МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**«Челябинский государственный университет»**

**(ФГБОУ ВО «ЧелГУ»)**

Институт информационных технологий

Кафедра информационных технологий и экономической информатики

**КУРСОВАЯ РАБОТА**

по дисциплине «Базы и хранилища данных»

Разработка базы данных для предметной области «Автостоянка “AutoCar”»

Выполнил студент Шапошников Даниил Валерьевич

Группы ИТЗ-201

заочной формы обучения

направления подготовки

02.03.02 Фундаментальная информатика и информационные технологии

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

|  |  |
| --- | --- |
|  | Научный руководитель  Фамилия, имя, отчество\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Должность\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Ученая степень \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Ученое звание \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_\_\_г. |

Челябинск

2023

**ЛИСТ ЗАДАНИЯ**

1. **Провести анализ предметной области по следующему описанию:**

В городе существует круглосуточная отапливаемая автостоянка “AutoCar” с установленной автоматизированной системой наблюдения, которая предоставляет гарантии безопасности автомобиля посетителя на своей территории, за счет того, что контролирует все выходы из комплекса автостоянки и сохраняет время автовладельцев на прогрев автомобиля, также в добавок ко всему закрытая автостоянка дает возможность избежать загрязнения от природных погодных условий. Клиент оплачивает место автостоянки ежемесячно.

**Перечень входных (первичных) документов:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ФИО клиента | Марка автомобиля | парковочный номер | Сумма оплаты | Дата оплаты | цена места | дата начисления оплаты | номер телефона | дата рождения | номер автомобиля | год выпуска |
| Иванов Иван Иванович | Mazda | 174 | 500 | 20/04/07 | 400 | 19/04/07 | 8935123456 | 20/04/1993 | А888НА174 | 1965 |
| Петров Алексей Петрович | LADa | 53 | 100 | 25/04/07 | 100 | 20/04/07 | 8241459673 | 23/04/1986 | O812OP74 | 1999 |

**Ограничение предметной области**

* Номер телефона клиента указывается в 10-ом формате;
* Номера автомобиля являются уникальный и имеют вид: Пример(А888НА174);
* Клиенты должны иметь возрастное ограничение 18+;
* Уникальный номер парковочного места имеет числовой формат с ограничением не более 999.
* дата рождения имеет вид (DD/MM/YYYY)
* Год выпуска имеет тип number, к примеру 1965;
* У каждого уникального парковочного места есть своя цена.
* Клиент паркуются только на своем парковочном месте исходя из номера автомобиля;
* У одного клиента может быть несколько автомобилей;
* У одного автомобиля может быть несколько клиентов.

1. **Выполнить концептуальное (инфологическое) проектирование.**
2. **Выполнить даталогическое проектирование для реляционной базы данных.**
3. **Выбрать СУБД (обосновав выбор) и выполнить физическое проектирование.**
4. **Создать базу данных в выбранной СУБД с учетом ограничений предметной области.**
5. **Реализовать следующие отчеты (запросы):**
   1. Выявить какой клиент больше всего задолжал автостоянке и когда у него был выполнен последний платеж.
   2. Посчитать сколько автомобилей имеют более одного хозяина и вывести всех хозяев.
   3. Вывести автомобиль, который имеет самый маленький долг к автостоянке за указанный пользователем период.
   4. Вывести сумму долга по всем клиентам за указанный пользователем период, которая вычисляется из расчета того, что начисление происходит 1 раз в месяц, оплаты могут происходить хоть каждый день. Все суммы начислений в данном поле складываются, а все оплаты идут со знаком минус. Долг может быть отрицательным!
   5. Вывести все номера и владельцев автомобилей, относящиеся к указанной пользователем марки автомобиля.
6. **Выбрать язык программирования и разработать приложение для работы с БД (формы ввода/редактирования данных и отчеты).**
7. **Оформить пояснительную записку.**

**СОДЕРЖАНИЕ**

1. Анализ предметной области 5

2. Концептуальная модель БД 6

3. Логическая модель БД 10

4. Физическая модель БД X

5. Описание приложения для работы с БД X

6. Заключение X

7. Приложение 1. SQL-скрипты для создания БД X

8. Приложение 2. SQL-скрипты запросов для отчетов X

9. Приложение 3. Исходный код программы для работы с БД X

10. Список литературы X

**АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

Основные ограничения предметной области и первичные документы уже даны нам в техническом задании (стр. 2), что упрощает дальнейшую работу.

В результате анализа предметной области по описанию технического задания можно прийти к следующим выводам:

* База данных создаётся для администрирования круглосуточной отапливаемой автостоянки.
* БД должна содержать данные о клиентах компании, автомобилях клиентов, парковочных местах, финансовом состоянии компании, посредством расчета задолженностей клиентов и предоставлять возможность получать разнообразные отчёты.

**КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ БД**

Для построения концептуальной модели БД выделим базовые сущности этой предметной области и представим эти сущности в виде таблицы сущностей (таб. 1):

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование сущности | Описание сущности | Первичный ключ | Кол-во экземпляров на основе первичных документов |
| Клиент | Посетитель автостоянки, пользующийся услугами автостоянки. | Код клиента | 2 |
| Автомобиль | Собственность клиента, оставляемая на хранение в рамках оказания услуг автостоянки владельцу этого вида собственности. | Номер автомобиля | 2 |
| Парковочное место | Площадь пространства на территории автостоянки, арендуемая клиентом с целью хранения на ней автомобиля, который закрепляется за этой площадью на срок аренды в рамках оказания услуг клиенту автостоянкой. | Парковочный номер | 2 |

Таблица 1

После определения базовых сущностей следующим шагом следует определить атрибуты каждой сущности.

Атрибуты сущности «Клиент» (таб. 2):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование атрибута | Признак ключа | Тип значения | Формат | Диапазон значений | Возможность принимать неопр. значения |
| Код клиента | Первичный ключ | Атомарное | Целое число | ≥0 | Нет |
| Фамилия | - | Атомарное | Строка.  Длина строки – до 100 символов | Допустимые символы “А-я” | Нет |
| Имя | - | Атомарное | Строка.  Длина строки – до 100 символов | Допустимые символы “А-я” | Нет |
| Отчество | - | Атомарное | Строка.  Длина строки – до 100 символов | Допустимые символы “А-я” | Да |
| Дата рождения | - | Множественное | Дата в формате  DD/MM/YYYY | От 01/01/1900 до сегодняшней даты | Нет |
| Номер телефона | - | Атомарное | Строка.  Длина - 11 цифр. | От 0 до 9 | Да |

Таблица 2

Атрибуты сущности «Автомобиль» (таб. 3):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование атрибута | Признак ключа | Тип значения | Формат | Диапазон значений | Возможность принимать неопр. значения |
| Номер автомобиля | Первичный ключ | Атомарное | Строка.  9 символов. | Допустимые символы “А-Я” и “0-9” | Нет |
| Марка автомобиля | - | Атомарное | Строка  127 символов. | Допустимые символы “A-z” и “А-я” и “0-9” | Да |
| Год выпуска | - | Атомарное | Целое число.  4 цифры. | 1900 – текущий год | Нет |

Таблица 3

Атрибуты сущности «Парковочное место» (таб. 4):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование атрибута | Признак ключа | Тип значения | Формат | Диапазон значений | Возможность принимать неопр. значения |
| Номер парковочного места | Первичный ключ | Атомарное | Целое число.  До 3 цифр. | 0…999 | Нет |
| Цена места | - | Атомарное | Число с плавающей точкой. | >0 | Нет |

Таблица 4

Сформировав сущности и их атрибуты закономерным шагом будет определение связей между этими сущностями (таб. 5):

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование связи | Тип связи | Хар-ка мощности связи | Перечень атрибутов связи |
| Договор аренды | Тернарная | 3 | Код клиента;  Номер места;  Номер автомобиля;  Дата договора;  Дата начисления;  Дата оплаты;  Сумма оплаты;  Задолженность. |

Таблица 5

После определения связей для каждой связи распишем её атрибуты.

Атрибуты связи «Аренда» (таб. 6):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование атрибута | Признак ключа | | Тип значения | Формат | Диапазон значений | Возможность принимать неопр. значения |
| Код клиента | Составной ключ | Внешний ключ | Атомарное | Целое число | ≥0 | Нет |
| Номер автомобиля | Атомарное | Строка.  9 символов. | Допустимые символы “А-Я” и “0-9” | Нет |
| Дата договора | - | | Атомарное | Дата в формате  DD/MM/YYYY | От 01/01/2000 до сегодняшней даты | Нет |
| Дата начисления | - | | Множественное | Дата в формате  DD/MM/YYYY | От 01/01/2000 до сегодняшней даты | Да |
| Дата оплаты | - | | Множественное | Дата в формате  DD/MM/YYYY | От 01/01/2000 до сегодняшней даты | Да |
| Сумма оплаты | - | | Атомарное | Число с плавающей точкой. | >0 | Да |
| Задолженность | - | | Атомарное | Число с плавающей точкой. | > -∞  < +∞ | Нет |

Таблица 6

На основании анализа предметной области представляется возможность проектирования концептуальной модели БД. (рис.1)

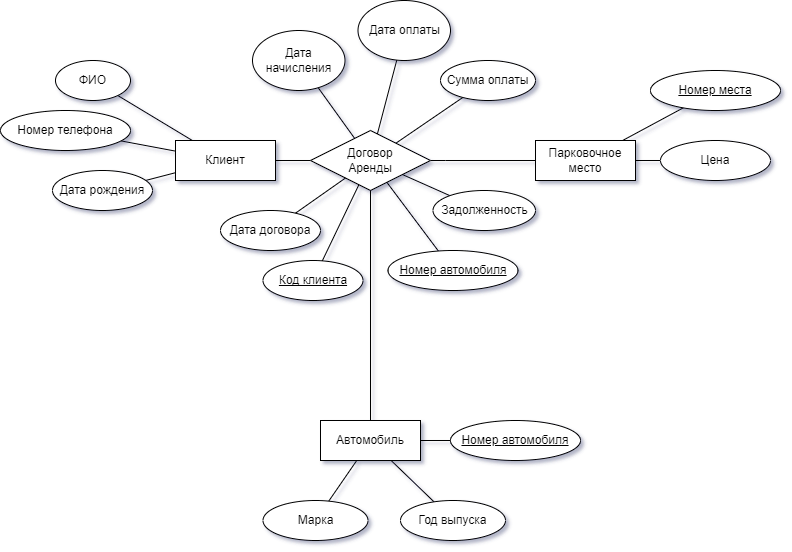


Рисунок 1 - ER диаграмма

Следующим шагом на пути разработки БД будет преобразование концептуальной модели в логическую модель базы данных.

**ЛОГИЧЕСКАЯ МОДЕЛЬ БД**

Перед созданием логической модели необходимо определиться с типом модели БД, которая будет использоваться на этапе физического проектирования.

Учитывая наше стремление к тому, чтобы получившаяся БД была достаточно гибкой для дальнейшего развития и расширения, а также, беря во внимание тенденции рынка, предпочтительней будет выбрать реляционную модель БД.

Соответственно следующие шаги по проектированию базы данных будут выполняться с учетом выбранной модели.

Сначала преобразуем сущности концептуальной модели в логическую.

Создание отношения «Клиент» (рис. 2):

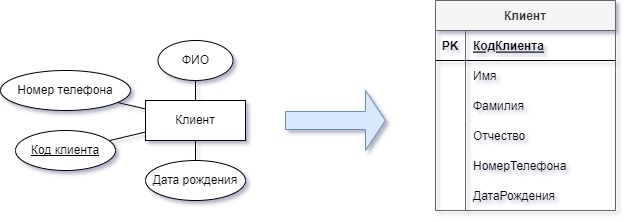


Рисунок 2 – «Клиент»

В рамках 1-й нормальной формы Атрибут «ФИО» преобразован в 3 атомарных атрибута – «Фамилия», «Имя» и «Отчество».

Создание отношения «Автомобиль» (рис. 3):

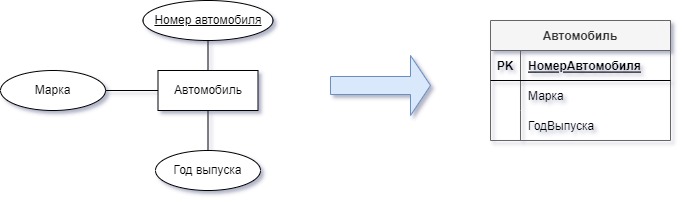


Рисунок 3 – «Автомобиль»

Создание отношения «Парковочное место» (рис. 4):

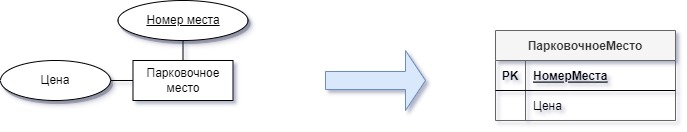


Рисунок 4 – «Парковочное место»

Учитывая то, что связь «Договор аренды» является тернарной и имеет атрибуты, перенести её в логическую модель можно будет только путем создания нового отношения «ДоговорАренды» (рис. 5):

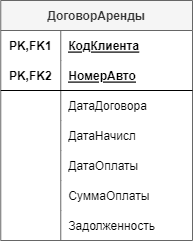


Рисунок 5 – «Договор Аренды»

При этом, учитывая то, что для одного парковочного места в один промежуток времени может быть один активный договор аренды, при этом парковочное место может быть свободным, то связь отношений «ДоговорАренды» и «ПарковочноеМесто» будет необязательной один-к-одному. Таким образом отношение «ПарковочноеМесто» принимает следующий вид (рис. 6):

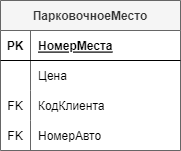


Рисунок 6 – обновленное «Парковочное Место»

В таком случае общая логическая модель БД принимает вид (рис. 7):

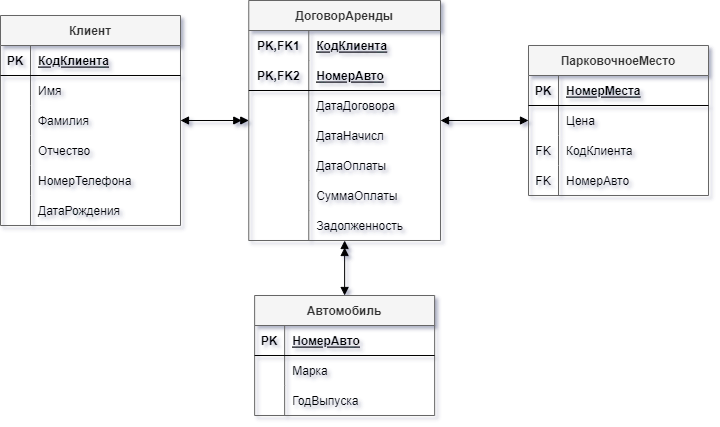


Рисунок 7

После прохождения проверок на 2НФ и 3НФ обнаружилось, что все атрибуты отношений находятся в полной ФЗ, из-за чего полученная модель находится в 3НФ и в дополнительных операциях нормализации не нуждается, в связи с чем имеет смысл приступить к физическому проектированию получившейся БД.

**ФИЗИЧЕСКАЯ МОДЕЛЬ БД**

На этапе физического проектирования БД стоит определиться с конкретной СУБД, на которой планируется разработка этой БД, т.к. это влечёт за собой различия в особенностях оптимизации и денормализации для различных СУБД.

В данном случае выбор пал на PostgreSQL по причине наличия мощных инструментов, обширной документации, бесплатной лицензии и популярности в профессиональной среде.

На этапе логического проектирования мы определи конкретные отношения-таблицы, которые будут присутствовать в нашей базе данных. Теперь же стоит рассмотреть размещение данных этих таблиц в рамках выбранной СУБД.

Размещение данных в отношении **Client** (логическое отношение “Клиент”) (таб. 7):

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование атрибута | Признак ключа | Тип данных | Возможность принимать NULL |
| ID | PRIMARY KEY | SERIAL | NOT NULL |
| FirstName | - | VARCHAR(100) | NOT NULL |
| LastName | - | VARCHAR(100) | NOT NULL |
| Patronymic | - | VARCHAR(100) | - |
| BirthDate | - | DATE | NOT NULL |
| PhoneNumber | - | VARCHAR(11) | - |

Таблица 7

Размещение данных в отношении **Car** (логическое отношение “Автомобиль”) (таб. 8):

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование атрибута | Признак ключа | Тип данных | Возможность принимать NULL |
| Number | PRIMARY KEY | VARCHAR(9) | NOT NULL |
| Brand | - | VARCHAR(100) | - |
| ReleaseYear | - | SMALLINT | NOT NULL |

Таблица 8

Размещение данных в отношении **ParkingSeat** (логическое отношение “ПарковочноеМесто”) (таб. 9):

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование атрибута | Признак ключа | Тип данных | Возможность принимать NULL |
| ID | PRIMARY KEY | SMALLINT | NOT NULL |
| Price | - | MONEY | NOT NULL |
| ClientID | FOREIGN KEY | INTEGER | - |
| CarNumber | FOREIGN KEY | VARCHAR(9) | - |

Таблица 9

Размещение данных в отношении **Contract** (логическое отношение “ДоговорАренды”) (таб. 10):

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование атрибута | Признак ключа | Тип данных | Возможность принимать NULL |
| ClientID | FOREIGN KEY / PRIMARY KEY | INTEGER | NOT NULL |
| CarNumber | FOREIGN KEY / PRIMARY KEY | VARCHAR(9) | NOT NULL |
| SeatID | FOREIGN KEY | SMALLINT | NOT NULL |
| ContractDate | - | DATE | NOT NULL |
| AccrualDate | - | DATE | - |
| Debt | - | MONEY | NOT NULL |
| PaymentDate | - | DATE | - |
| PaymentAmount | - | MONEY | - |

Таблица 10

После размещения данных стоит рассмотреть возможность денормализации либо оптимизации в рамках СУБД PostgreSQL.

В рамках оптимизации предпочтительно для отношения «ДоговорАренды» было принято решение заменить составной первичный / внешний ключ на искусственный ключ, при этом атрибуты составного ключа сделать обязательными и создать для них индексы.

Таким образом размещение данных в отношении **Contract** обретает следующий вид (таб. 11):

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование атрибута | Признак ключа | Тип данных | Возможность принимать NULL |
| ID | PRIMARY KEY | INTEGER | NOT NULL |
| ClientID | FOREIGN KEY | INTEGER | NOT NULL |
| CarNumber | FOREIGN KEY | VARCHAR(9) | NOT NULL |
| SeatID | FOREIGN KEY | SMALLINT | NOT NULL |
| ContractDate | - | DATE | NOT NULL |
| AccrualDate | - | DATE | - |
| Debt | - | MONEY | NOT NULL |
| PaymentDate | - | DATE | - |
| PaymentAmount | - | MONEY | - |

Таблица 11

В свою очередь размещение данных в отношении **ParkingSeat** обретает следующий вид (таб. 12):

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование атрибута | Признак ключа | Тип данных | Возможность принимать NULL |
| ID | PRIMARY KEY | SMALLINT | NOT NULL |
| Price | - | MONEY | NOT NULL |
| СontractID | FOREIGN KEY | INTEGER | - |

Таблица 12

При этом стоит добавить B-Tree индексы для следующих атрибутов:

* Contract.ClientID
* Contract.CarNumber
* ParkingSeat.ContractID

Следующим шагом остается реализация физической модели БД в СУБД PostgreSQL и разработка интерфейса для взаимодействия с БД.

**ОПИСАНИЕ ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ С БД**

Приложение для работы с получившейся БД представляет собой Web приложение написанное на ASP.NET Core 7.0 с применением технологии Razor Pages и фронтенд фреймворком Bootstrap.

Приложение имеет следующие формы:

1. Форма списка клиентов
   * Форма добавления нового клиента
   * Форма редактирования данных существующего клиента
2. Форма списков договоров, оплаты и начисления
   * Форма заключение нового договора
3. Форма списка автомобилей
   * Форма добавления нового автомобиля
4. Форма списка парковочных мест
   * Форма заключение нового договора (для выбранного свободного парковочного места)

Далее подробнее по каждой форме.

Форма списка клиентов (рис. 8):

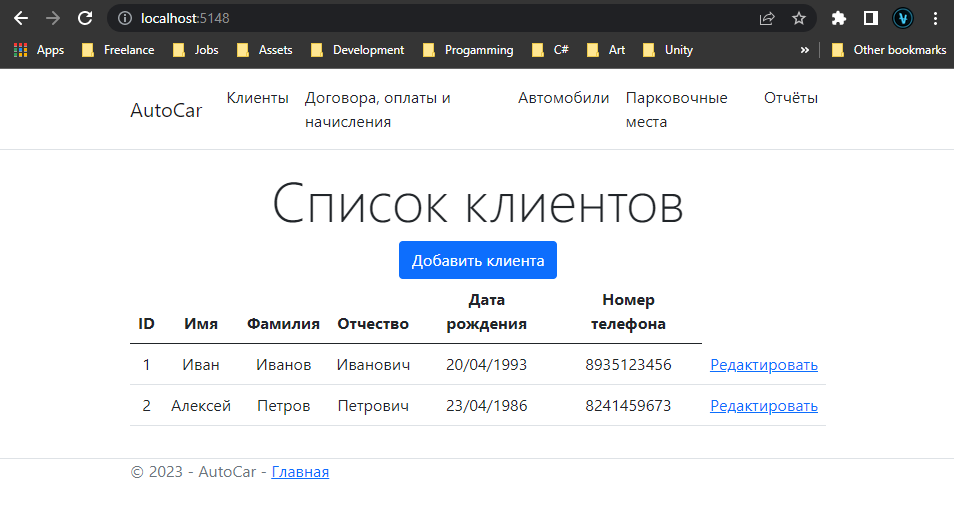


Рисунок 8

Позволяет посмотреть список доступных клиентов, при этом содержит ссылки на форму добавление нового клиента. Так же имеется ссылки на формы редактирования существующих клиентов.

Форма добавления нового клиента (рис. 9):

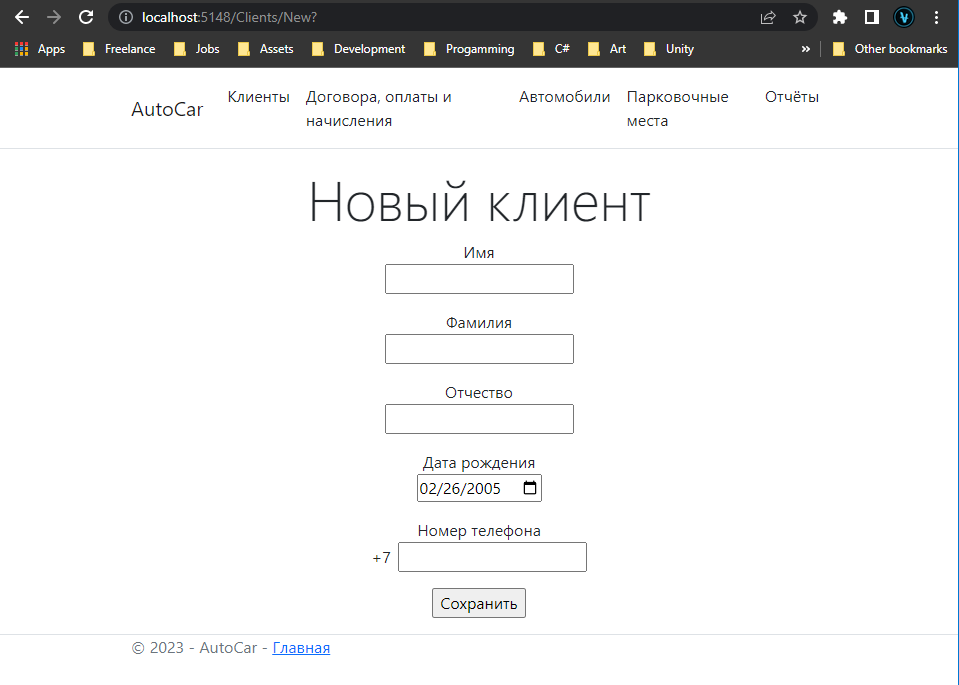


Рисунок 9

Форма позволяет добавить нового клиента, имеет валидацию данных на уровне обработки запроса.

Форма редактирования существующего клиента (рис. 10):

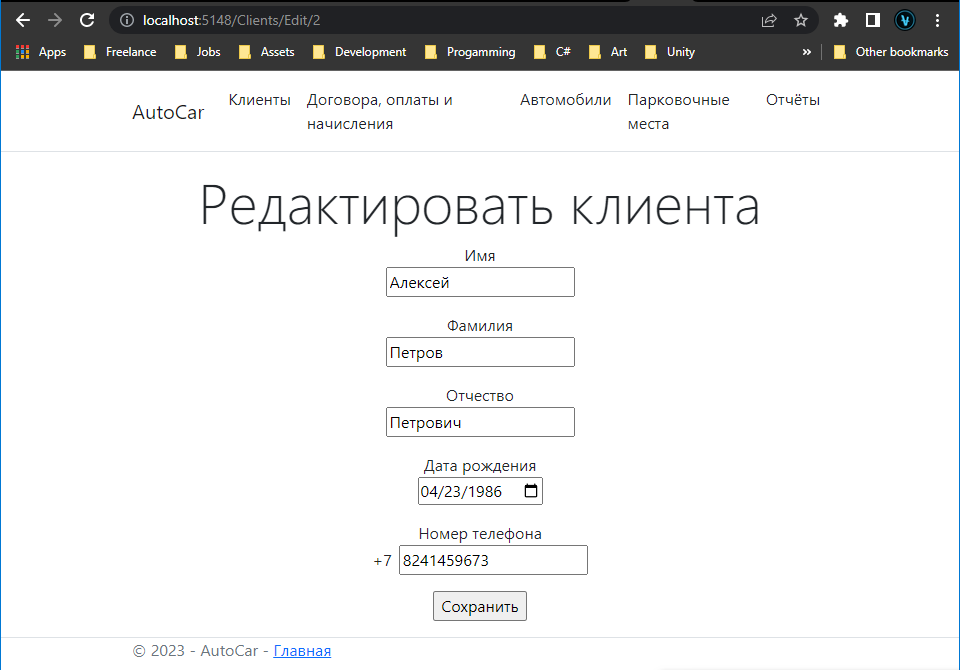


Рисунок 10

Форма позволяет редактировать существующего клиента, имеет валидацию данных на уровне обработки запроса.

Форма списка автомобилей (рис. 11):

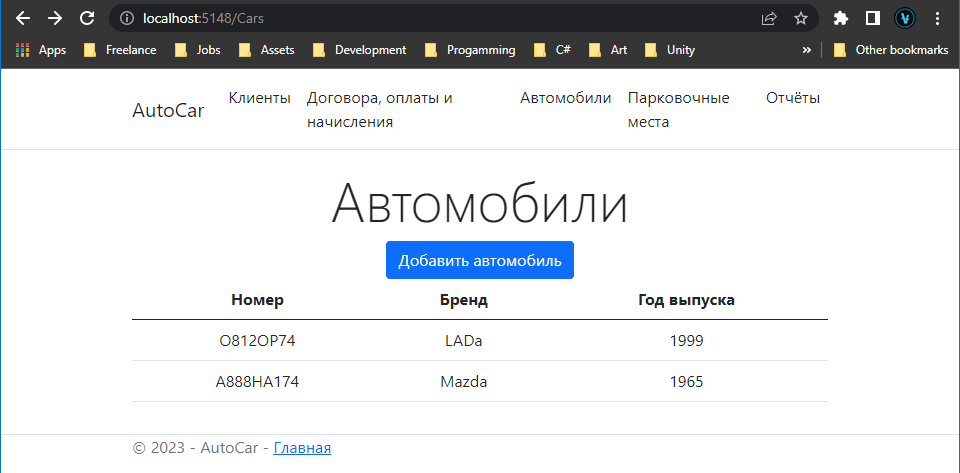


Рисунок 11

Форма позволяет просмотреть список автомобилей. Также имеется ссылка на форму добавления нового автомобиля.

Форма добавления автомобиля (рис. 12):

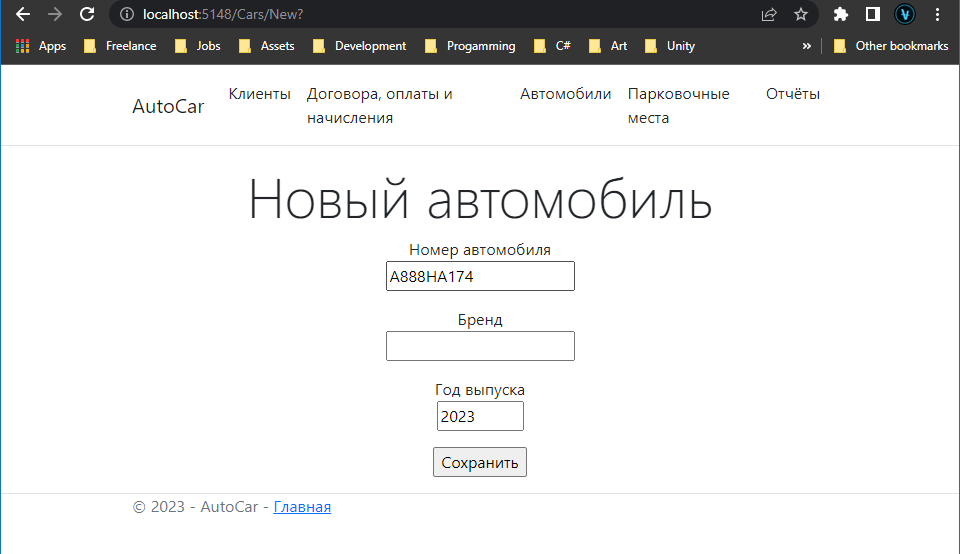


Рисунок 12

Форма позволяет добавить новый автомобиль, имеет валидацию данных на уровне обработки запроса.

**ЗАКЛЮЧЕНИЕ**

**СПИСОК ЛИТЕРАТУРЫ**

1. И.П. Карпова. Базы данных: Учебное пособие. СПб: Питер, 2013г. – 240с.
2. М.П. Малыхина. Базы данных: основы, проектирование, использование. 2-е изд. СПб: БХВ-Петербург, 2007.
3. В.В. Кирилов, Г.Ю. Громов. Введение в реляционные базы данных. – СПб.:БХВ-Петербург, 2009г. – 464с.
4. Г. Хансен, Дж. Хансен. Базы данных: разработка и управление. М: Бином, 1999г. – 704с.

**ПРИЛОЖЕНИЕ 1**