**Введение**

В условиях стремительного развития городской инфраструктуры и увеличения мобильности населения, аренда автомобилей становится неотъемлемой частью современной жизни. В последние годы наблюдается значительный рост спроса на услуги аренды машин, что обусловлено множеством факторов, таких как урбанизация, изменение стиля жизни, а также стремление к экономии и удобству. Данный курсовой проект посвящен исследованию важности и актуальности рынка аренды автомобилей в наши дни.

Аренда автомобилей предоставляет множество преимуществ как для частных лиц, так и для бизнеса. Для частных пользователей это возможность временного использования транспортного средства без необходимости его покупки и обслуживания, что особенно актуально для жителей крупных городов, где владение автомобилем может быть экономически невыгодным. Для бизнеса аренда автомобилей позволяет гибко управлять автопарком, снижать затраты на содержание и оперативно реагировать на изменения в спросе.

Важность аренды автомобилей также обусловлена экологическими и социальными аспектами. Развитие каршеринга и других моделей краткосрочной аренды способствует снижению количества личных автомобилей на дорогах, что, в свою очередь, уменьшает нагрузку на транспортную инфраструктуру и снижает уровень загрязнения окружающей среды. Кроме того, аренда автомобилей способствует развитию туризма и улучшению качества жизни, предоставляя людям возможность свободного передвижения и исследования новых мест.

Целью курсового проекта является создание автоматизированной информационной системы (АИС) для упрощения использования арендованных автомобилей. Система призвана упростить регистрацию пользователей, автоматизировать и обезопасить аренду автомобилей.

В процессе проектирования необходимо решить следующие задачи:

* **﻿﻿**провести анализ существующих систем;
* ﻿﻿сформулировать требования к программе;
* ﻿﻿построить структуру базы данных:
* ﻿﻿выбрать среду разработки;
* ﻿﻿реализовать программный продукт:
* ﻿﻿реализовать меры зашиты информации;
* ﻿﻿разработать техническую документацию.

**1. Проектирование**

**1.1 Обоснование среды разработки**

Для разработки программного обеспечения (ПП) можно использовать различные инструментальные средства. Рассмотрим несколько популярных интегрированных сред разработки (IDE):

* Visual Studio Community Edition
  + линейка систем разработки программного обеспечения от компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft). В своем составе имеют [интегрированную среду разработки (IDE)](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) и ряд других инструментов.
* Rider:
* Rider предлагает мощные инструменты для рефакторинга, анализа кода, отладки и тестирования. Она также поддерживает работу с различными системами контроля версий и интеграцию с популярными инструментами разработки. Rider работает на Windows, macOS и Linux.
* Visual Studio Professional :
  + Visual Studio Professional 2022 предоставляет эффективные функции, позволяющие быстро понимать код. CodeLens помогает сосредоточиться на работе: это решение отображает ссылки на код и изменения в коде, показывает, кто вносил в метод последние изменения, и определяет, успешно ли выполняются тесты. Все эти возможности предоставляются непосредственно при работе с кодом

Исходя из всех характеристик, которые мне удалось найти, выбор пал на Visual Studio Community Edition. Эта IDE, которая обеспечивает все возможности для создания, тестирования и отладки приложений.  
 Visual Studio поддерживает огромное количество языков программирования, в том числе:

* C#:
* Разработан компанией Microsoft и используется для создания приложений под платформу .NET.
* Объектно-ориентированный язык программирования с синтаксисом, похожим на Java.
* Имеет мощные инструменты для разработки, такие как отладка, рефакторинг, автодополнение кода и т.д.
* C++:
  + Разработан компанией Microsoft и используется для создания приложений под платформу .NET.
  + Объектно-ориентированный язык программирования с синтаксисом, похожим на Java.
  + Имеет мощные инструменты для разработки, такие как отладка, рефакторинг, автодополнение кода и т.д.
* JavaScript:
* Разработан компанией Microsoft и используется для создания приложений под платформу .NET.
* Объектно-ориентированный язык программирования с синтаксисом, похожим на Java.
* Имеет мощные инструменты для разработки, такие как отладка, рефакторинг, автодополнение кода и т.д.

Проведя анализ большинства языков программирования. Был сделан выбор в пользу языка программирования C# по нескольким причинам. C# обладает широким функционалом и мощными инструментами, которые позволяют разрабатывать сложные и производительные приложения. Этот язык имеет интегрированную среду разработки Visual Studio, которая облегчает процесс написания и отладки кода. C# поддерживает объектно-ориентированное программирование, что делает код более структурированным и понятным. Это позволяет легко масштабировать и расширять приложение в дальнейшем. Кроме того, C# имеет богатую стандартную библиотеку классов, которая содержит множество готовых компонентов и функций, упрощающих разработку приложений. Это позволяет сократить время разработки и повысить производительность приложения. C# является платформенно-независимым языком благодаря платформе .NET, что позволяет запускать приложения на различных операционных системах. Это делает C# универсальным и эффективным выбором для разработки приложений в разных сферах и на различных устройствах.

**1.2 Выбор способа организации входных и выходных данных**

При выборе способа организации входных и выходных данных для системы важно учитывать множество факторов, включая объем данных, типы операций с данными, требования к доступности и безопасности, а также характеристики используемых баз данных (БД). Рассмотрим несколько популярных систем управления базами данных, их характеристики и преимущества, которые могут помочь в принятии решения.

* MySQL
  + Тип: реляционная СУБД;
  + Лицензия: открытый исходный код (Community Edition) и коммерческие версии;
  + Поддерживаемые языки программирования: PHP, Python, Java, C#, Ruby и другие;
  + Объем данных: поддерживает большие объемы данных, масштабируемая;
  + Скорость: высокая производительность для чтения данных, хорошо оптимизирована для быстрого исполнения запросов;
  + Безопасность: поддерживает аутентификацию пользователей, шифрование и резервное копирование;
  + Функциональность: поддержка транзакций, индексов, триггеров, хранимых процедур.
* Microsoft SQL Server
  + Тип: реляционная СУБД;
  + Лицензия: коммерческое ПО;
  + Поддерживаемые языки программирования: .NET, C#, T-SQL;
  + Объем данных: поддерживает большие объемы данных, масштабируемая;
  + Скорость: высокая производительность для чтения и записи данных, оптимизирована для работы с другими продуктами Microsoft;
  + Безопасность: мощные средства аутентификации и авторизации, шифрование данных;
  + Функциональность: поддержка транзакций, индексов, триггеров, хранимых процедур, аналитических функций.
* PostgreSQL
* Тип: реляционная СУБД;
* Лицензия: открытый исходный код;
* Поддерживаемые языки программирования: PHP, Python, Java, C#, Ruby и другие;
* Объем данных: поддерживает большие объемы данных, масштабируемая;
* Скорость: хорошая производительность для чтения и записи данных;
* Безопасность: мощные средства аутентификации и авторизации, возможность шифрования данных;
* Функциональность: поддержка полного набора SQL-стандартов, транзакций, индексов, триггеров, хранимых процедур.

Сравнив вышеперечисленные СУБД, можем сделать вывод, что MySQL отлично подходит для данного проекта, так как это одна из самых популярных и широко используемых реляционных баз данных. Она обладает высокой производительностью, надежностью и устойчивостью к нагрузкам. MySQL имеет мощный набор функций и возможностей для работы с данными, а также поддерживает расширение функционала с помощью различных дополнительных инструментов и плагинов. Благодаря своей открытости и бесплатной лицензии, MySQL является отличным выбором для малых и средних проектов, а также для стартапов и компаний, которым важна экономия затрат на базы данных.

База данных состоит из 5 таблиц в соответствие с рисунком 1.2.1.

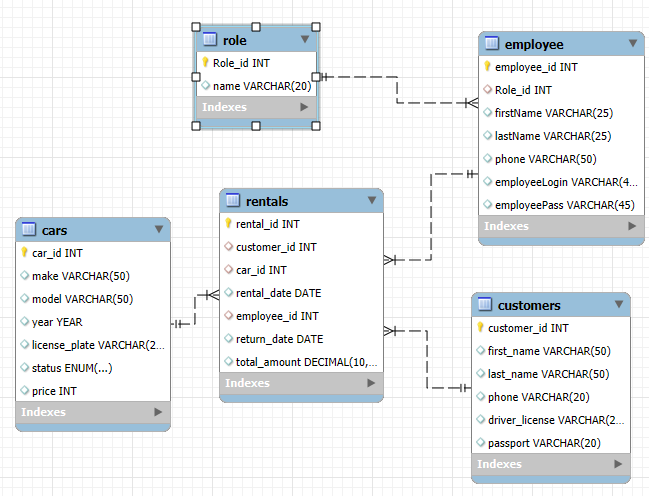


Рисунок 1.2.1. – Структура БД

Основной таблицей в БД является «Аренды»(Rentals). Таблица состоит из 7 полей в соответствии с таблицей 1.2.1.

Таблица 1.2.1. – Структура таблицы «Аренды»

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| ID | rental\_id | INT |  |
| Клиент | customer\_id | INT |  |
| Машина | car\_id | INT |  |
| Дата сдачи | Rental\_date | DATETIME |  |
| Менеджер | Employee\_id | INT |  |
| Дата возвращения | Return\_date | DATETIME |  |
| Сумма | Total\_amount | DECIMAL |  |

Таблица «Сотрудники»(employee) состоит из 7 полей в соответствии с таблицей 1.2.2.

Таблица 1.2.2. – структура таблицы «Сотрудники»

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| ID | employee\_id | INT |  |
| Роль | Role\_id | INT |  |
| Имя | firstName | VARCHAR | (25) |
| Фамилия | lastName | VARCHAR | (25) |
| Телефон | Phone | VARCHAR | (50) |
| Логин | employeeLogin | VARCHAR | (45) |
| Пароль | employeePass | VARCHAR | (100) |

Таблица «Клиенты»(customers) состоит из 6 полей в соответствии с таблицей 1.2.3.

Таблица 1.2.3. – структура таблицы «Клиенты»

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| ID | customer\_id | INT |  |
| Имя | First\_name | VARCHAR | (50) |
| Фамилия | Last\_name | VARCHAR | (50) |
| Телефон | Phone | VARCHAR | (20) |
| Вод.Удостоверение | Driver\_license | VARCHAR | (20) |
| Паспорт | passport | VARCHAR | (20) |

Таблица «Машины»(cars) состоит из 7 полей в соответствии с таблицей 1.2.4.

Таблица 1.2.4. – Структура таблицы «Машины»

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| ID | car\_id | INT |  |
| Марка | Make | VARCHAR | (50) |
| Модель | Model | VARCHAR | (50) |
| Год выпуска | Year | YEAR |  |
| Гос.Номер | License\_plate | VARCHAR | (20) |
| Статус | Status | ENUM | (‘Свободная’ , ‘Занята’) |
| Цена | price | INT |  |

Таблица «Роли»(role) состоит из 2 полей в соответствии с таблицей 1.2.5.

Таблица 1.2.5 – Структура таблицы «Роли»

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| ID | Role\_id | INT |  |
| Имя | name | VARCHAR | (20) |

**1.3 Проектирование модели данных**

При проектировании модели данных были разработаны диаграммы вариантов использования, деятельности и последовательности.

Диаграмма вариантов использования описывает функциональное назначение системы, т.е. что система будет делать в процессе своего функционирования. Является исходной концептуальной моделью системы в процессе ее проектирования и разработки. Диаграмма вариантов использования представлена в соответствии с рисунком 1.3.1.

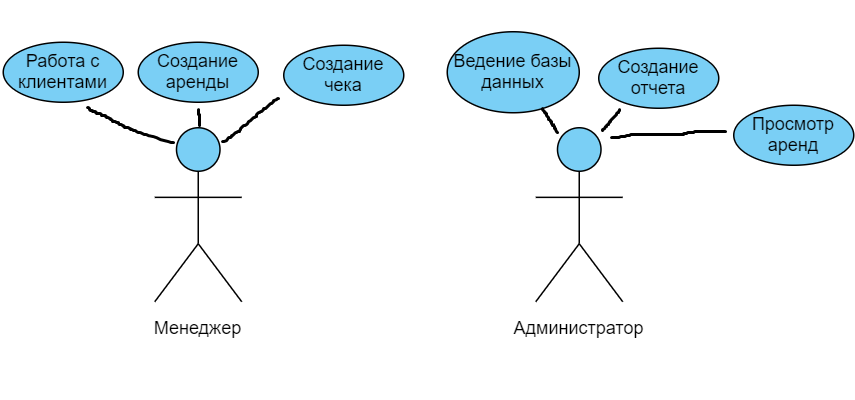


Рисунок 1.3.1 – Диаграмма вариантов использования

Сценарий использования - вариант использования, пользовательский  
сценарий в разработке программного обеспечения и системном проектировании. Сценарий диаграммы использования дополняет диаграмму, раскрывая содержание и логическую последовательность отдельных действий, которые выполняются системой и актерами. Сценарий представлен в соответствии с таблицей 1.3.1.

Таблица 1.3.1. - Сценарий диаграммы вариантов использования

|  |  |
| --- | --- |
| Вариант использования | Создание аренды |
| Актёры | Администратор, менеджер |
| Краткое описание | Менеджер регистрирует клиентов. Просматривает все данные о клиентах, машинах, арендах, а так же создает аренды. Также он может их редактировать и  удалять  Администратор управляет базой данных, отвечая за добавление, редактирование и удаление данных в системе. А также создает отчеты об арендах |
| Цель | Сдача машины в аренду |
| Тип | Базовый |
| Ссылки на другие варианты использования | Создание чека, работа с клиентами, ведение базы данных , создание отчета, просмотр аренд |

Таблица 1.3.2 — Последовательность действий

|  |  |
| --- | --- |
| Действия актеров | Отклик |
| 1. Администратор запускает программу | 1. Открытие формы авторизации |
| 1. Администратор вводит логин пароль   Исключение 1. Пароль или логин введены неверно | 1. Открытие главной формы администратора 2. Вывод сообщения об ошибке |
| 1. Переход на форму «сотрудники» по нажатию на кнопку «сотрудники» | 1. Открытие формы со списком сотрудников |
| 1. Добавление нового сотрудника по нажатию на кнопку «Добавить» | 1. Открытие формы добавление сотрудника |
| 1. Заполнение всех полей на форме «Добавление сотрудника» и нажатие на кнопку «Добавить»   Исключение 1. Заполнены не все поля  Исключение 2. Пользователь с таким логином уже существует | 1. Добавление нового сотрудника в базу данных 2. Вывод сообщения с просьбой заполнить все поля 3. Вывод сообщения об уже существующем пользователе |
| 1. Переход в главное меню по нажатию на кнопку «Назад» | 1. Открытие главной формы администратора |
| 1. Переход на форму «машины» по нажатию на кнопку «машины» | 1. Открытие формы со списком машины |
| 1. Добавление новой машины по нажатию на кнопку «Добавить» | 1. Открытие формы добавление машины |

Продолжение таблицы 1.3.2

|  |  |
| --- | --- |
| 1 | 2 |
| 1. Заполнение всех полей на форме «Добавление машины» и нажатие на кнопку «Добавить»   Исключение 1. Заполнены не все поля  Исключение 2. Дубликат машины | 1. Добавление новой машины в базу данных 2. Вывод сообщения с просьбой заполнить все поля 3. Вывод сообщения об уже существующем машине |
| 1. Переход в главное меню по нажатию на кнопку «Назад» | 1. Открытие главной формы администратора |
| 1. Переход на форму «Аренды» по нажатию на кнопку «Аренды» | 1. Открытие формы со списком аренд |
| 1. Переход в главное меню по нажатию на кнопку «Назад» | 1. Открытие главной формы администратора |
| 1. Переход на форму «Клиенты» по нажатию на кнопку «Клиенты» | 1. Открытие формы со Клиентами |

Продолжение таблицы 1.3.2

|  |  |
| --- | --- |
| 1 | 2 |
| 1. Переход в главное меню по нажатию на кнопку «Назад» | 1. Открытие главной формы администратора |
| 1. Выход из приложения осуществляется по нажатию на кнопку «Выйти» | 1. Закрытие приложения |
| 1. Смена пользователя из приложения осуществляется по нажатию на кнопку «Сменить пользователя» | 1. Открытие формы авторизации |
| 1. Менеджер запускает программу | 1. Открытие формы авторизации |
| 1. Менеджер вводит логин пароль   Исключение 1. Пароль или логин введены неверно | 1. Открытие главной формы менеджера 2. Вывод сообщения об ошибке |

Продолжение таблицы 1.3.2

|  |  |
| --- | --- |
| 1 | 2 |
| 1. Переход на форму «Клиенты» по нажатию на кнопку «Клиенты» | 1. Открытие формы со списком клиентов |
| 1. Добавление нового клиента по нажатию на кнопку «Добавить» | 1. Открытие формы добавление клиента |
| 1. Заполнение всех полей на форме «Добавление клиента» и нажатие на кнопку «Добавить»   Исключение 1. Заполнены не все поля  Исключение 2. Дубликат клиента | 1. Добавление нового клиента в базу данных 2. Вывод сообщения с просьбой заполнить все поля 3. Вывод сообщения об уже существующем клиенте |
| 1. Переход в главное меню по нажатию на кнопку «Назад» | 1. Открытие главной формы менеджера |
| 1. Переход на форму «Аренды» по нажатию на кнопку «Аренды» | 1. Открытие формы со списком аренд |
| 1. Добавление новой аренды по нажатию на кнопку «Добавить» | 1. Открытие формы добавление аренды |
| 1. Менеджер выбирает машину, клиента из выпадающего списка и сроки аренды и нажимает кнопку «Добавить   Исключение 1. Заполнены не все поля | 1. Добавление новой аренды в базу данных 2. Вывод сообщения с просьбой заполнить все поля |

Продолжение таблицы 1.3.2

|  |  |
| --- | --- |
| 1 | 2 |
| 1. Ввод в поисковую строку «Геннадий» на форме «Клиенты» | 1. Вывод всех клиентов, которых зовут «Геннадий» |
| 1. Выбор сортировки в выдающем списке «Сортировка» | 1. Данные будут отсортированы согласно выбранной сортировке |
| 1. Выбор фильтрации в выдающем списке «Фильтры» | 1. Данные будут отфильтрованы согласно выбранной фильтрации |
| 1. Переход в главное меню по нажатию на кнопку «Назад» | 1. Открытие главного меню менеджера |

Диаграмма деятельности — диаграмма, позволяющая подробно иллюстрировать отдельный вариант использования и его сценарий. Диаграммы деятельности помогают визуализировать шаги или этапы выполнения процесса, а также последовательности действий и условия, при которых эти действия выполняются.

Диаграмма деятельности представлена в соответствии с рисунком 1.3.2.

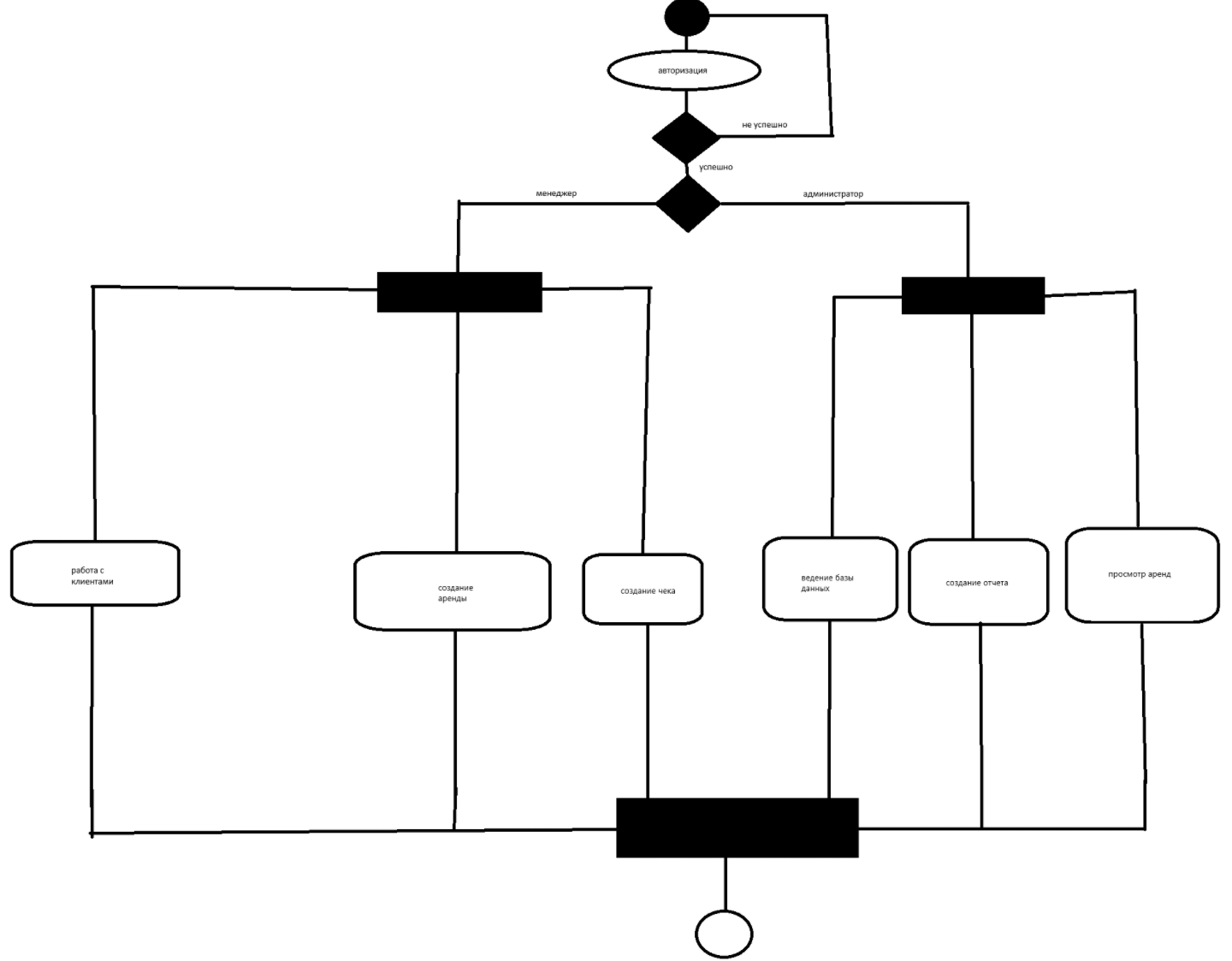


Рисунок 1.3.2 – Диаграмма деятельности

Диаграмма последовательности — графическая модель, которая для определенного сценария варианта использования показывает динамику взаимодействия объектов во времени. Эта диаграмма показывает, как объекты взаимодействуют друг с другом через обмен сообщениями для выполнения определенной функциональности.

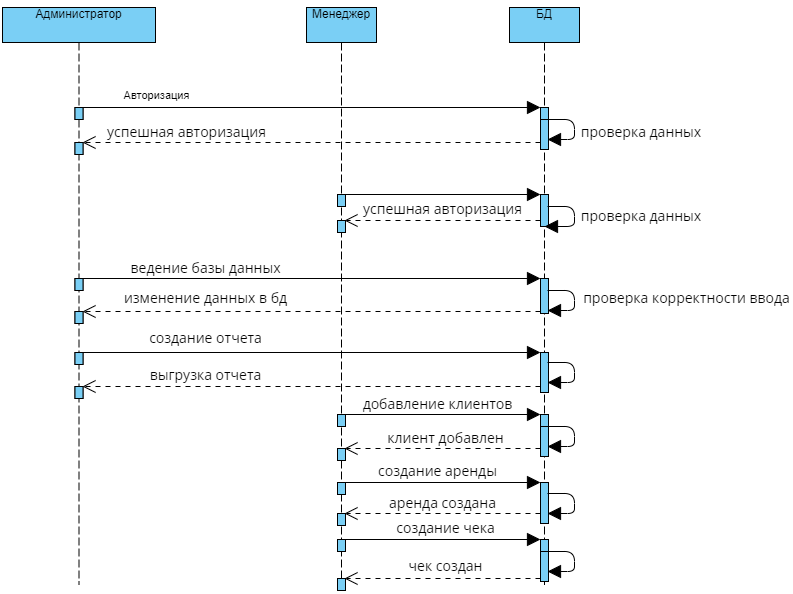


Рисунок 1.3.3 — Диаграмма последовательности

**2. Алгоритм представления задачи**

*2.1 Детальное проектирование пользовательского интерфейса*

Прототип — это первообраз, предварительная модель или образец какого-либо объекта, системы или концепции, который используется для тестирования, демонстрации или доработки.

Виды прототипов:

* горизонтальный прототип — прототип, моделирующий интерфейс пользователя приложения, не затрагивая логики, обработки и структур данных;
* вертикальный прототип — прототип, направленный не столько на проектирование интерфейса пользователя, сколько на реализацию вертикального «среза» системы, затрагивая все уровни ее реализации;
* одноразовый прототип — прототип, создающийся, когда нужно быстро получить макет разработанной программной системы;
* эволюционный прототип — прототип, создающийся как первое приближение системы, призванная впоследствии стать самой системой;
* бумажный прототип — наброски интерфейсов на бумаге.

При разработке автоматизированной информационной системы «Сдача машин в аренду» на базе платформы .NET на этапе проектирования был выбран эволюционный прототип. Он позволяет гибко адаптировать проект в процессе разработки, реагируя на обратную связь пользователей и изменения требований. Такой подход способствует выявлению ошибок на ранних стадиях и позволяет улучшать функциональность приложения по мере его создания, что в конечном итоге повышает качество и удовлетворенность конечного продукта.

Прототип программы представлен в соответствии с рисунками 2.1.1 — 2.1.15.

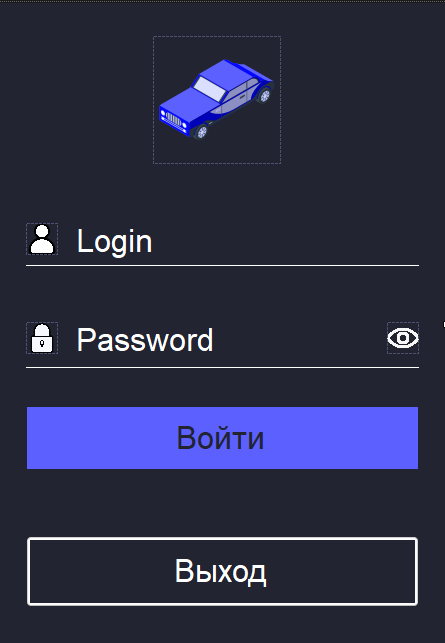


Рисунок 2.1.1. — Форма авторизации

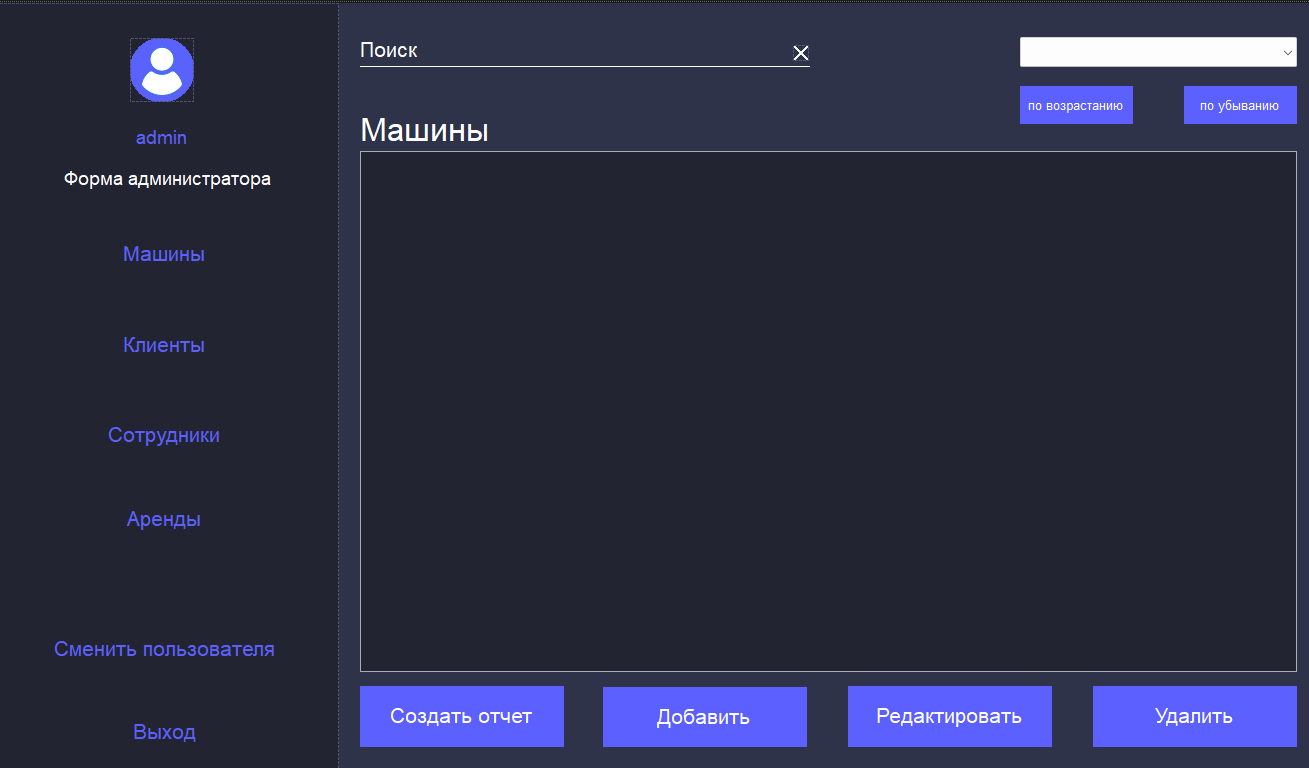


Рисунок 2.1.2. — Главное меню администратора

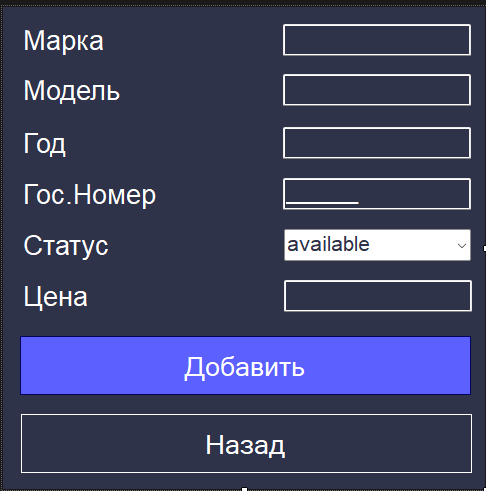


Рисунок 2.1.3.— Форма добавления машин

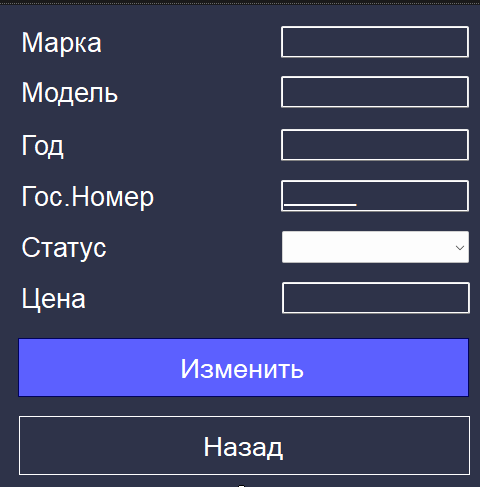


Рисунок 2.1.4.— Форма редактирования машин

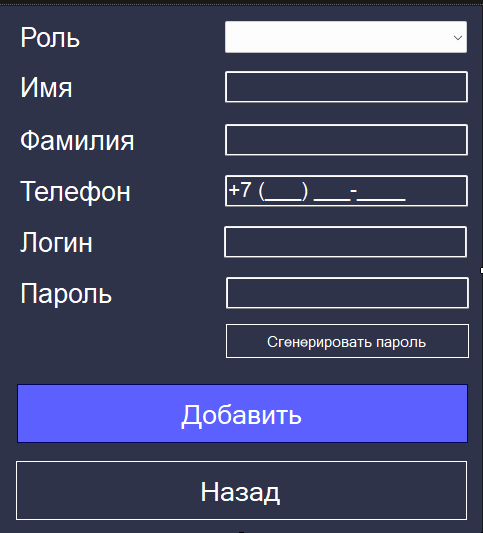


Рисунок 2.1.5— Форма добавления сотрудников

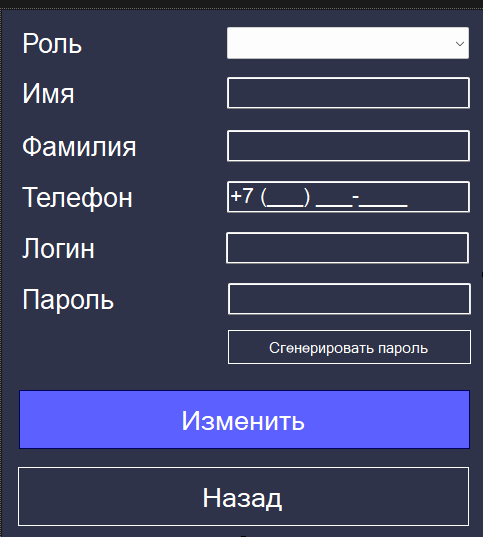


Рисунок 2.1.6.— Форма редактирования сотрудников

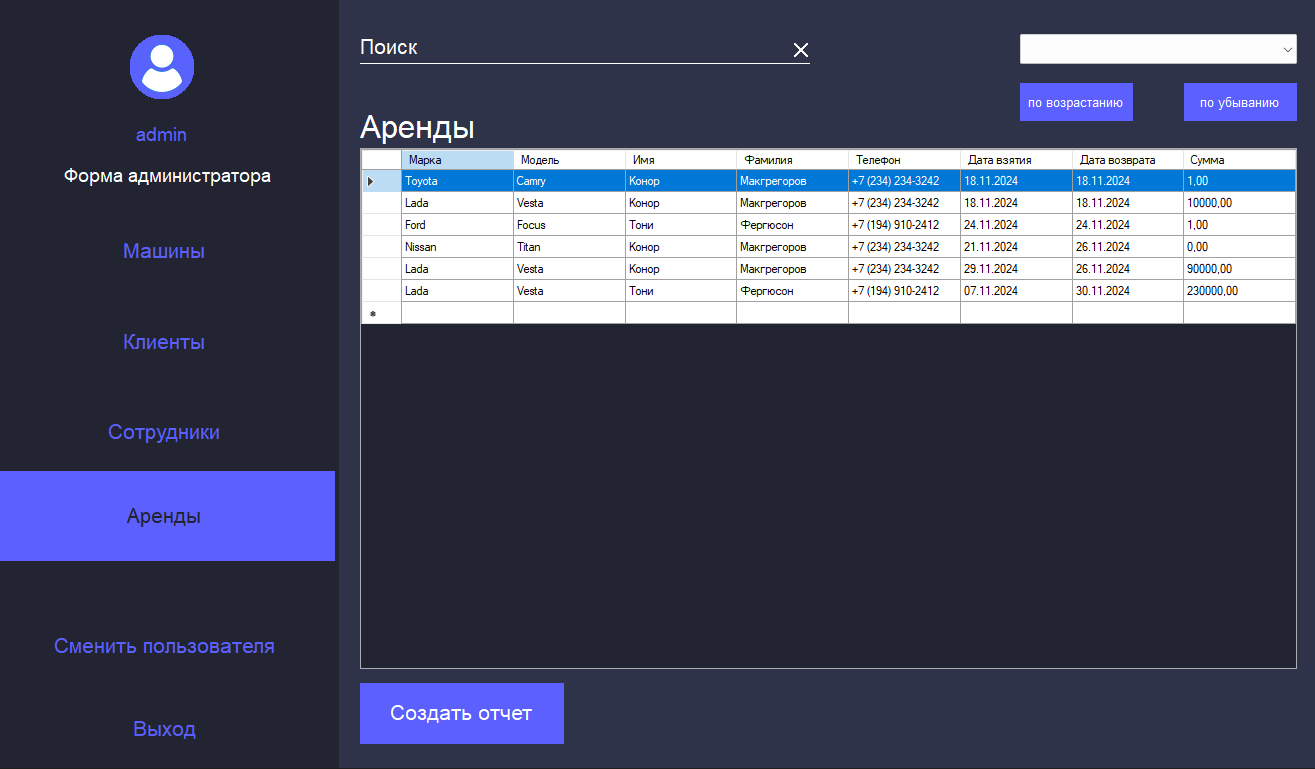


Рисунок 2.1.7. — Форма просмотра списка аренд

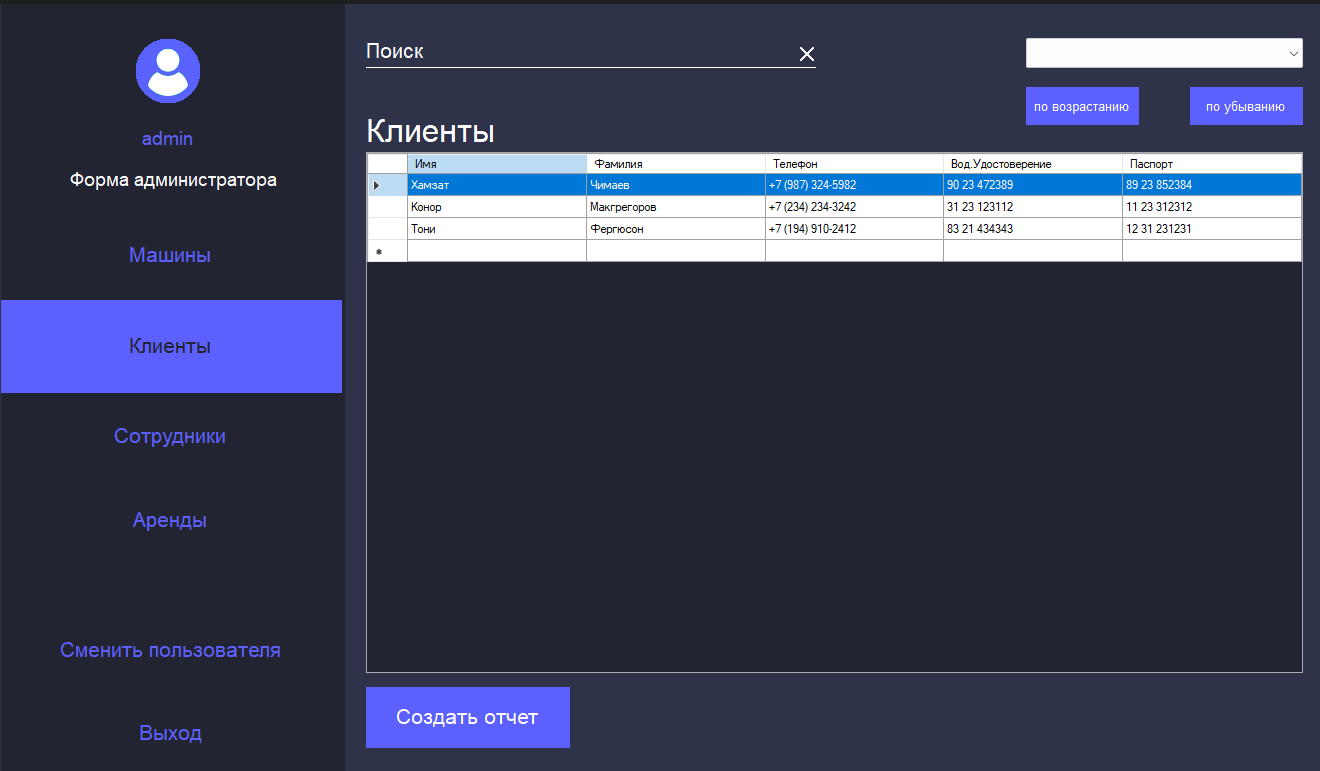


Рисунок 2.1.8.— Форма просмотра списка клиентов

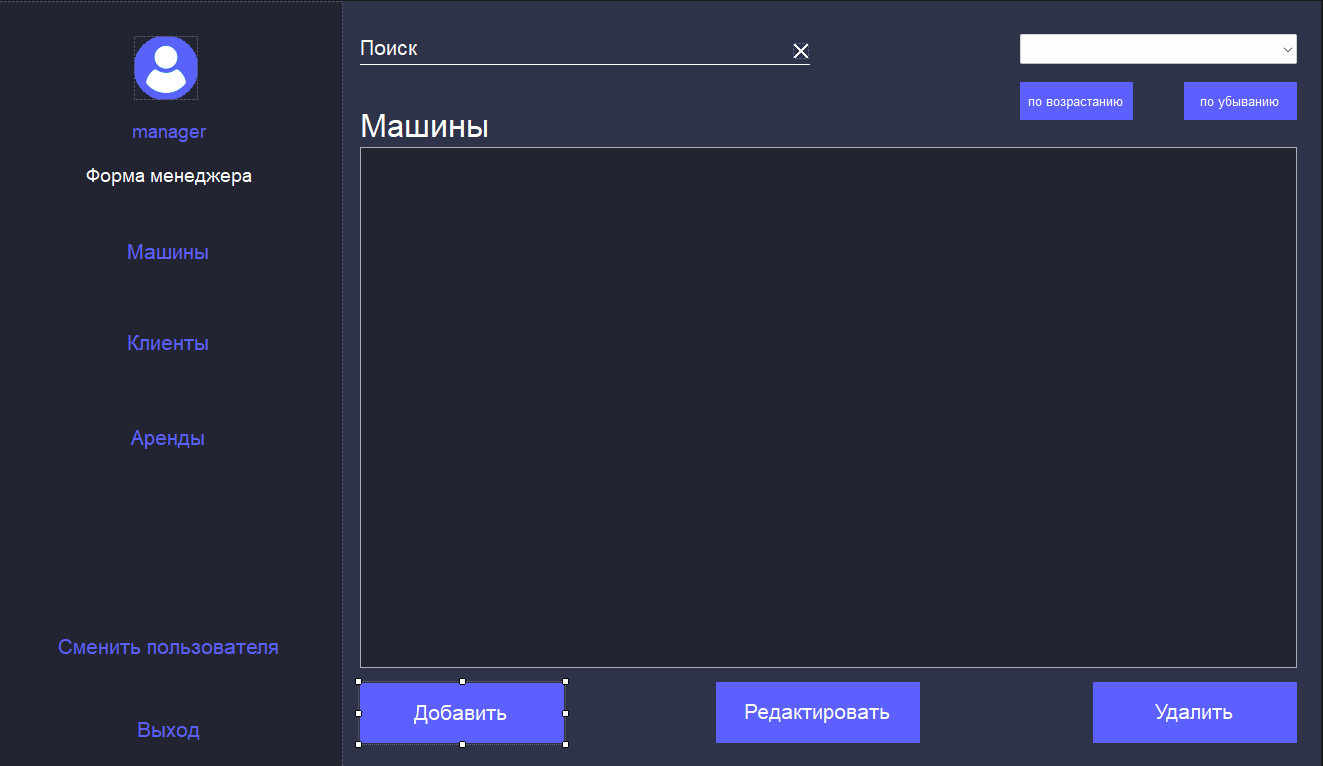


Рисунок 2.1.9. — Форма главного меню менеджера

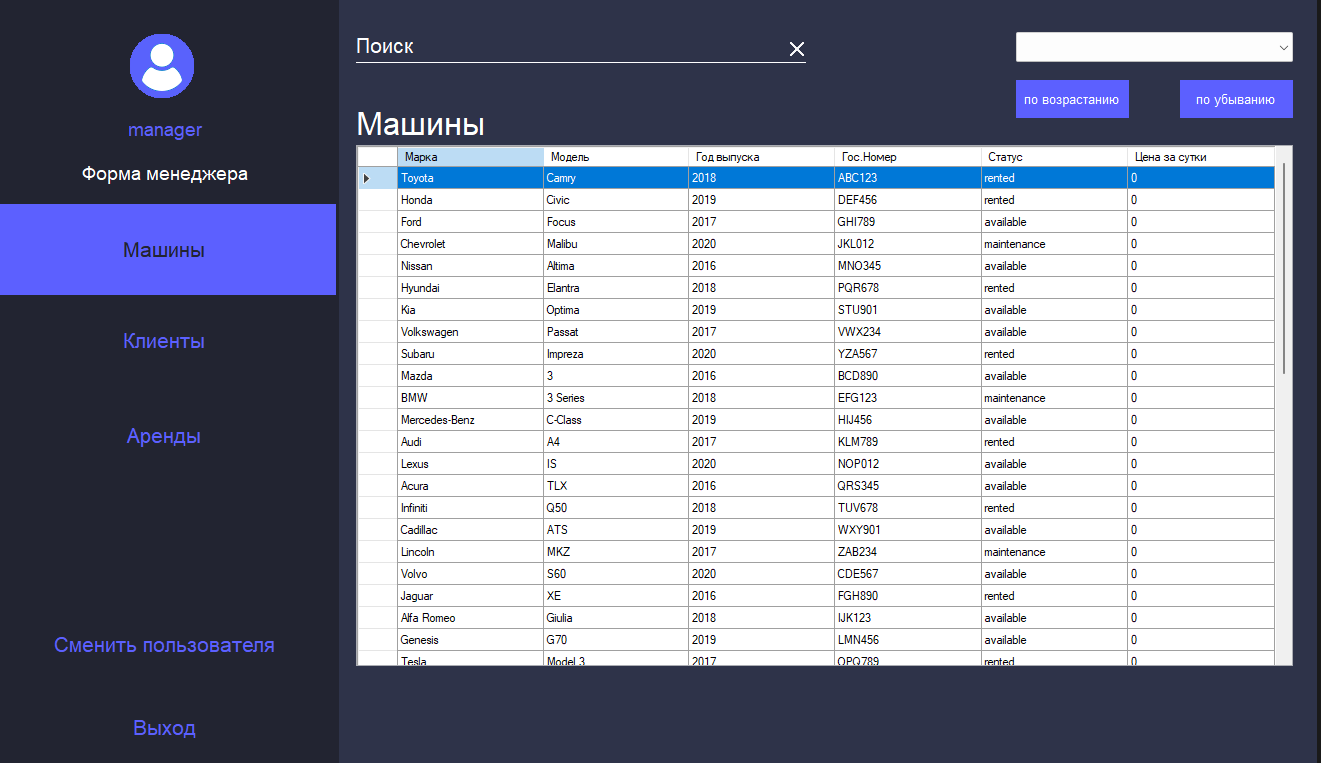


Рисунок 2.1.10.— Форма просмотра списка машин

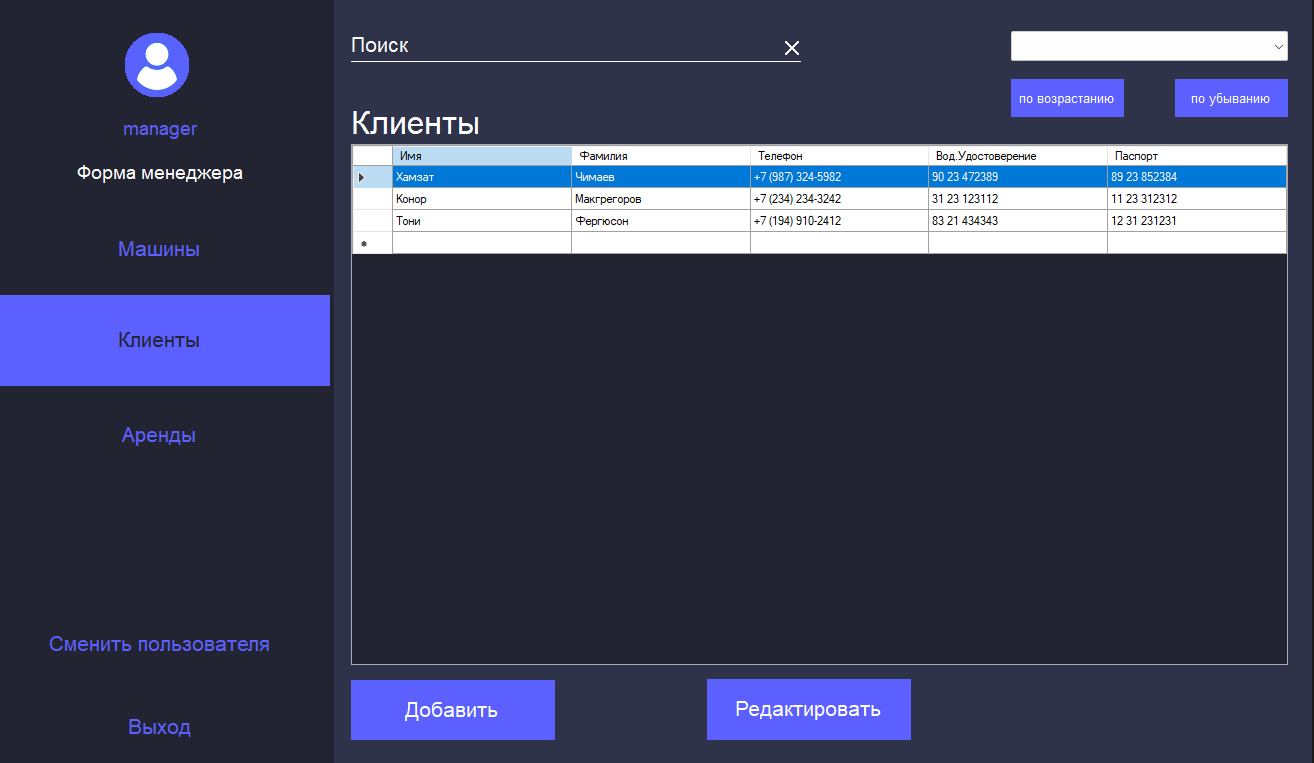


Рисунок 2.1.11. — Форма просмотра списка клиентов

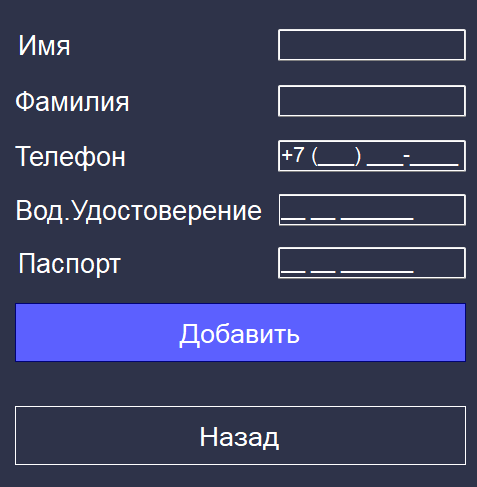


Рисунок 2.1.12.— Форма добавления клиента

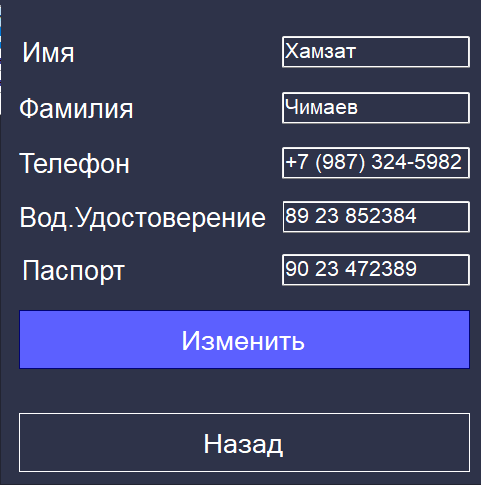


Рисунок 2.1.13.— Форма редактирования клиента

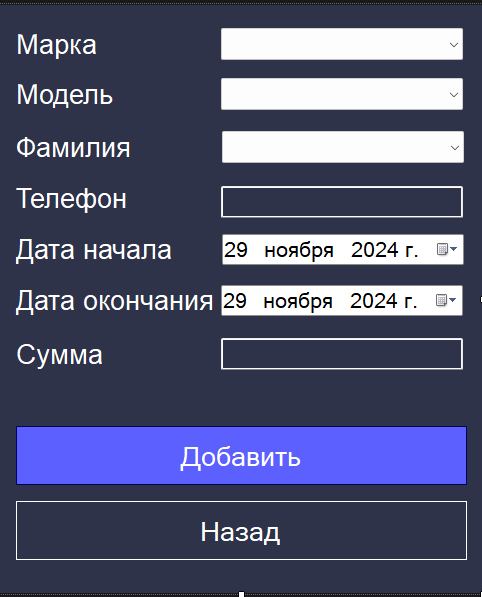


Рисунок 2.1.14.— Форма создания аренды

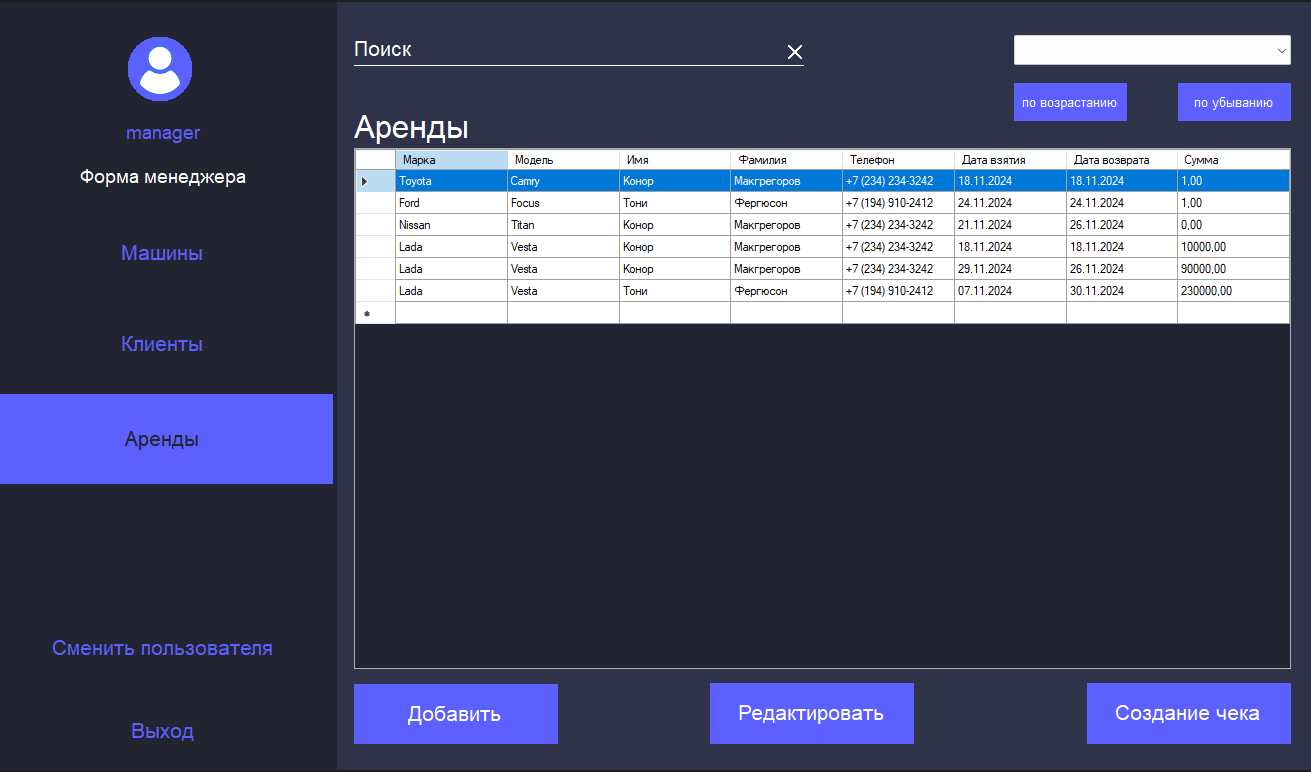


Рисунок 2.1.15— Форма просмотра списка заказов

*2.2 Описание основных программных модулей*

* Модуль авторизации

private void button1\_Click(object sender, EventArgs e){} — данный модуль выполняет вход на форму администратора или менеджера.

private void button2\_Click(object sender, EventArgs e){} — данный модуль выполняет выход из приложения через форму авторизации.

* Модуль главного меню администратора

private void button1\_Click(object sender, EventArgs e){} — данный модуль выполняет переключает меню на работу с таблицей «Машины».

private void button2\_Click(object sender, EventArgs e){} — данный модуль выполняет переключает меню на работу с таблицей «Клиенты».

private void button3\_Click(object sender, EventArgs e){} — данный модуль выполняет переключает меню на работу с таблицей «Сотрудники».

private void button4\_Click(object sender, EventArgs e){} — данный модуль выполняет переключает меню на работу с таблицей «Аренды».

private void button7\_Click(object sender, EventArgs e){} — данный модуль открывает форму добавления».

private void button8\_Click\_1(object sender, EventArgs e){} — данный модуль открывает форму редактирования».

private void button9\_Click(object sender, EventArgs e){} — данный модуль удаляет выбранную строку».

private void reportBtn\_Click(object sender, EventArgs e){} — данный модуль создает отчет».

* Модуль главного меню менеджера

private void button1\_Click(object sender, EventArgs e){} — данный модуль выполняет переключает меню на работу с таблицей «Машины».

private void button2\_Click(object sender, EventArgs e){} — данный модуль выполняет переключает меню на работу с таблицей «Клиенты».

private void button3\_Click(object sender, EventArgs e){} — данный модуль выполняет переключает меню на работу с таблицей «Сотрудники».

private void button4\_Click(object sender, EventArgs e){} — данный модуль выполняет переключает меню на работу с таблицей «Аренды».

private void button7\_Click(object sender, EventArgs e){} — данный модуль открывает форму добавления».

private void button8\_Click\_1(object sender, EventArgs e){} — данный модуль открывает форму редактирования».

private void button9\_Click(object sender, EventArgs e){} — данный модуль создает чек аренды».

*2.3 Применяемые меры защиты информации*

Защите данных — это обеспечение защиты информации от несанкционированного (случайного или умышленного) доступа к данным, их изменения или разрушения со стороны пользователя.

В процессе разработки информационной системы были реализованы следующие меры защиты информации:

* *парольная защита на разграничение прав доступа для входа в программу* представлена в соответствии с рисунком 2.3.1;

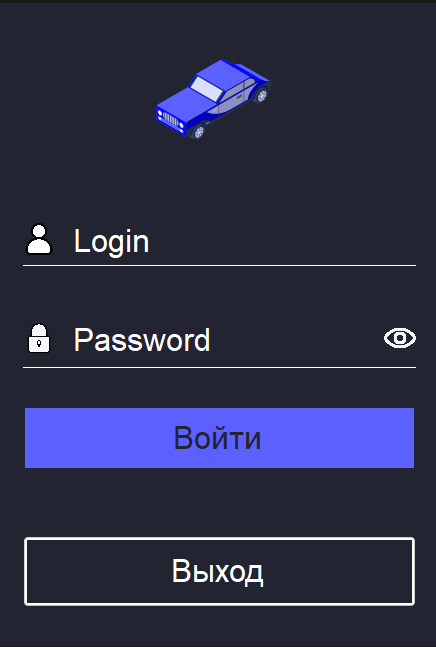


Рисунок 2.3.1 — Разграничение прав доступа

* *скрытие пароля на форме «Авторизация»* представлено в соответствии с рисунком 2.3.2;



Рисунок 2.3.2 — Скрытие пароля

* *блокировка удаления данных при их связи с другими таблицами* представлена в соответствии с рисунком 2.3.3;

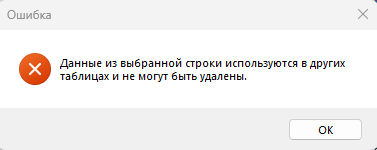


Рисунок 2.3.3 — Блокировка удаления машины

* *удаление данных осуществляется только после подтверждения* представлено в соответствии с рисунком 2.3.4;

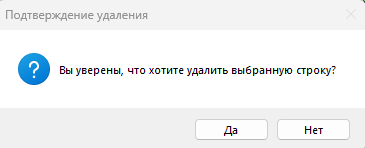


Рисунок 2.3.4 — Подтверждение удаления

* *выход из приложения осуществляется только через меню* представлено в соответствии с рисунком 2.3.5;

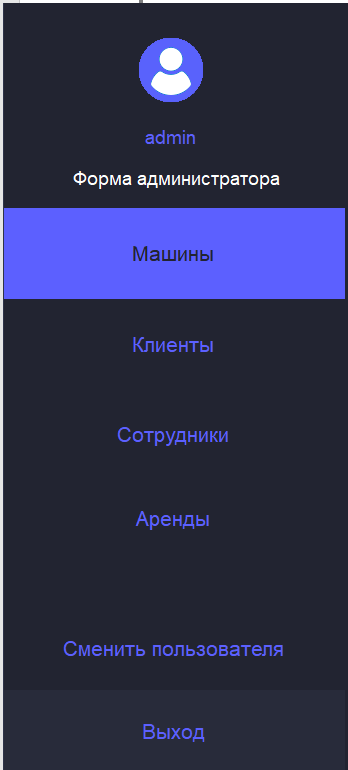


Рисунок 2.3.5 — Выход через главное меню

* *выход из приложения осуществляется после подтверждения* представлено в соответствии с рисунком 2.3.6;

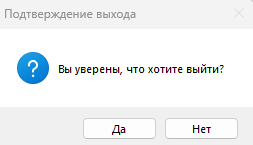


Рисунок 2.3.6 — Выход через главное меню

* *выбор в полях с датой производится через календарь с автоматической проверкой с системной датой* представлен в соответствии с рисунком 2.3.7;

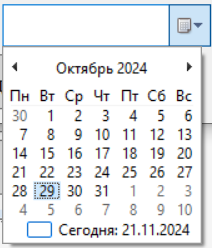


Рисунок 2.3.7 — Выбор даты через календарь

* при оформлении заказа на форме «Добавление аренды» автоматически подставляется текущая дата в поле «Дата».

**Заключение**

В ходе выполнения курсового проекта была создана автоматизированная информационная система сдачи машин в аренду на базе платформы .NET, отвечающая поставленным требованиям.

В процессе написания программного кода использовался язык программирования высокого уровня. Были улучшены навыки по работе с платформой .NET.

В программном приложении реализованы следующие функции:

* + - для доступа к системе необходимо реализовать разграничение прав для 2-х ролей: администратор и менеджер;
    - выводить логин сотрудниках на главной форме;
    - программное приложение должно иметь формы: ввода и вывода информации о автомобилях, заказах, пользователях и клиентах;
    - реализовать «живой» поиск при работе с информацией о автомобилях по наименованию, году выпуска, гос. номеру ;
    - реализовать «живой» поиск при работе с информацией о клиентах по ФИО, вод. удостоверению, номеру телефона, паспорту ;
    - реализовать «живой» поиск при работе с информацией о сотрудниках по фамилии , имени , логину , номеру телефона ;
    - реализовать фильтрацию на формах с информацией по статусу аренды;
    - обеспечить сортировку информации о аренде автомобилей по дате аренды;
    - в режиме «Администратор» пользователю предоставляются возможности:
      * добавление, редактирование, удаление информации о сотрудниках, автомобилях;
      * просмотр информации о клиентах, арендах;
      * формирование отчета по арендам, содержащего информацию о наименовании автомобиля, клиента, стоимости аренды за 1 сутки и итоговой стоимости;
    - печать отчетов в формате документа Word версии 2007;
    - в режиме «Менеджер» пользователю предоставляются возможности:
      * оформление новой аренды с выпадающем списком информации о клиенте, автомобиле и количестве дней с последующей печатью чека об оказании услуг;
      * просмотр информации о автомобилях, арендах;
      * просмотр, добавление и редактирование информации о клиентах;
    - автоматизация ввода исходных данных:
      * выпадающий список используется при выборе клиента при оформлении аренды;
      * выпадающий список используется при выборе марке машины при оформлении аренды;
      * выпадающий список используется при выборе модели машины при оформлении аренды;
      * при создании аренды используется системная дата;
      * при формировании отчета используются календари для выбора даты отчета;
      * возможность автоматической генерация логина и пароля при добавлении нового пользователя;
      * при вводе номера телефона, водительского удостоверения, Гос.номера автомобиля, паспорта используется маска ввода.

После внедрения программы могут быть доработаны следующие функции:

* возможность резервного копирования базы данных;
* поддержка масштабируемости;
* блокировка системы на 10 секунд после неудачной попытки авторизации с помощью captcha;
* формирование отчета в Excel;
* функция защиты персональных данных;
* блокировка и перевод на форму авторизации при отсутствии действий от пользователя в течение 15 минут;
* возможность восстановления базы данных;
* импорт и экспорт csv файлов;
* дополнительная защита авторизации с помощью captcha;
* возможность изменения настроек подключения к базе данных;
* пагинация на формах со списком товаров.

**Литература**

1 Бен Форта, Освой самостоятельно SQL за 10 минут. М., Вильямс, 2021

2 Мацяшек Л.А., Лионг Б.Л. Практическая программная инженерия на основе учебного примера - М.:БИНОМ, Лаборатория знаний – ЭБК BOOK.RU 2022.

3 Могрунов Е.П. Основы языка SQL, учебное пособие. – СПб.: БХВ-Петербург, 2022.

4 Новиков Б.А., Основы технологий баз данных, Postgres Professional. - М., 2021.

5 Прайс Марк Дж. C# 7 и .NET Core Кросс-платформенная разработка для профессионалов. 3-е изд. - СПб.: Питер, 2022.

6 Эндрю Троелсен Язык программирования C# 6.0 и платформа .NET 4.6, 7-е изд.: пер. с англ. - М.: ООО “И.Д. Вильямс”, 2021.