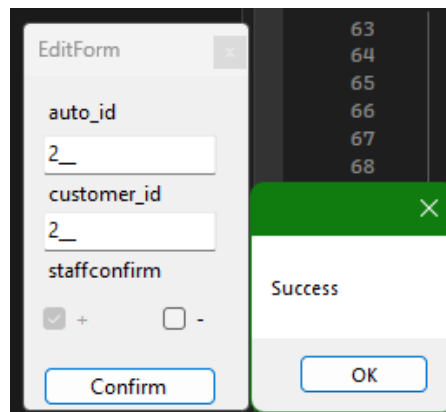


Рисунок 6 – Скриншот выполнение программы при изменении данных

В ходе этой проверки, ошибок обнаружено не было. Программа успешно изменила данные

Проверкой 4 условия является удаление данных из таблицы. Для проверки этого условия необходимо выбрать данные, которые надо удалить и нажать кнопку «Delete sale».

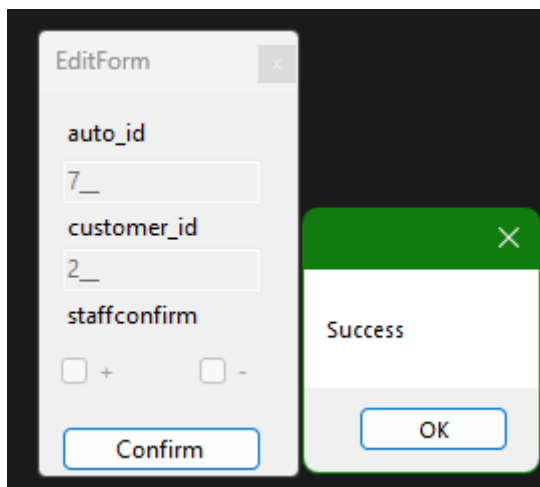


Рисунок 7 – Скриншот выполнение программы при удалении данных

Программа сработала успешно, ошибок не обнаружено.

Чтобы проверить пятое условие, необходимо зайти в программу от лица пользователя, выбрать авто и нажать кнопку «Заказать»(Рисунок 8)

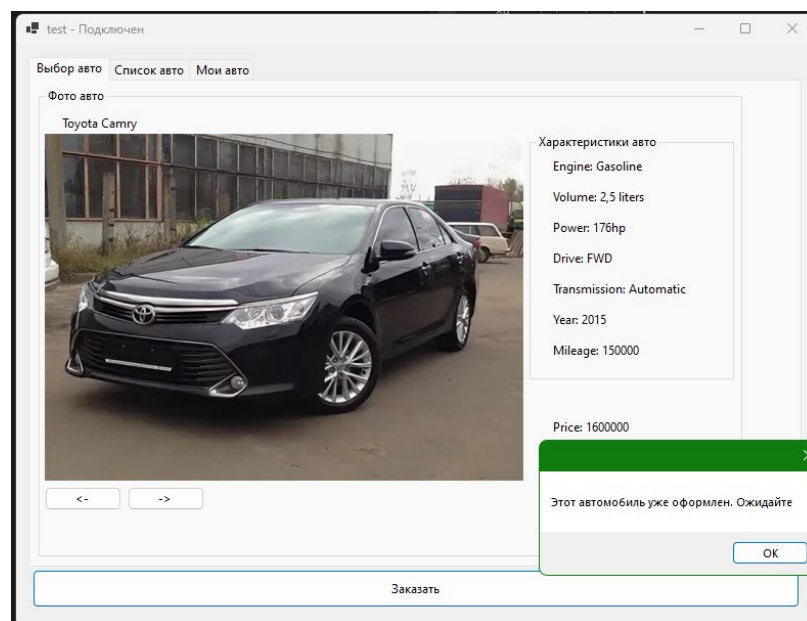


Рисунок 8 – Скриншот выполнение программы при повторном заказе авто

В ходе проверки программа сработала стабильно, ошибок не обнаружено.

Проверкой шестого условия, служит вход в систему под неизвестным пользователем(Рисунок 9)

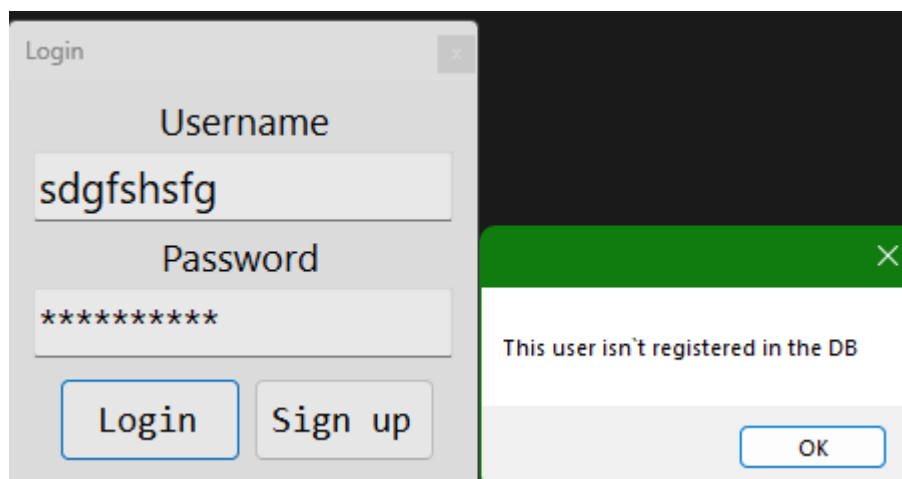


Рисунок 9 – Скриншот выполнение программы при редактировании отзыва



## 5 Заключение

В ходе данной работы были выполнены все поставленные задачи, была разработана информационная система «Автосалон», которая предназначена для быстрого действия обработки информации, а также облегчить доступ к чтению и редактированию большого количества информации и автоматизировать процесс заполнения нужных данных. Реализованная программа имеет интуитивно понятный интерфейс, что облегчит пользователям работу с данными.

Для программы был проведен ряд тестов, которые охватывают весь функционал программы. Тесты показали стабильную и правильную работу приложения, а также работоспособность всех компонентов приложения

					МИВУ 09.03.04-13.000ПЗ	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		

## 6 Список литературы

1. Краткий обзор языка C# [Электронный ресурс] // Microsoft – документация. 2020 URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/> (дата обращения 18.11.2024)
2. Кратка характеристика инфологической, даталогической и физической моделей [Электронный ресурс] // sdamzavas.net 2020 URL: <https://sdamzavas.net/3-6241.html> (дата обращения 25.11.2024)
3. PostgreSQL обзор [Электронный ресурс] // Wikipedia свободная энциклопедия 2011-2020. URL: <https://ru.wikipedia.org/wiki/PostgreSQL> (дата обращения 01.12.2024)
4. SQL запросы быстро. Часть 1 [Электронный ресурс] // 2006 – 2020 «Habr». URL: <https://habr.com/ru/post/480838/> (дата обращения 05.12.2024)

# Приложение А

## Заполненные данными таблицы

Таблица А.1 auto

3	3	56200	6300000.00
4	4	378000	400000.00
5	5	40633	1200000.00
7	7	214502	963000.00
2	2	150000	1600000.00
3	3	56200	6300000.00
4	4	378000	400000.00
5	5	40633	1200000.00
7	7	214502	963000.00
2	2	150000	1600000.00
3	3	56200	6300000.00
4	4	378000	400000.00

Таблица А.2 brand

1	"Toyota"
2	"BMW"
3	"Mercedes-Benz"
4	"Audi"
5	"Ford"
6	"Honda"
7	"Hyundai"
8	"Nissan"
9	"Volkswagen"
10	"Kia"
11	"Tesla"

Таблица А.3 brandmodel

1	"Camry"	1
2	"Corolla"	1
3	"X5"	2
4	"C-Class"	3

Таблица А.4

characteristic

2	1	14	1	1	2015
3	3	2	1	1	2017
4	4	15	2	2	2010
5	9	16	2	3	2019
6	13	17	3	3	2
7	10	18	1	1	2013

Таблица А.5 engine

1	"Gasoline"	2	150
2	"Diesel"	3	250
3	"Electric"	0	200
4	"Hybrid"	1.5	120
5	"Gasoline"	2.5	180
6	"Diesel"	1.8	140
7	"Electric"	0	300
8	"Gasoline"	3	220
9	"Diesel"	2.2	170
10	"Hybrid"	2	160
11	"Diesel"	2.6	332
12	"Diesel"	2.5	123
13	"Electric"	0	674
14	"Gasoline"	2.5	176
15	"Gasoline"	1.8	130
16	"Hybrid"	0	130
17	"Gasoline"	0.1	2
18	"Gasoline"	2.4	213

Приложение Б

Текст программы

<https://github.com/dexterfredo/KursSUBD>

					МИВУ 09.03.04-13.000ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

## Приложение В

### Скриншоты программы

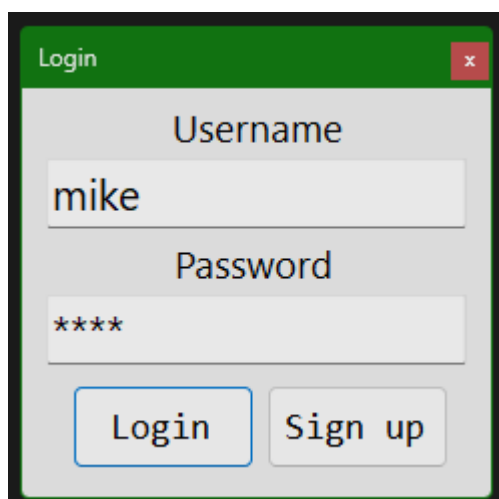


Рисунок В.1– Скриншот окна авторизации

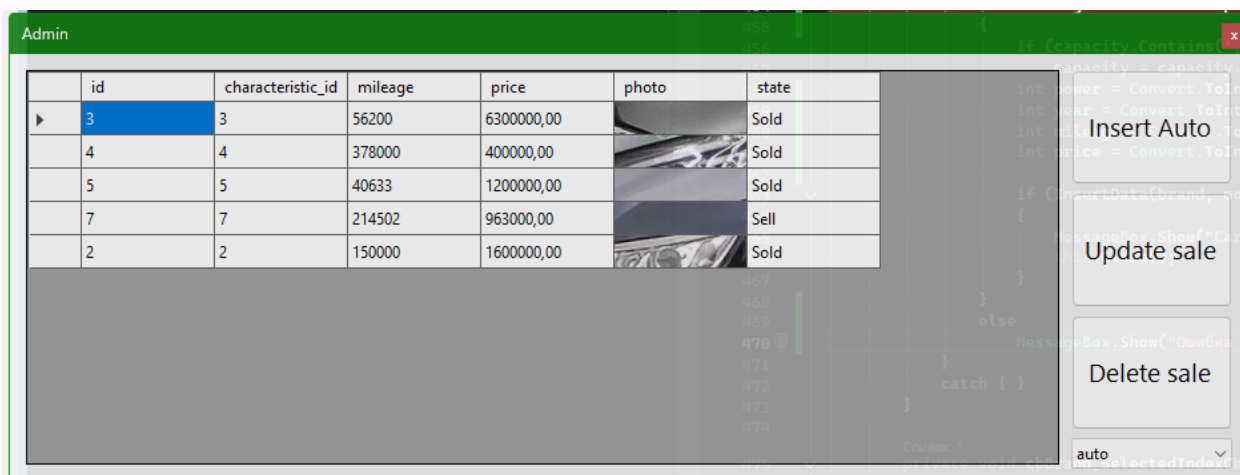


Рисунок В.2 – окно администратора

Рисунок В.3 – изменить данные от лица Администратора

Рисунок В.4 – добавить данные от лица Администратора

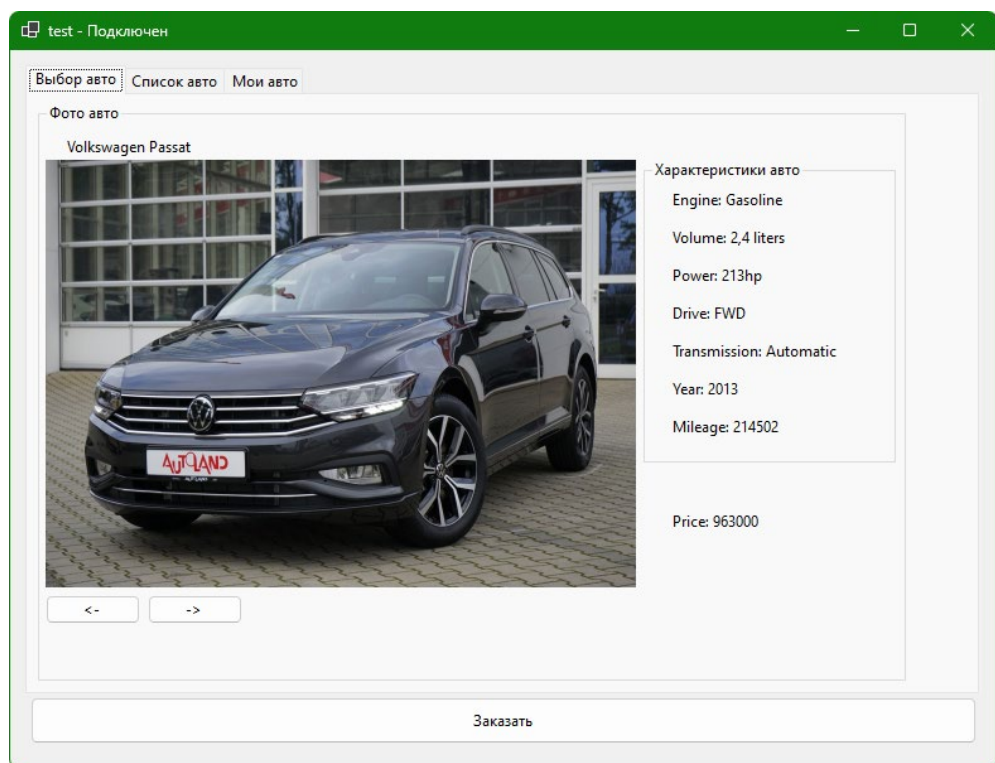


Рисунок В.5 – окно пользователя

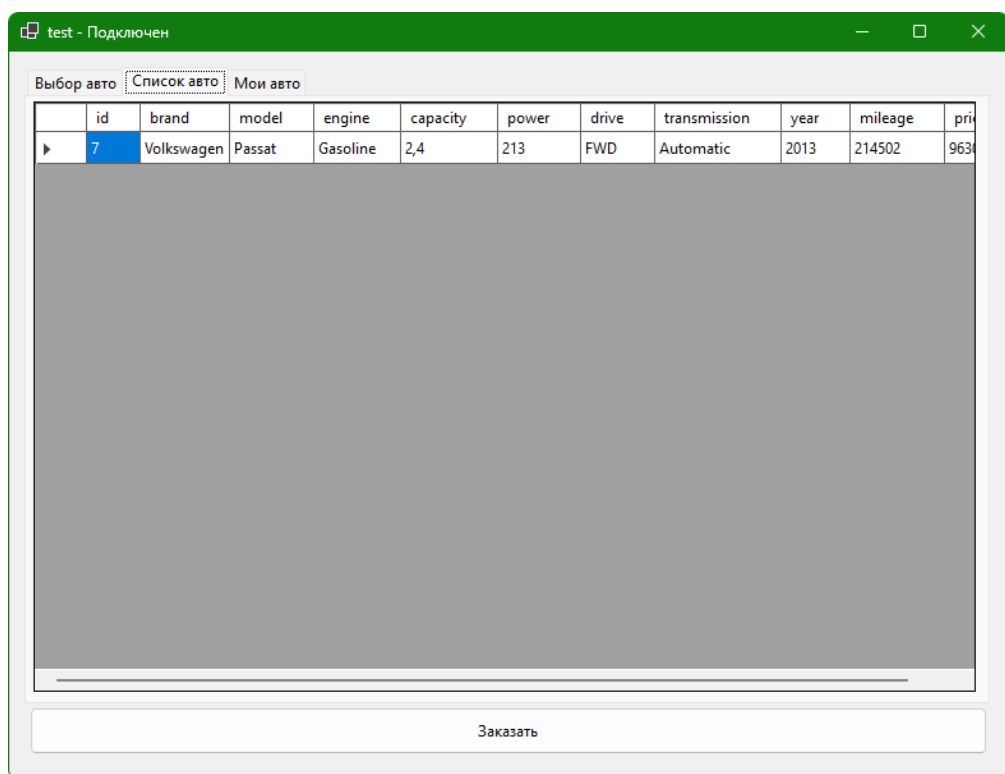


Рисунок В.6– просмотр таблицы всех авто



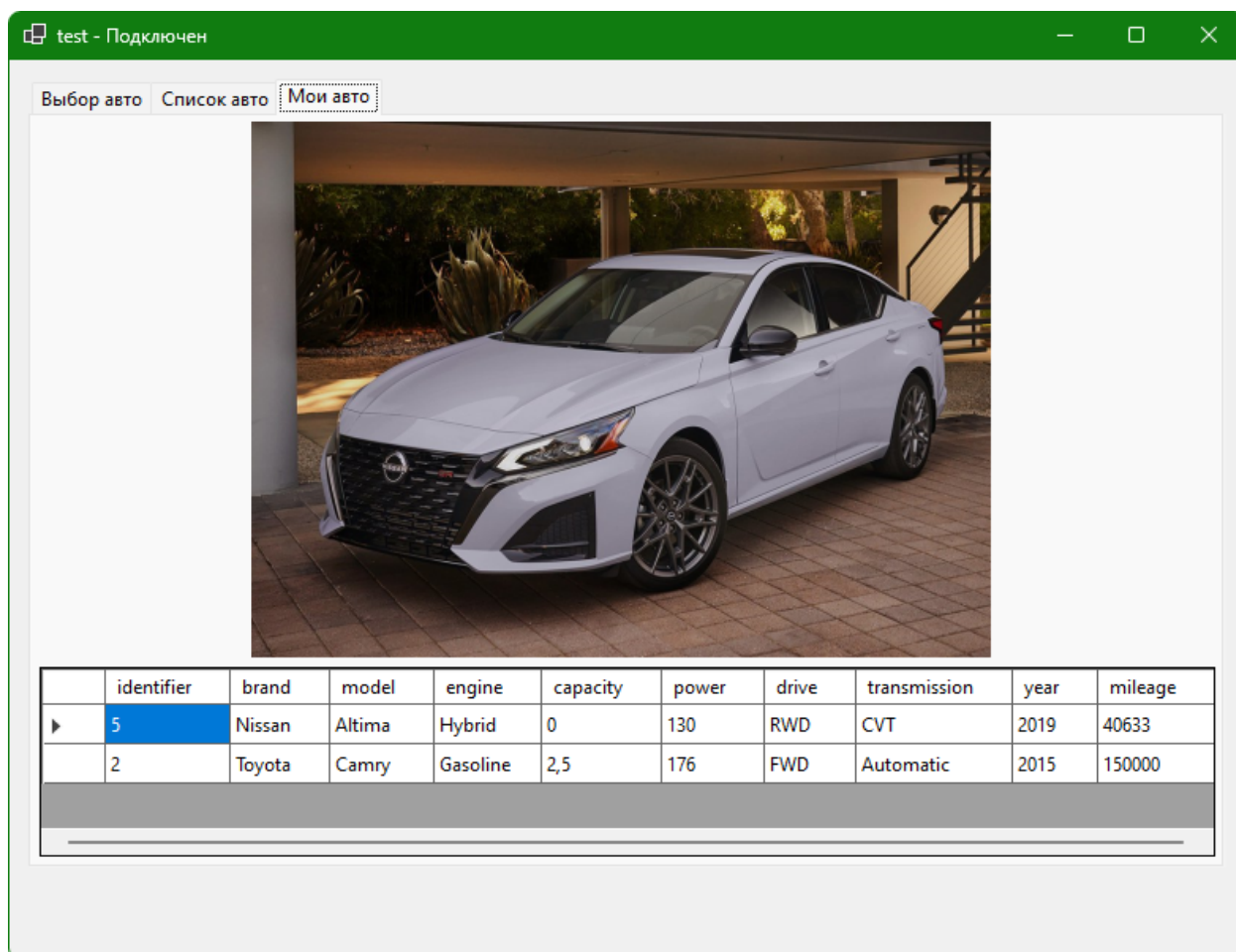


Рисунок В.7– просмотр таблицы своих авто

Form Registry

×

Firstname

Lastname

Phone

Username

Password

Confirm

Рисунок В.8 – окно регистрации