

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

канд. техн. наук, доцент

должность, уч. степень, звание

подпись, дата

Н.В. Кучин

инициалы, фамилия

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №10

РАЗРАБОТКА ПРОСТЕЙШЕГО ПРИЛОЖЕНИЯ НА ОСНОВЕ
ТЕХНОЛОГИИ «КЛИЕНТ-СЕРВЕР»

по курсу: СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4941

подпись, дата

Н. С. Горбунов

инициалы, фамилия

Санкт-Петербург 2022

Цель работы: изучить основные принципы взаимодействия приложений, разработанных в архитектуре «клиент-сервер», реализовать простейшее клиентское приложение, осуществляющее доступ к базе данных по технологии ODBC (или другой технологии взаимодействия с базами данных), изучить основные принципы работы клиентского приложения с API ODBC и с другими технологиями доступа к базам данных.

Тема: «Автосалон». Вариант 7

Автосалон	Ввод и редактирование данных об автомобилях, их комплектации и сроках поставки.	Продавец: заключение договоров о покупке автомобиля, заказ автомобиля нужной комплектации.
-----------	---	--

Используемые технологии и платформа реализации

- СУБД: PostgreSQL;
- ОС: Windows / GNU/Linux / MacOS;
- язык программирования: C#;
- фреймворк: ASP .NET Core;
- тип приложения: веб-приложение.

Подробное описание клиент-серверного приложения:

Для разработки веб-приложения были использованы следующие программные средства:

- язык программирования C# и его фреймворк ASP .NET Core
- система управления базами данных PostgreSQL

ASP.NET Core является кроссплатформенной, высокопроизводительной средой с открытым исходным кодом для создания современных облачных приложений, подключенных к Интернету.

Описание используемых методов взаимодействия с БД

База данных подключается в классе CarDbContext проекта. Модель базы данных описана в файлах, находящихся в директории DbModel.

Для доступа к БД использована ORM EntityFrameworkCore.

Entity Framework (EF) Core — это простая, кроссплатформенная и расширяемая версия популярной технологии доступа к данным Entity Framework с открытым исходным кодом.

EF Core может использоваться в качестве объектно-реляционного модуля сопоставления (O/RM), который:

Позволяет разработчикам .NET работать с базой данных с помощью объектов .NET.

Устраняет необходимость в большей части кода для доступа к данным, который обычно приходится писать.

EF Core поддерживает множество систем баз данных.

Реляционная схема разработанной БД

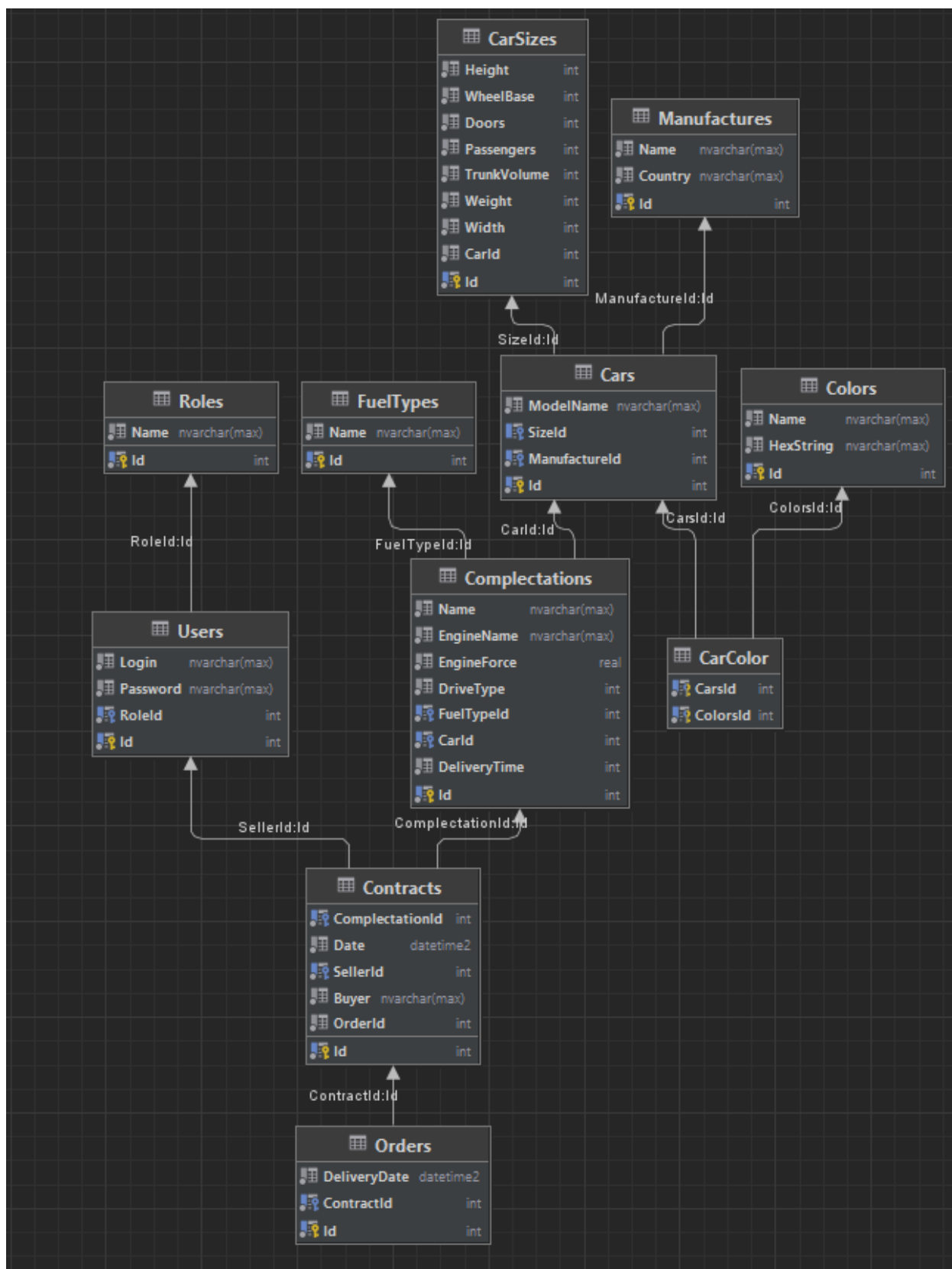


Рисунок 1 – реляционная схема базы данных

Описания применения приложения со снимками экрана

CarShowRoom Home Страница Администратора Страница Продавца

Вход на сайт

[Регистрация](#)

Введите Login

Введите пароль

Войти

Рисунок 2 – страница авторизации

CarShowRoom Home Страница Администратора Страница Продавца

Автосалон

Все модели

[Corolla](#)

Рисунок 3 – главная страница

CarShowRoom Home Страница Администратора

Добавить цвет

Имя

Черный

Нек-строка

#000000

Создать

Рисунок 4 – добавление цвета

CarShowRoom Home Страница Администратора

Добавить производителя

Имя

Toyota

Страна

Япония

Создать

Рисунок 5 – добавление производителя

Добавить тип Топлива

Название типа

95

Создать

Рисунок 6 – добавление типа топлива

Добавить модель

Имя

Corolla

Производитель

Toyota

Цвет

Название цвета	Выбрано
Черный	<input type="checkbox"/>
Белый	<input checked="" type="checkbox"/>

Высота

1435

Количество колес

4

Количество дверей

4

Максимум пассажиров

5

Объем багажника

470

Масса

1370

Ширина

1780

Создать

Рисунок 7 – добавление модели автомобиля

Модель Corolla

Производитель Toyota
Размеры
Высота 1435
Количество колес 4
Количество дверей 4
Максимум пассажиров 5
Объем багажника 470
Масса 1370
Ширина 1780
Цвета:

Название цвета	Hex
Черный	#000000

Комплектации:

Название
Стандартная

Рисунок 8 – просмотр добавленной модели

Новый договор

Модель [Corolla](#)

ФИО покупателя

Горбунов Никита Сергеевич

Комплектация [Стандартная](#) ▼

Создать

Рисунок 9 – оформление нового договора продавцом

Договор №1

Модель [Corolla](#)

ФИО покупателя Горбунов Никита Сергеевич

Продавец [seller@mail.ru](#)

Комплектация [Стандартная](#)

Сделать заказ

Рисунок 10 – оформленный договор

Страница продавца

Все договоры:

[Договор №1](#)

Все заказы:

Заказ №1 по договору [№1](#)

Рисунок 11 – Страница продавца

Приложение реализует ролевую модель:

- администратор может добавлять новые модели автомобилей, их комплектации, производителей и т.п.
- продавец может оформлять новые договоры и заказы

Программный код

Так как исходный код очень большой, я решил не представлять его в полном объеме в отчете, но оставляю ссылку на GitHub с репозиторием приложения:

<https://github.com/NOoBODDY/SPOWebApp>

Так же для теста работоспособности, приложение было опубликовано на выделенном сервере, получить к нему доступ можно по ссылке

<http://offmysoap1.fvds.ru:7030/>

Данные для входа:

- Администратор: admin@mail.ru Пароль: 123456
- Продавец: seller@mail.ru Пароль: 123456

Вывод: в ходе выполнения лабораторной работы были изучены основные принципы взаимодействия приложений, разработанных в архитектуре «клиент-сервер», реализовано клиентское приложение, осуществляющее доступ к базе данных по технологии EntityFrameworkCore