

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем
Направление подготовки 09.03.02 – Информационные системы и технологии

ДОПУСТИТЬ К ЗАЩИТЕ
зав. кафедрой

_____ А.В. Бушманов
« ____ » _____ 2024 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка ИС «Учет грузоперевозок для компании ООО «КОНТИ-
НЕНТ»»

Исполнитель

студент группы 0104-об

(подпись, дата)

М. А. Колосов

Руководитель

доцент

(подпись, дата)

И. М Акилова

Консультант

по безопасности

и экологичности

доцент, канд. техн. наук

(подпись, дата)

А. Б. Булгаков

Нормоконтроль

инженер кафедры

(подпись, дата)

В.Н. Адаменко

Благовещенск 2024

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем
Направление подготовки 09.03.02 – Информационные системы и технологии

УТВЕРЖДАЮ
зав. кафедрой

_____ А.В. Бушманов
«__» _____ 2023 г.

З А Д А Н И Е

К бакалаврской работе студента Колосов Максим Алексеевич

1. Тема выпускной квалификационной работы: Разработка ИС «Учет грузоперевозок для компании ООО «КОНТИНЕНТ»»

2. Срок сдачи студентом законченной работы: 19.06.2024г.

3. Исходные данные к бакалаврской работе: техническое задание, специальная литература, нормативные документы.

4. Содержание бакалаврской работы (перечень подлежащих разработке вопросов): анализ предметной области, проектирование ИС, реализация ИС.

5. Перечень материалов приложения (наличие чертежей, таблиц, графиков, схем, программных продуктов, иллюстративного материала и т.п.): техническое задание.

6. Консультанты по бакалаврской работе:

Консультант по безопасности и экологичности – Булгаков А. Б., доцент, канд. техн. наук

7. Дата выдачи задания: 02.10.2023г.

Руководитель бакалаврской работы: доцент И.М. Акилова

Задание принял к исполнению(дата): 02.10.2023г. Колосов М. А.

РЕФЕРАТ

Бакалаврская работа содержит 153 страниц, 51 рисунок, 12 таблиц, 4 приложения

ИНФОРМАЦИОННАЯ СИСТЕМА, REST-СЕРВИС, API,
POSTGRESQL

В данной работе выполнена разработка информационной системы мониторинга учета грузоперевозок с использованием языка программирования с#.

Объектом исследования являются – организация и ведение учета грузоперевозок.

Целью данной работы является разработка ИС учета грузоперевозок.

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты и нормативные документы:

ГОСТ 19. 001-77. Единая система программной документации (ЕСПД). Общие положения.

ГОСТ 19. 002-80. ЕСПД. Схемы алгоритмов и программ. Правила выполнения.

ГОСТ 19. 003-80. ЕСПД. Схемы алгоритмов и программ. Обозначения условные графические.

ГОСТ 19. 004-80. ЕСПД. Термины и определения.

ГОСТ 19. 101-77. ЕСПД. Виды программ и программных документов.

ГОСТ 19. 102-77. ЕСПД. Стадии разработки.

ГОСТ 19. 201-78. ЕСПД. Техническое задание. Требования к содержанию и оформлению.

ГОСТ 19. 301-79. ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению.

ГОСТ 19. 402-78. ЕСПД. Описание программы.

ГОСТ 19. 502-78. ЕСПД. Описание применения.

ГОСТ 2. 701-84. Схемы. Типы и виды. Общие требования к выполнению.

ГОСТ 19. 701-90. Схемы алгоритмов, программ, данных и систем.

ГОСТ 19. 102-77. Стадии разработки.

ГОСТ 19. 404-79. Пояснительная записка. Требования к содержанию и оформлению.

ГОСТ Р 53891-2010 Информационные технологии. Средства защиты информации. Классификация и общие требования.

СОКРАЩЕНИЯ, ОБОЗНАЧЕНИЯ И ОПРЕДЕЛЕНИЯ

ИС – информационная система

СУБД – система управления базами данных

БД – база данных

IDE – интегрированная среда разработки

ПО – программное обеспечение

ОС – операционная система

MVS – Microsoft Visual Studio

ПЭВМ – персональная электронно-вычислительная машина

СОДЕРЖАНИЕ

Введение	8
1 Анализ предметной области	11
1.1 Описание предметной области	11
1.2 Общие сведения о компании ООО «Континент»	12
1.3 Обоснование необходимости создания ИС	15
1.4 Обзор существующих решений	15
1.5 Требования к информационной системе	18
2 Разработка базы данных информационной системы	20
2.1 Инфологическое проектирование базы данных	20
2.2 Логическое проектирование базы данных	23
2.3 Физическое проектирование базы данных	25
3 Разработка информационной системы	31
3.1 Выбор средств разработки	31
3.2 Определение системных требований	32
3.3 Определение функциональных возможностей	32
3.4 Разработка REST-сервиса	34
3.4.1 Реализация программного кода основных функций	34
3.4.2 Описание методов взаимодействия	36
3.4.3 Тестирование основных функций	41
3.5 Разработка клиентского программного обеспечения	50
3.5.1 Описание взаимодействия	50
3.5.2 Интерфейс пользователя	52
3.5.2 Тестирование взаимодействия	56
4 Защита данных и надежность информационной системы	65
4.1 Организация защиты данных	65
4.2 Надежность ИС	66
5 Безопасность и экологичность	67

5.1 Безопасность	67
5.1.1 Организация рабочего места	68
5.1.2 Освещение	70
5.1.3 Микроклимат	72
5.1.4 Графический интерфейс пользователя	73
5.2 Экологичность	74
5.3 Чрезвычайные ситуации	76
5.3.1 Меры пожарной безопасности	76
5.3.2 Рекомендованные действия при угрозе совершения террористического акта	77
Заключение	80
Библиографический список	81
Приложение А. Техническое задание	83
Приложение Б. ER-диаграмма модели в нотации Чена	91
Приложение В. Логическая схема данных	92
Приложение Г. Описание методов взаимодействия	93

ВВЕДЕНИЕ

Большинство современных предприятий и организаций все чаще применяют для улучшения своей деятельности различные информационные системы (далее ИС). ИС внедряются в различные направления деятельности предприятий, это и бухгалтерский учет, и управление персоналом, и предоставление различных услуг клиентам. Информационные технологии помогают выполнять работу человеку в различных сферах деятельности, так как современные информационные технологии позволяют автоматизировать большинство задач, что приводит к улучшению экономических показателей предприятия. С помощью информационных технологий можно производить сложнейшие для человека вычисления за секунды, и при этом избежать ошибок вычислений. Кроме того, информационные системы обрабатывают огромные массивы взаимосвязанной информации, которые нуждаются в хранении, обновлении, корректировке, а также можно выполнять разнообразные запросы к базам данных.

Актуальность выпускной квалификационной работы состоит в необходимости разработки и внедрении информационной системы для компании по грузоперевозкам, что позволит автоматизировать работу в компании оказывающей услуги по грузоперевозкам.

Объектом исследования выпускной квалификационной работы является процесс управления заявками в компании, занимающейся грузоперевозками, а также вывод необходимой отчетной документации.

Предметом исследования выпускной квалификационной работы является процесс автоматизации бизнес-процессов в компании по грузоперевозкам и управление заявками на оказание услуг по грузоперевозкам, а также вывод необходимой отчетной документации.

Цель выпускной квалификационной работы: разработать информационную систему учета грузоперевозками.

Задачи выпускной квалификационной работы:

- провести анализ предметной области;
- спроектировать концептуальную модель информационной системы по грузоперевозкам;
- выполнить обзор и анализ известных аналогов информационной системы по грузоперевозкам;
- разработать логическую модель данных информационной системы по грузоперевозкам;
- разработать физическую модель данных информационной системы по грузоперевозкам;
- подобрать инструментальные средства для разработки информационной системы и базы данных системы.

Методы исследования: реинжиниринг процессов, методы структурного и объектно-ориентированного анализа и проектирования автоматизированных информационных систем, реляционное моделирование данных.

Практическая значимость работы представляется в разработке информационной системы для компании по грузоперевозкам, которая позволит сэкономить материальные и временные ресурсы и соответственно повысит прибыль компании.

В первой главе проводится анализ предметной области по грузоперевозкам, а также разрабатывается концептуальная модель, описывается процесс формирования заявки и организация отчетности с применением автоматизированного подхода.

Во второй главе проводится проектирование логической и физической моделей данных, выбор инструментальных средств для реализации информационной системы, и, собственно, сам процесс создания информационной системы по грузоперевозкам на основе полученных результатов в первой главе.

В третьей главе приводятся основные функциональные возможности информационной системы, такие как отслеживание груза, сроки доставки, заявка

на перевозку и бухгалтерские документы. А также приведено тестирование основных кейсов ИС для компании по грузоперевозкам.

В заключении подводятся основные итоги и выводы по работе.

В результате выполнения выпускной квалификационной работы была разработана информационная система по грузоперевозкам, которая позволила уменьшить временные затраты на получения и обработки информации. Кроме того, данная информационная система позволила уменьшить расходы компании как материальные, так и временные ресурсы, на формирование отчетной документации.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Описание предметной области

В наши дни грузоперевозки занимают важное место, так как они играют одну из важных ролей в жизни человечества. Кроме того, грузоперевозки, являются важным аспектом деловой жизни. Грузоперевозки – сфера, требующая особого подхода и наличия базовых знаний. Перед тем, как начинать бизнес в сфере грузоперевозок, нужно тщательно учитывать специфику работы различных видов транспорта, а также брать в учет определенные обстоятельства. Все время в мире что-то перевозят, транспортируют, создавая товарные связи. Ни для кого не секрет, что деньги можно переводить разными способами, но в случае с товаром – другая ситуация. Иногда его нужно перевозить в другой город или страну. В таких случаях уже не воспользуешься просто машиной и наемным водителем, нужно что-то более масштабное и серьезное. Существует множество компаний, которые обеспечивает разнообразие услуг, связанных с транспортировкой товара, его перевозкой в другую страну, а для этого нужно иметь необходимые документы, в которых учтены все международные правила и требования. Конечно, есть и негативный аспект таких перевозок: их цена. Ведь расстояние может быть огромным – такая же будет стоимость.

Грузоперевозки имеют ряд особенностей:

- отсутствие долгосрочного планирования;
- ограниченные возможности транспорта;
- ограниченное время работы некоторых видов транспорта;
- недостаточный охват;
- большие расстояния;
- высокая конкуренция;
- специфичные логистические маршруты;
- «растянутые» сроки;

- упаковка, страховка, разрешительная и сопроводительная документация.

Основные функции перевозок: принять заявку, оказать услугу по перевозке груза, загрузка и выгрузка груза.

Автоперевозки – наиболее популярный вид перевозки грузов. Основные преимущества автоперевозки:

- экономичность;
- быстрая доставка;
- гибкое планирование маршрутов;
- контроль груза во время перевозки.

1.2 Общие сведения о компании ООО «Континент»

Транспортная компания ООО «Континент» на рынке перевозок уже 4 года, находится по адресу г. Благовещенск, ул. Калинина, д.103. Основной вид деятельности организации: код ОКВЭД 49.41 - деятельность автомобильного грузового транспорта. ОКВЭД 49.41 - эта группировка включает: все виды перевозок грузов автомобильным транспортом по автомобильным дорогам: опасных грузов, крупногабаритных и/или тяжеловесных грузов, грузов в контейнерах и транспортных пакетах, скоропортящихся грузов, массовых навалочных грузов, сельскохозяйственных грузов, грузов строительной отрасли, грузов промышленных предприятий, прочих грузов

На рисунке 1 показана организационная структура управления ООО «Континент». Генеральный директор отвечает за стратегическое планирование и принятие решений при определении направления развития компании. Директор по маркетингу отвечает за создание и продвижение продукта или услуги на рынке. Он отвечает за разработку маркетинговых стратегий, определение целевой аудитории и позиционирование товара или услуги. Кроме того, директор по маркетингу отвечает за управление брендом компании и формирование ее имиджа. Коммерческий директор принимает активное участие в формировании ценовой политики компании. Анализирует рыночные условия, изучает

структуру затрат и рентабельность продукции или услуг, чтобы установить оптимальные цены. Он работает над установлением и поддержанием стратегических партнерств с другими компаниями. Исполнительный директор планирует, разрабатывает и внедряет стратегических планы для компании, организации экономически обоснованной операционной деятельности. Главный бухгалтер в компании назначается генеральным директором. Его основная функция – управление процессом бухгалтерского учёта и документооборота в организации.



Рисунок 1 – Организационная структура

Внешний документооборот изображен на рисунке 2 и состоит из

- Клиентов;
- Банка ПАО Сбербанк;
- Субподрядчика в автоперевозках ООО "Трансэкспедиция";
- Грузополучателей;
- Страховой компании СПАО «Ингосстрах»;
- УФНС России по Амурской области;
- УПФ РФ по Амурской области.
- Таможенный представитель "СТТ Логистика"

1.3 Обоснование необходимости создания ИС

Кажется очевидным, что прозрачность в ответственном деле, от которого зависят доходы не только фирмы и руководства, но и самого водителя, способна повысить заинтересованность в честном исполнении обязанностей. Во-первых, непродуктивные расходы резко упадут, и за счет этого доходы будут соответствовать расчетным величинам. Во-вторых, стабильность работы и высокий уровень ответственности со стороны фирмы и её водителей привлекательны для серьезных клиентов, которые рассчитывают на долгосрочную перспективу. В-третьих, экономия топлива и времени становится реально выполнимой, что также повышает уровень доходов всего предприятия.

На предприятии используются стандартные информационные технологии (ИТ), а именно: несколько персональных компьютеров, объединенных в локальную сеть; на сервере расположена общая база данных предприятия и аналитические программы, на рабочих станциях организованы АРМ специалистов.

Программное обеспечение представляет из себя сетевую и консольные версии Windows XP, офисные программы, антивирусы, архиваторы. Кроме того, различного рода аппаратура, средства связи, выход в интернет, оргтехника дают возможность эффективно работать.

Для осуществления экономической деятельности предприятие не использует программы таких как 1С торговля и другие программы, поэтому было предложено решение, создать свою систему учета грузоперевозок.

1.4 Обзор существующих решений

В настоящее время существует множество программных продуктов по учету перевозок для предприятий, например: 1С:Предприятие 8, 1С – Логистика: Управление перевозками, Умная Логистика, АвтоПеревозки, ГрузоПлан, КиберЛог, НоваТранс и многие другие.

К наиболее распространенным относятся:

- «1С - Логистика: Управление перевозками»;

- «Умная Логистика»;
- «АвтоПеревозки».

«1С – Логистика: Управление перевозками» – программа на платформе «1С – Предприятие 8», предназначена для решения широкого спектра задач автоматизации учета и управления, стоящих перед быстро развивающимися современными предприятиями.

Достоинства данной программы:

- с помощью программы «1С – Логистика: Управление перевозками» можно вести весь действующий учет перевозок;
- легкость и простота адаптации к условиям работы практически любого предприятия, специфике его технологических и организационных требований; программа обладает высокой производительностью, что дает возможность решать с ее помощью самые сложные задачи.

Недостатки программы «1С – Логистика: Управление перевозками»:

- каждое предприятие индивидуально, поэтому для эффективной его работы, требуются индивидуальные решения по автоматизации бизнес-процессов;
- при переходе на «1С – Логистика: Управление перевозками» с другой программы могут возникнуть трудности при переносе информации из одной программы в другую;
- программа достаточно сложна в освоении и требует специального обучения пользователей;
- высокая стоимость лицензии и обслуживания программного продукта.

Программа «Умная Логистика» – это программный продукт для оперативной работы транспортно-экспедиционных компаний. Программа обладает внушительными инструментами, которые позволяют моделировать всю цепочку бизнес-процессов автотранспортного предприятия, начиная от заполне-

ния путевого листа, загрузки и выгрузки товара и заканчивая сбором информации для полного анализа деятельности предприятия и дальнейшего использования результатов.

Достоинства программы «Умная Логистика»:

- работа в online в любое время и в любом месте;
- оптимизация всех бизнес-процессов;
- интерфейс на базе 1С;
- бесплатные обновления;
- автоматическое заполнение документов, заявок и отчетов;
- данные защищены SSL протоколами, хранятся на серверах в дата-центре уровня Tier 3.

В программе приведено подробное описание всех функций и возможностей. «Умная Логистика» поставляется в тарифах «Серебряный» «Золотой» и «Платиновый».

Недостатки программы «Умная Логистика»:

- высокая стоимость тарифов;
- пользователь должен обладать определенными компьютерными навыками;
- не слишком удобный интерфейс, усложненный лишними параметрами.

Программа «АвтоПеревозки» – это программа тоже создана для учета грузоперевозок. Одним из важных преимуществ является удобный интерфейс.

Основные функции программы «АвтоПеревозки»:

- заполнение, оформление, обработка, учет путевых листов автомобилей;
- производить расчет норм расхода топлива;
- вести заявки клиентов и на их основе заводить путевые листы;
- получать подробные отчеты по работе автомобилей и водителей;
- вести учет замененных частей автомобилей, ремонтов.

Дополнительные возможности системы:

- работа с несколькими организациями;
- работа в сети;
- программа может одновременно работать на нескольких компьютерах с одной базой данных;
- бланки документов хранятся в формате Word и Excel.

Программе также нужен или грамотный специалист, приглашенный со стороны обслуживающего предприятия, или штатный сотрудник, обученный тем же предприятием. Другие представители программного обеспечения созданы примерно также, как и предыдущие, это или программы из семейства 1С, или программы, созданные другими разработчиками со своими достоинствами и недостатками, которые исправлять практически некому кроме программистов именно этой организации. Следовательно, решения, существующие в данный период, являются универсальными. Только они платные, трудны в освоении и слишком неудобны. Таким образом, было принято решение о необходимости разработки собственной программы, которая будет отвечать всем необходимым требованиям, и будет иметь простой и удобный интерфейс. И не потребуется покупать лицензию, что является немаловажным фактом. Программный продукт, разрабатываемый в рамках выпускной квалификационной работы, ориентирован на пользователей, не имеющих специальных компьютерных навыков, а именно для работников автотранспортного предприятия. При разработке интерфейса программы минимизируется вводимая с клавиатуры информация. программный лингвистический автотранспортный грузоперевозка.

1.5 Требования к информационной системе

Анализа предметной области и поставленных задач позволил выделить следующие функции и требования, которые необходимо реализовать в ИС.

ИС должна представлять собой серверную часть (REST-сервис) доступный в режиме 7/24/365 и клиентского приложения.

ИС должна состоять из следующих взаимосвязанных модулей:

- услуги и цены;
- сроки доставки грузов;
- заявка на перевозку;
- составление отчетов;
- запрос бухгалтерских документов.

Для поддержания и эксплуатации ИС от сотрудников компании требуются только пользовательские навыки работы с программными продуктами.

Связь с клиентами поддерживается через электронную почту и личный кабинет менеджера и клиента. Клиентское приложение должно работать на телефоне и ПК, и предоставляется возможность управление заявками в онлайн-режиме.

2 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Для реализации информационной системы необходимо разработать базу данных для формирования заказов и учета грузоперевозок. Эта база будет служить хранилищем для всей информации, необходимой для корректного функционирования информационной системы.

Качество базы данных влияет на эффективность всей системы. Ее структура должна быть способна обработать и хранить большие объемы данных, а также обеспечивать быстрый доступ к информации.

В процессе разработки используется многоуровневый подход к ее проектированию. Весь процесс разделен на три основных этапа: создание инфологической модели, логической модели и физической модели.

2.1 Инфологическое проектирование базы данных

Инфологическая модель базы данных – это концептуальное представление данных, отображающее основные сущности и связи между ними. В нашем случае мы имеем дело с такими сущностями, как: «Сотрудники», «Роли», «Должности», «Клиенты», «Акты о расходах», «Типы расхода», «Автомобили», «Путевой лист», «Виды груза», «Заказы», «Счета по заказам».

Сотрудники (Users). Хранит информацию по сотрудникам, которые могут авторизовываться в системе и работать с данными. Атрибуты сущности включают в себя ИД, фамилию, имя, отчество, должность, дату рождения, дату принятия на работу и дату увольнения.

Роли (Roles). Представляет собой информацию о ролях, данная сущность позволяет разграничивать доступы взаимодействия с данными для сотрудников. Атрибуты включают ИД, название.

Должности (UserPositions). Данные о наименовании должностей для сотрудников. Атрибуты данной сущности включают ИД, название.

Клиенты (Customers). Хранится информация по клиентам, для которых формируется заказ по доставке груза. Атрибуты включают в себя ИД,

наименование, тип клиента (ЮЛ или ФЛ), ИНН (для ЮЛ), паспорт (для ФЛ), телефон и почта.

Акты о расходах (OrderExpenses). Представляет собой информацию о расходах по заказам. Атрибуты сущности ИД, дата, заказ, тип расхода, сумма и описание.

Типы расхода (OrderExpenseTypes). Хранит информацию о типах расходов по заказам. Атрибуты сущности включают в себя ИД и название.

Автомобили (Cars). Содержит данные об автомобилях. Атрибуты сущности ИД, наименование, государственный номер, грузоподъемность, расход ГСМ, год выпуска.

Путевые листы (Waybills). Содержит информацию об автомобилях и водителях по заказам. Атрибуты включают в себя ИД, дата, заказ, водитель, автомобиль, расстояние.

Виды груза (OrderCargoType). Хранит информацию о видах груза для заказов. Атрибуты сущности ИД, название.

Заказы (Orders). Хранит информацию по заказам, одна из основных сущностей для формирования отчетов. Атрибуты сущности включают в себя ИД, дата, клиент, вид груза, масса груза, пункт назначения, пункт отправления, счет, менеджер, статус, комментарий и путевой лист.

Счета по заказам (OrderPayments). Содержит информацию счетов по заказам. Атрибуты включают в себя ИД, дата, заказ, сумма, бухгалтер, статус оплаты, комментарий.

Статусы заказа (OrderStatuses). Представляет собой информацию о статусах для заказов. Атрибуты сущности ИД, название, цвет текста, цвет фона.

Связи между этими сущностями, следующие:

– сотрудники связаны с ролями как многие ко многим, т.е. сотрудник может иметь несколько ролей, а для одной роли может соответствовать множество сотрудников;

– сотрудники связаны с должностями как один ко многим, т.е. для одного сотрудника может соответствовать только одна должность, а к одной должности может быть привязано множество сотрудников;

– сотрудники связаны с путевыми листами как один ко многим, т.е. для одного путевого листа может соответствовать только один сотрудник, а для одного сотрудника может соответствовать множество значений путевого листа;

– сотрудники связаны с заказами как один ко многим, для одной записи заказа соответствует один и только один сотрудник, как для сотрудника может быть множество заказов;

– сотрудники связаны с счетами по заказам как один ко многим, для одной записи по счетам будет соответствовать только один сотрудник, а для сотрудника может соответствовать множество счетов;

– акты о расходах связаны с типами расходов как один ко многим, т.е. для одного акта будет соответствовать только один тип, в то время как, для одного типа может соответствовать множество актов;

– акты о расходах связаны с заказами как один ко многим, т.е. для одного заказа может соответствовать множество актов, в то время как для одного акта может быть только один заказ;

– путевой лист связан с автомобилями как один ко многим, т.е. для одного путевого листа будет соответствовать только один автомобиль, а для одного автомобиля может соответствовать множество записей путевого листа.

– виды груза связаны с заказами как один ко многим, т.е. для одной записи вида груза соответствует множество записей по заказам, в то время как для одного заказа может быть указан только один вид груза;

– заказы связаны с путевыми листами как один к одному, т.е. для одной записи заказа будет соответствовать только одна запись путевого листа;

– заказы связаны с счетами по заказам как один ко многим, т.е. для одной записи заказа будет соответствовать множество записей счетов по заказам, а для одной записи будет счета будет только один заказ;

– заказы связаны с клиентами как один ко многим, т.е. для одной записи клиента будет соответствовать множество заказов, а для одной записи заказа соответствует один клиент.

В приложении Б представлена ER-диаграмма для данной модели в нотации Чена.

2.2 Логическое проектирование базы данных

При проектировании логической модели базы данных одним из ключевых моментов является процесс нормализации отношений. Нормализация – это методологический подход, целью которого является устранение дублирования данных и обеспечение связности и эффективности базы данных путем установления набора правил или норм:

Применяя процедуры нормализации, разрабатываются следующие таблицы:

– таблица «*UserPositions*». Содержит поля: «Id» (первичный ключ) и «Title». Первичный ключ используется для уникальной идентификации каждой должности в системе. Нормализация гарантирует, что каждая должность имеет уникальный идентификатор, и вся информация находится в одной записи;

– таблица «*Users*». Содержит поля: «Id» (первичный ключ), «LastName», «FirstName», «MiddleName», «PositionId» (внешний ключ), «Birthday», «EmploymentDate», «DismissalDate», «UserName», «Email», «PasswordHash» и «PhoneNumber». Процесс нормализации в данном случае гарантирует, что вся информация о конкретной записи находится в одном месте, и она ясно связана с определенной должностью через внешний ключ;

– таблица «*Roles*». Включает следующие поля: «Id» (первичный ключ) и «Name»;

– таблица «*OrderPayments*». Включает в себя поля: «Id» (первичный ключ), «OrderId» (внешний ключ), «Date», «Amount», «AccountantId» (внешний ключ), «Status», и «Comment». Нормализация в этом случае обеспечивает

связность данных и гарантирует, что каждый счет связан с определенным заказом и сотрудником по внешнему ключу;

– таблица «*Waybills*». Содержит поля: «*Id*» (первичный ключ), «*OrderId*» (внешний ключ), «*Date*», «*DriverId*» (внешний ключ), «*CarId*» (внешний ключ) и «*Distance*». Аналогично таблице «*OrderPayments*», нормализация обеспечивает связность данных и гарантирует, что каждый путевой лист явно связана с заказом, водителем и автомобилем через внешний ключ;

– таблица «*OrderStatuses*». Включает следующие поля: «*Id*» (первичный ключ), «*Title*», «*ForegroundColor*» и «*BackgroundColor*»;

– таблица «*Orders*». Включает в себя поля: «*Id*» (первичный ключ), «*Date*», «*CustomerId*» (внешний ключ), «*CargoTypeId*» (внешний ключ), «*CargoMass*», «*DeparturePoint*», «*Destination*», «*ManagerId*» (внешний ключ), «*StatusId*» (внешний ключ), «*Comment*» и «*WaybillId*». В данном случае нормализация гарантирует, что вся информация о конкретной записи находится в одном месте, и она ясно связана с клиентом, типом груза, сотрудником, статусом и путевым листом по внешнему ключу;

– таблица «*Customers*». Содержит поля: «*Id*» (первичный ключ), «*Title*», «*Type*», «*INN*», «*Passport*», «*Phone*» и «*Email*». Первичный ключ используется для уникальной идентификации каждого клиента в системе. Нормализация гарантирует, что каждый клиент имеет уникальный идентификатор, и вся информация находится в одной записи;

– таблица «*OrderExpenseTypes*». Включает следующие поля: «*Id*» (первичный ключ) и «*Title*»;

– таблица «*OrderExpenses*». Включает в себя поля: «*Id*» (первичный ключ), «*OrderId*» (внешний ключ), «*Date*», «*ExpenseTypeId*» (внешний ключ), «*Amount*» и «*Comment*». В данном случае нормализация гарантирует, что вся информация о конкретной записи находится в одном месте, и она ясно связана с заказом и типом расхода по внешнему ключу;

– таблица «CargoTypes». Содержит следующие поля: «Id» (первичный ключ) и «Title»;

– таблица «Cars». Содержит поля: «Id», «Title», «NumberPlate», «CarryingCapacity», «FuelConsumption», и «Year».

Нормализация обеспечивает ясную и эффективную структуру для нашей базы данных, минимизирует дублирование данных и облегчает манипуляции с данными.

Логическая схема приведена в приложении В.

2.3 Физическое проектирование базы данных

В процессе физического проектирования базы данных определяются конкретные атрибуты каждой таблицы, типы данных, ограничения и индексация. В результате физического проектирования получились следующие таблицы.

Таблица 1 – Физическая структура таблицы «UserPositions»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да
Title	Строка	NOT NULL	VARCHAR(128)	Нет

Таблица 2 – Физическая структура таблицы «Users»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да
LastName	Строка	NOT NULL	VARCHAR(128)	Нет
FirstName	Строка	NOT NULL	VARCHAR(128)	Нет
MiddleName	Строка	NULL	VARCHAR(128)	Нет
PositionId	Целое число	FOREIGN KEY	INTEGER	Нет
Birthday	Дата и время	NULL	TIMESTAMP	Нет

Employment-Date	Дата и время	NULL	TIMESTAMP	Нет
DismissalDate	Дата и время	NULL	TIMESTAMP	Нет
UserName	Строка	NOT NULL	VARCHAR(256)	Нет
Email	Строка	NOT NULL	VARCHAR(256)	Нет
PasswordHash	Строка	NOT NULL	TEXT	Нет
PhoneNumber	Строка	NULL	VARCHAR(16)	Нет

Таблица 3 – Физическая структура таблицы «Roles»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да
Name	Строка	NOT NULL	VARCHAR(256)	Нет

Таблица 4 – Физическая структура таблицы «OrderPayments»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да
OrderId	Целое число	FOREIGN KEY	INTEGER	Нет
Date	Дата и время	NOT NULL	TIMESTAMP	Нет
Amount	Дробное число	NOT NULL	DOUBLE	Нет
AccountantId	Целое число	FOREIGN KEY	INTEGER	Нет
Status	Строка	NOT NULL	VARCHAR(128)	Нет
Comment	Строка	NULL	TEXT	Нет

Таблица 5 – Физическая структура таблицы «Waybills»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да
OrderId	Целое число	FOREIGN KEY	INTEGER	Нет
Date	Дата и время	NOT NULL	TIMESTAMP	Нет
DriverId	Целое число	FOREIGN KEY	INTEGER	Нет
CarId	Целое число	FOREIGN KEY	INTEGER	Нет
Distance	Дробное число	NOT NULL	DOUBLE	Нет

Таблица 6 – Физическая структура таблицы «OrderStatuses»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да
Title	Строка	NOT NULL	VARCHAR(128)	Нет
Foreground- Color	Строка	NULL	VARCHAR(32)	Нет
Background- Color	Строка	NULL	VARCHAR(32)	Нет

Таблица 7 – Физическая структура таблицы «Orders»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да

CustomerId	Целое число	FOREIGN KEY	INTEGER	Нет
Date	Дата и время	NOT NULL	TIMESTAMP	Нет
CargoTypeId	Целое число	FOREIGN KEY	INTEGER	Нет
CargoMass	Дробное число	NOT NULL	DOUBLE	Нет
Departure- Point	Строка	NOT NULL	VARCHAR(512)	Нет
Destination	Строка	NOT NULL	VARCHAR(512)	Нет
ManagerId	Целое число	FOREIGN KEY	INTEGER	Нет
StatusId	Целое число	FOREIGN KEY	INTEGER	Нет
Comment	Строка	NULL	TEXT	Нет
WaybillId	Целое число	FOREIGN KEY	INTEGER	Нет

Таблица 8 – Физическая структура таблицы «Customers»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да
Title	Строка	NOT NULL	VARCHAR(128)	Нет
Type	Строка	NOT NULL	VARCHAR(12)	Нет
INN	Строка	NULL	VARCHAR(11)	Нет
Passport	Строка	NULL	VARCHAR(32)	Нет
Phone	Строка	NULL	VARCHAR(16)	Нет
Email	Строка	NOT NULL	VARCHAR(32)	Нет

Таблица 9 – Физическая структура таблицы «OrderExpenseTypes»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да
Title	Строка	NOT NULL	VARCHAR(128)	Нет

Таблица 10 – Физическая структура таблицы «OrderExpenses»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да
OrderId	Целое число	FOREIGN KEY	INTEGER	Нет
Date	Дата и время	NOT NULL	TIMESTAMP	Нет
ExpenseType- eId	Целое число	FOREIGN KEY	INTEGER	Нет
Amount	Дробное число	NOT NULL	DOUBLE	Нет
Comment	Строка	NULL	TEXT	Нет

Таблица 11 – Физическая структура таблицы «CargoTypes»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да
Title	Строка	NOT NULL	VARCHAR(128)	Нет

Таблица 12 – Физическая структура таблицы «Cars»

Атрибут	Тип данных	Условие	Формат данных	Индексация
Id	Целое число	PRIMARY KEY	IDENTITY	Да
Title	Строка	NOT NULL	VARCHAR(128)	Нет
NumberPlate	Строка	NOT NULL	VARCHAR(12)	Нет
Carrying- Capacity	Дробное число	NOT NULL	DOUBLE	Нет
FuelConsump- tion	Дробное число	NOT NULL	DOUBLE	Нет
Year	Целое число	NOT NULL	INTEGER	Нет

Все описанные выше таблицы и их атрибуты были проектированы с учетом особенностей работы с СУБД PostgreSQL. Это включает в себя выбор типов данных, которые являются наиболее подходящими для PostgreSQL, и учет функционала индексации, предоставляемого этой системой.

На этапе физического проектирования были также учтены такие факторы, как производительность и масштабируемость базы данных. Итоговая физическая структура базы данных полностью соответствует требованиям и целям проекта, обеспечивая надежное и эффективное хранение данных, необходимых для мониторинга заказов

3 РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1 Выбор средств разработки

Для разработки программного продукта была выбрана среда разработки Microsoft Visual Studio 2022 и язык программирования C Sharp (c#).

Microsoft Visual Studio (далее MVS) – это интегрированная среда разработки (IDE), разработанная компанией Microsoft. MVS предназначен для разработки различных программных приложений, включая настольные приложения, веб-приложения, мобильные приложения и игры.

MVS включает в себя множество инструментов для разработки, таких как редактор кода, отладчик, компилятор, систему контроля версий, инструменты тестирования и многие другие. Она поддерживает множество языков программирования, включая C++, C#, Visual Basic, Python, JavaScript.

Основными преимуществами Microsoft Visual Studio являются:

- *Доступность*: MVS поставляется с тремя видами лицензии, Community, Professional и Enterprise, одна из которых является бесплатная и можно использовать без ограничений для личного пользования;
- *Широкие возможности*: MVS предоставляет широкий набор инструментов для разработки различных типов приложений, что делает ее удобной для использования в различных проектах;
- *Интеграция*: MVS интегрируется с многими другими продуктами Microsoft, такими как Microsoft Azure, Microsoft Office и Microsoft SQL Server, что позволяет легко интегрировать приложения с другими продуктами Microsoft;
- *Обучение*: MVS предоставляет обширную документацию, учебные ресурсы и сообщество пользователей, которые могут помочь новым пользователям быстро освоиться с платформой;
- *Поддержка*: Microsoft предоставляет широкую поддержку для MVS, включая обновления, исправления ошибок и техническую поддержку;

Недостатки Microsoft Visual Studio включают в себя:

- *Сложность*: MVS может быть сложной для новых пользователей, особенно если они не знакомы с инструментами разработки;
- *Требования к системе*: MVS может потребовать высоких системных требований, особенно для больших проектов.

В целом, Microsoft Visual Studio является мощной и широко используемой IDE для разработки различных приложений. Она предоставляет широкий набор инструментов и обширную поддержку из-за чего она и была выбрана в качестве среды разработки данного проекта.

3.2 Определение системных требований

Для корректного функционирования и отзывчивого интерфейса REST-сервиса (серверная часть), сервер должен соответствовать следующим системным требованиям:

- *Процессор*: 4 ядра с частотой 2 ГГц или более (x64);
- *Оперативная память*: 8 ГБ;
- *Объем жесткого диска*: 100 ГБ;
- *Доступ к интернету*;
- *ОС*: Ubuntu.

3.3 Определение функциональных возможностей

Определение функциональных возможностей разрабатываемой ИС необходимо для того, чтобы обеспечить эффективную работу с данными, а именно сохранение, изменение и удаление с базы. Они должны соответствовать концепции программного обеспечения.

Первая важная функциональная возможность – это разграничение прав доступа к данным, а именно аутентификация с последующей авторизацией пользователя, работающего с данными.

Аутентификация – это процесс проверки подлинности пользователя и его идентификации. Она используется для установления того, что пользователь, который пытается получить доступ к системе, действительно тот, за кого

себя выдает. Для этого пользователь должен предоставить уникальные данные, которые могут быть проверены на подлинность, такие как имя пользователя и пароль, биометрические данные, токены или сертификаты.

Авторизация – это процесс проверки, имеет ли пользователь право на доступ к определенным ресурсам и функциональности в системе. Это означает, что после того, как пользователь был аутентифицирован, система должна проверить его права доступа, чтобы определить, какие действия он может выполнить в системе и какие ресурсы он может использовать.

Таким образом, аутентификация и авторизация являются важными составляющими безопасности информационных систем. Аутентификация гарантирует, что пользователь является тем, за кого себя выдает, а авторизация гарантирует, что пользователь имеет право на доступ к ресурсам и функциональности в системе. Обе функции должны работать совместно, чтобы обеспечить безопасность системы и защитить ее от несанкционированного доступа и злоупотреблений.

Вторая важная функциональная возможность приложения вывод, изменение и удаление содержимого таблиц: «сотрудники», «роли», «должности», «клиенты», «автомобили», «путевые листы», «заказы», «виды груза», «статусы заказа», «акты о расходах», «типы расхода» и «счета по заказам».

Третья функциональная особенность – разделение возможностей пользователей: администраторы, операторы и водители. Для водителей доступен только просмотр заказов, за которым они прикреплены и смену статуса по заказу, факт доставки. Оператор может редактировать и создавать записи по всем таблицам, кроме «сотрудники», «роли», «должности» и «статусы заказов», а также получать отчеты и документы на печать. У администраторов полный доступ.

3.4 Разработка REST-сервиса

3.4.1 Реализация программного кода основных функций

Самая основная функция, без которой дальнейшее взаимодействие с REST-сервисом будет невозможным, это аутентификации. Код данной функции приведен на рисунке 4.

```
12 [Tags("OAuth")]
13 [Route("api/OAuth")]
14 public class Authenticate: EndpointBaseAsync.WithRequest<AuthenticateRequest>.WithActionResult {
15     private readonly UserManager<User> _userManager;
16     private readonly ITokenClaimsService _tokenClaimsService;
17
18     Ссылка: 0
19     public Authenticate( UserManager<User> userManager, ITokenClaimsService tokenClaimsService ) {
20         _userManager = userManager;
21         _tokenClaimsService = tokenClaimsService;
22     }
23
24     [HttpPost("authenticate")]
25     [ProducesResponseType(typeof(AuthenticateResponse), StatusCodes.Status200OK)]
26     [ProducesResponseType(typeof(ErrorDetails), StatusCodes.Status400BadRequest)]
27     [ProducesResponseType(typeof(ErrorDetails), StatusCodes.Status500InternalServerError)]
28     Ссылка: 0
29     public override async Task<ActionResult> HandleAsync( [FromBody] AuthenticateRequest request, CancellationToken cancellationToken = default ) {
30         var normalizedUserName = request.UserName.ToUpper();
31         var loggedUser = _userManager.Users
32             .FirstOrDefault(u =>
33                 u.NormalizedEmail == normalizedUserName
34                 || u.NormalizedUserName == normalizedUserName
35             );
36         if ( loggedUser == null || !await _userManager.CheckPasswordAsync(loggedUser, request.Password) )
37             throw WebApiException.BadRequest("Неверный email/логин или пароль");
38
39         var expiresToken = DateTime.UtcNow.AddHours(1);
40         var response = new RefreshTokenResponse {
41             Token = await _tokenClaimsService.GetTokenAsync(loggedUser.UserName, expiresToken),
42             ExpiresToken = expiresToken,
43             RefreshToken = await _tokenClaimsService.GetRefreshTokenAsync(loggedUser.UserName)
44         };
45         return Ok(response);
46     }
47 }
```

Рисунок 4 – Код метода аутентификации

Для вызова данного метода, необходимо отправить POST запрос по адресу «/api/OAuth/authenticate». В тело запроса необходимо передать логин/email с паролем, если пользователь существует и пароль указан верный, то в ответ будет сформирован токен взаимодействия со сроком «жизни», а также токен сброса, необходимый для обновления токена взаимодействия без повторного ввода логина/email с паролем. Пример взаимодействия представлен в п.3.4.3.

Вторая функция по значимости, это функция обновления токена. Данный метод позволяет «продлевать» сессию без повторного ввода логина/email и пароля, а необходимо передать токен взаимодействия и токен сброса. Код приведен на рисунке 5.

```

14 [Tags("OAuth")]
15 [Route("api/OAuth")]
16 public class RefreshToken: EndpointBaseAsync.WithRequest<RefreshTokenRequest>.WithActionResult {
17     private readonly UserManager<User> _userManager;
18     private readonly ITokenClaimsService _tokenClaimsService;
19
20     public RefreshToken( UserManager<User> userManager, ITokenClaimsService tokenClaimsService ) {
21         _userManager = userManager;
22         _tokenClaimsService = tokenClaimsService;
23     }
24
25     [HttpPost("refresh-token")]
26     [ProducesResponseType(typeof(RefreshTokenResponse), StatusCodes.Status200OK)]
27     [ProducesResponseType(typeof(ErrorDetails), StatusCodes.Status400BadRequest)]
28     [ProducesResponseType(typeof(ErrorDetails), StatusCodes.Status500InternalServerError)]
29     public override async Task<ActionResult> HandleAsync( [FromBody] RefreshTokenRequest request, CancellationToken cancellationToken = default ) {
30         var token = new JwtSecurityTokenHandler().ReadJwtToken(request.Token);
31         var userName = token?.Claims.FirstOrDefault(c => c.Type == JwtRegisteredClaimNames.UniqueName)?.Value ?? string.Empty;
32         var normalizedUserName = userName.ToUpper();
33
34         var loggedUser = _userManager.Users
35             .FirstOrDefault(u =>
36                 u.NormalizedEmail == normalizedUserName
37                 || u.NormalizedUserName == normalizedUserName
38             )
39             ?? throw WebApiException.UserNotFound(normalizedUserName);
40
41         var isValid = await _userManager.VerifyUserTokenAsync(loggedUser, TokenOptions.DefaultProvider, "RefreshToken", request.RefreshToken);
42         if ( !isValid )
43             throw WebApiException.BadRequest("Токен недействителен");
44
45         var expiresToken = DateTime.UtcNow.AddHours(1);
46         var response = new RefreshTokenResponse {
47             Token = await _tokenClaimsService.GetTokenAsync(loggedUser.UserName, expiresToken),
48             ExpiresToken = expiresToken,
49             RefreshToken = await _tokenClaimsService.GetRefreshTokenAsync(loggedUser.UserName)
50         };
51         return Ok(response);
52     }
53 }

```

Рисунок 5 – Код метода сброса токена взаимодействия

Для вызова данного метода, необходимо отправить POST запрос по адресу «/api/OAuth/refresh-token». В тело запроса необходимо передать токен взаимодействия и токен сброса (полученные при успешной аутентификации), если токены действительны, то в ответ будет сформирован новый токен взаимодействия со сроком «жизни», а также токен сброса. Пример взаимодействия приведен в п.3.4.3.

Также, не менее важная функция – это получение информации авторизованного пользователя. Код такой функции представлен на рисунке 6.

```

14 [Authorize]
15 [Tags("Пользователь")]
16 [Route("api/account")]
17 public class MyInfo: EndpointBaseAsync.WithoutRequest.WithActionResult {
18     private readonly IReadRepository<User> _userRepository;
19
20     public MyInfo( IReadRepository<User> userRepository ) {
21         _userRepository = userRepository;
22     }
23
24     [HttpPost("my-info")]
25     [ProducesResponseType(typeof(MyInfoResult), StatusCodes.Status200OK)]
26     [ProducesResponseType(typeof(ErrorDetails), StatusCodes.Status400BadRequest)]
27     [ProducesResponseType(typeof(ErrorDetails), StatusCodes.Status404NotFound)]
28     [ProducesResponseType(typeof(ErrorDetails), StatusCodes.Status500InternalServerError)]
29     public override async Task<ActionResult> HandleAsync( CancellationToken cancellationToken = default ) {
30         var userName = User.Identity?.Name ?? string.Empty;
31         var user = await _userRepository.SingleOrDefaultAsync(new UserByUserNameSpecification(userName), cancellationToken)
32             ?? throw WebApiException.UserNotFound(userName);
33
34         return Ok(new MyInfoResult(user) {
35             Roles = User.Claims.Where(c => c.Type.Equals(ClaimTypes.Role)).Select(c => c.Value),
36             Permissions = User.Claims.Where(c => c.Type.Equals(Authorization.ClaimConstants.PERMISSION)).Select(c => c.Value)
37         });
38     }
39 }

```

Рисунок 6 – Код метода получения данных пользователя

Для вызова данного метода, необходимо отправить POST запрос по адресу «/api/account/my-info», а в заголовке «Authorization» необходимо передать полученный ранее токен взаимодействия. Если пользователь успешно пройдет авторизацию, то в ответ получит свои данные. Пример взаимодействия продемонстрирован в п.3.4.3.

3.4.2 Описание методов взаимодействия

Взаимодействие с методами REST-сервиса выполняется по протоколу HTTP. Протокол HTTP лежит в основе обмена данными в Интернете. Кодировка для всех и запросов, и ответов – UTF-8. Все данные, кроме URI и двоичных файлов, передаются в формате JSON.

В запросах обязательно указывается заголовок «Ассерт» со значением «application/json», а в ответе с непустым телом обязательно наличие заголовка ответа «Content-Type» со значением «application/json».

Для взаимодействия с REST-сервисом используются следующие методы HTTP: GET, POST, PUT и DELETE.

Список методов:

- аутентификация пользователя – метод для получения токена взаимодействия;

– обновление токена взаимодействия – метод возвращает обновленный токен взаимодействия;

– получение личной информации – метод позволяет получить полную информацию о текущем пользователе;

– смена пароля – метод для установки нового пароля для текущего пользователя.

– сотрудники:

– получить запись – возвращает все данные сотрудника по его идентификатору;

– удалить запись – удаляет данные сотрудника по его идентификатору;

– создать запись – создает запись нового сотрудника и возвращает его данные;

– изменить запись – изменяет запись сотрудника по его идентификатору;

– получить список записей – метод позволяет получить информацию всех записей по сотрудникам;

– установить пароль – устанавливает новый пароль сотруднику по его идентификатору.

– роли:

– получить запись – возвращает все данные роли по его идентификатору;

– удалить запись – удаляет данные роли по его идентификатору;

– создать запись – создает запись новой роли и возвращает его данные;

– изменить запись – изменяет запись роли по его идентификатору;

– получить список записей – метод позволяет получить информацию всех записей по ролям.

– должности:

– получить запись – возвращает все данные должности по его идентификатору;

- удалить запись – удаляет данные должности по его идентификатору;
- создать запись – создает запись новой должности и возвращает его данные;
- изменить запись – изменяет запись должности по его идентификатору;
- получить список записей – метод позволяет получить информацию всех записей по должностям;
- виды груза:
 - получить запись – возвращает все данные вида груза по его идентификатору;
 - удалить запись – удаляет данные вида груза по его идентификатору;
 - создать запись – создает запись нового вида груза и возвращает его данные;
 - изменить запись – изменяет запись вида груза по его идентификатору;
 - получить список записей – метод позволяет получить информацию всех записей по видам груза.
- типы расхода:
 - получить запись – возвращает все данные типа расхода по его идентификатору;
 - удалить запись – удаляет данные типа расхода по его идентификатору;
 - создать запись – создает запись нового типа расхода и возвращает его данные;
 - изменить запись – изменяет запись типа расхода по его идентификатору.
 - получить список записей – метод позволяет получить информацию всех записей по типам расхода.
- статусы заказа:

- получить запись – возвращает все данные статуса заказа по его идентификатору;
- удалить запись – удаляет данные статуса заказа по его идентификатору;
- создать запись – создает запись нового статуса заказа и возвращает его данные;
- изменить запись – изменяет запись статуса заказ по его идентификатору;
- получить список записей – метод позволяет получить информацию всех записей по статусам заказ.
- клиенты:
 - получить запись – возвращает все данные клиента по его идентификатору;
 - удалить запись – удаляет данные клиента по его идентификатору;
 - создать запись – создает запись клиента и возвращает его данные;
 - изменить запись – изменяет запись клиента по его идентификатору;
 - получить список записей – метод позволяет получить информацию всех записей по клиентам.
- автомобили:
 - получить запись – возвращает все данные автомобиля по его идентификатору;
 - удалить запись – удаляет данные автомобиля по его идентификатору;
 - создать запись – создает запись автомобиля и возвращает его данные;
 - изменить запись – изменяет запись автомобиля по его идентификатору;
 - получить список записей – метод позволяет получить информацию всех записей по автомобилям.
- акты о расходах:

- получить запись – возвращает все данные актов о расходах по его идентификатору;
- удалить запись – удаляет данные акта о расходах по его идентификатору;
- создать запись – создает запись акта о расходах и возвращает его данные;
- изменить запись – изменяет запись акта о расходах по его идентификатору;
- получить список записей – метод позволяет получить информацию всех записей по актам о расходах.
- счета по заказам:
 - получить запись – возвращает все данные счета заказа по его идентификатору;
 - удалить запись – удаляет данные счета заказа по его идентификатору;
 - создать запись – создает запись счета заказа и возвращает его данные;
 - изменить запись – изменяет запись счета заказа по его идентификатору;
 - получить список записей – метод позволяет получить информацию всех записей по счетам по заказам.
- путевые листы:
 - получить запись – возвращает все данные путевого листа по его идентификатору;
 - удалить запись – удаляет данные путевого листа по его идентификатору;
 - создать запись – создает запись путевого листа и возвращает его данные;
 - изменить запись – изменяет запись путевого листа по его идентификатору;

- получить список записей – метод позволяет получить информацию всех записей по путевым листам.
- путевые листы:
 - получить запись – возвращает все данные путевого листа по его идентификатору;
 - удалить запись – удаляет данные путевого листа по его идентификатору;
 - создать запись – создает запись путевого листа и возвращает его данные;
 - изменить запись – изменяет запись путевого листа по его идентификатору;
 - получить список записей – метод позволяет получить информацию всех записей по путевым листам.

Полное взаимодействие всех методов приведено в приложении Г.

3.4.3 Тестирование основных функций

Для тестирования взаимодействия с веб-сервисом было взято ПО Postman. Это инструмент для работы с API, который позволяет тестировщику посылать запросы к сервисам и работать с их ответами. С его помощью можно протестировать бэкэнд (серверную часть) и убедиться, что он корректно работает.

Инструментов с аналогичным функционалом существует много. Выбор пал именно на Postman, поскольку он самый популярный. Но у него есть и другие преимущества:

- интуитивно-понятен и простой в использовании, не требует какой-то сложной настройки или знания языков программирования;
- бесплатный;
- поддерживает разные API (REST, SOAP, GraphQL);
- расширяется под любые нужды с помощью Postman API;

Как видно из рисунка выше, при успешной аутентификации сервер возвращает токен взаимодействия, срок его жизни и токен обновления. Пример возможных ошибок приведен на рисунке 8.

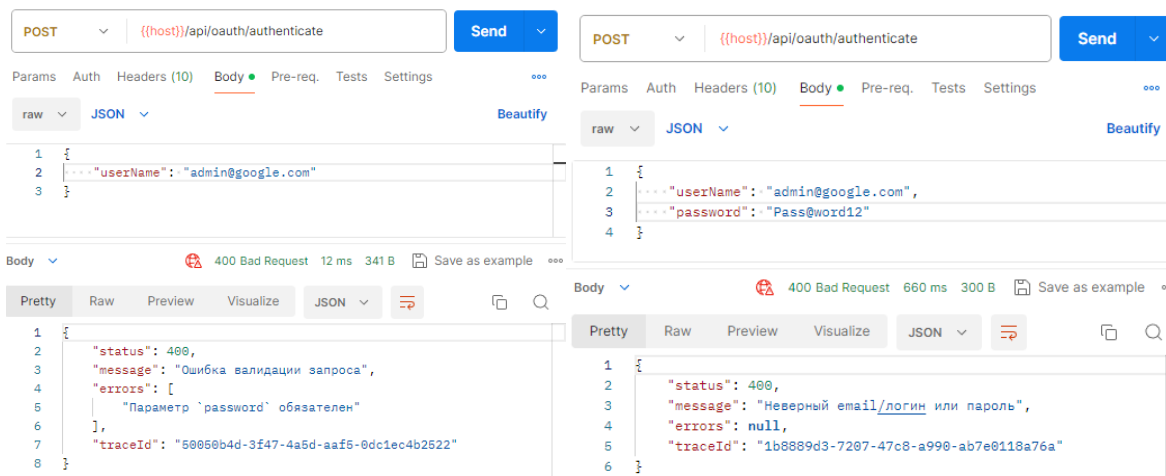


Рисунок 8 – Пример ошибок при аутентификации в системе

Для авторизации в системе используется токен взаимодействия, который был получен в прошлом шаге. Но данный токен имеет срок жизни и, в связи с этим, чтобы оставаться «в сети», необходимо вызвать метод для обновления токена. Пример взаимодействия по обновлению токена приведен на рисунке 9.

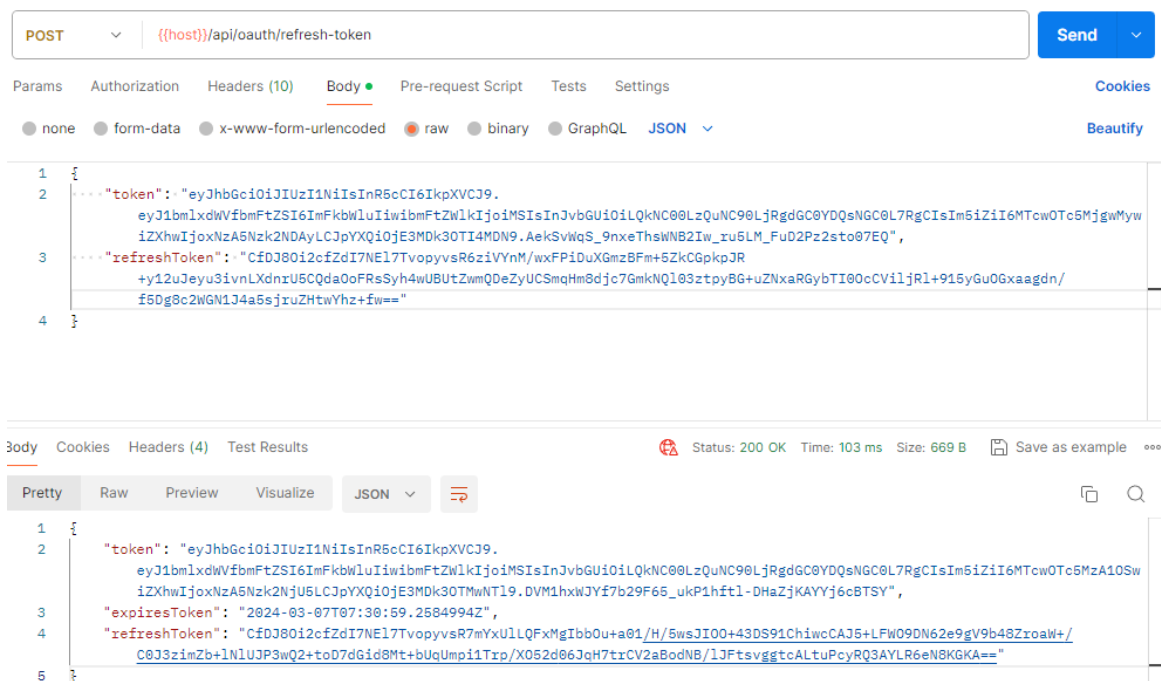


Рисунок 9 – Обновление токена

Как видно, на рисунке 9, при успешном обновлении токена сервер возвращает обновленный токен взаимодействия с новым сроком жизни и новый токен обновления. Пример возможных ошибок приведен на рисунке 10.

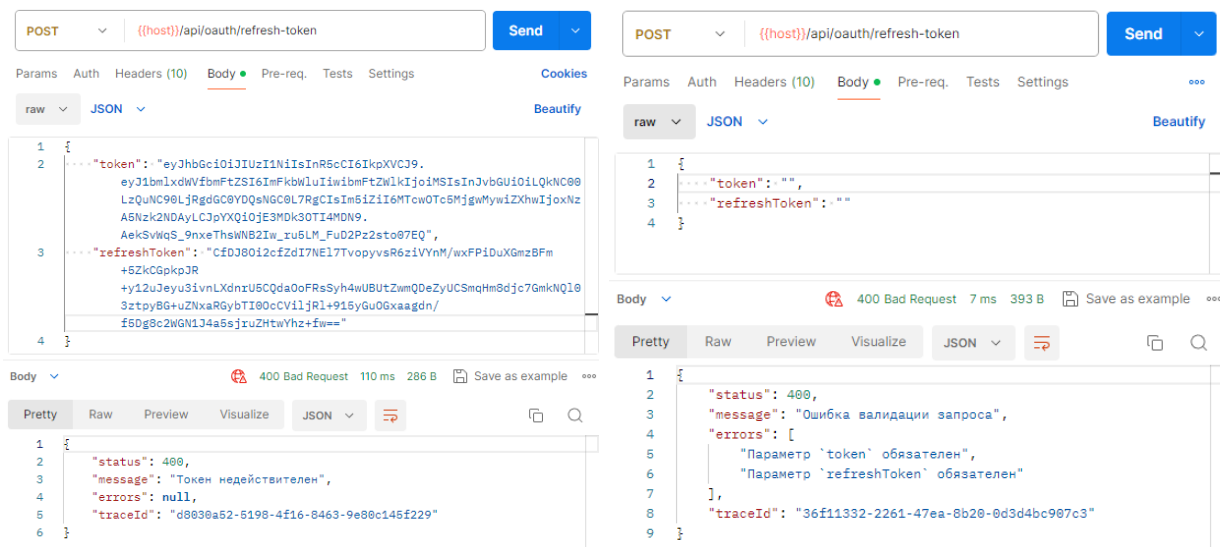


Рисунок 10 – Пример ошибок при попытке обновить токен

После успешного получения токена взаимодействия, становятся доступны разрешенные методы в системе. Один из которых – получение информации о текущем пользователе, данный метод доступен всем авторизованным в системе пользователям (т.е. всем, кто прошел успешно аутентификацию и получил токен взаимодействия). Пример получения информации авторизованного пользователя в системе и возможные ошибки представлены на рисунках 11 и 12 соответственно.

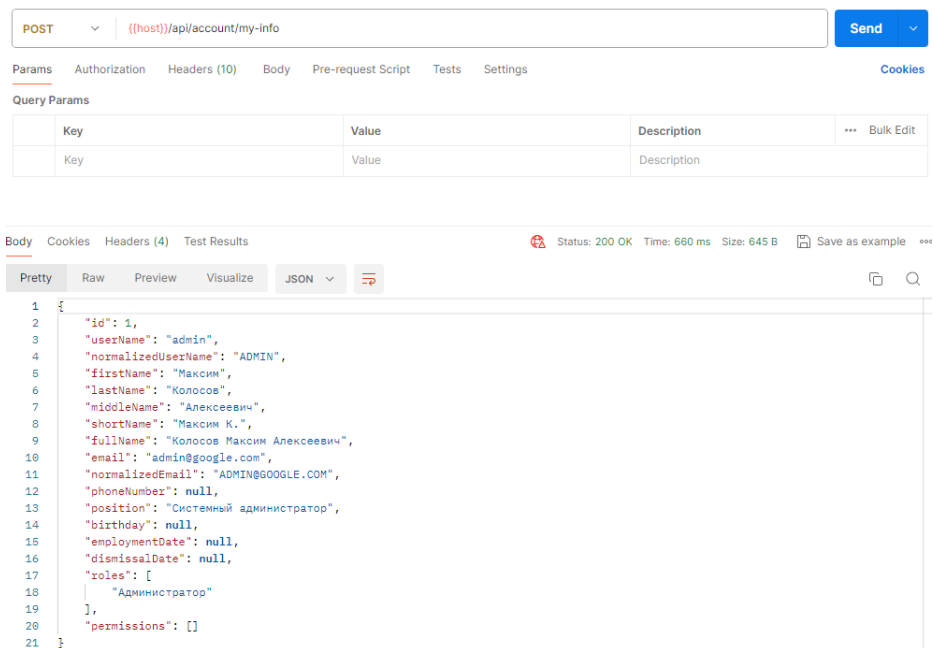


Рисунок 11 – Получение информации авторизованного пользователя

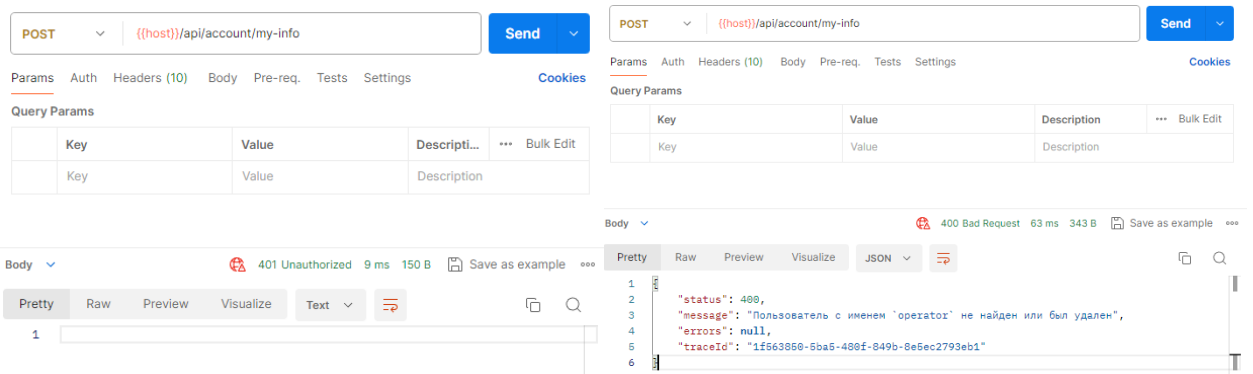


Рисунок 12 – Пример ошибок при получении данных пользователя

Для работы с данными (добавление, изменение, удаление и получение списка) используются идентичные методы, единственное что отличаются в адресе и параметрах в запросе и ответе. Поэтому рассмотрим только на примере метода для работы с «клиентами».

Пример успешного получения списка всех клиентов и возможных ошибок продемонстрированы на рисунках 13 и 14 соответственно.

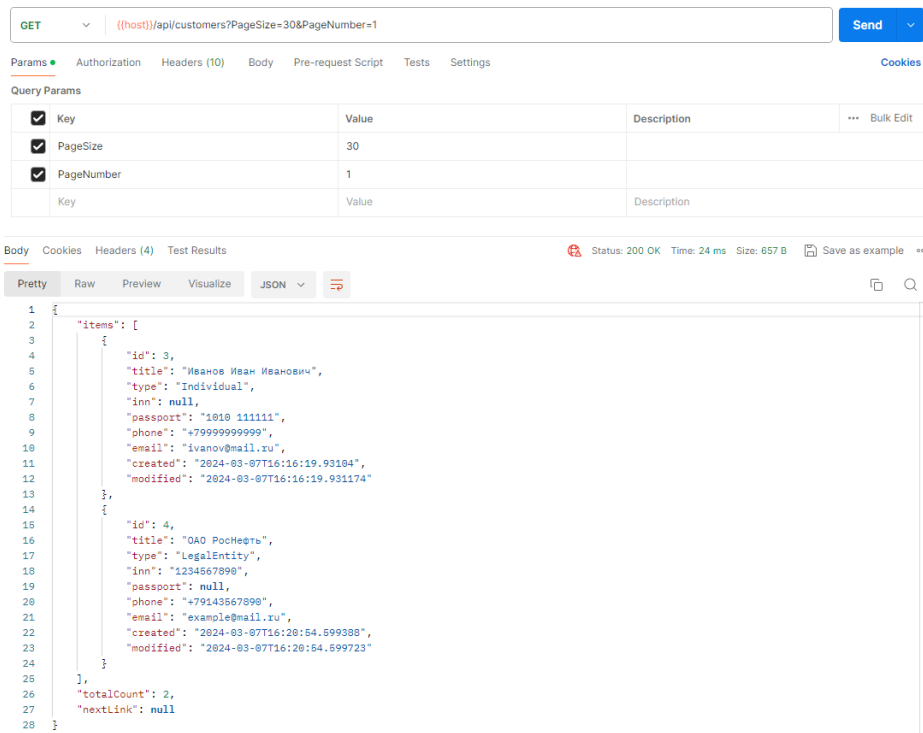


Рисунок 13 – Успешное получение списка клиентов

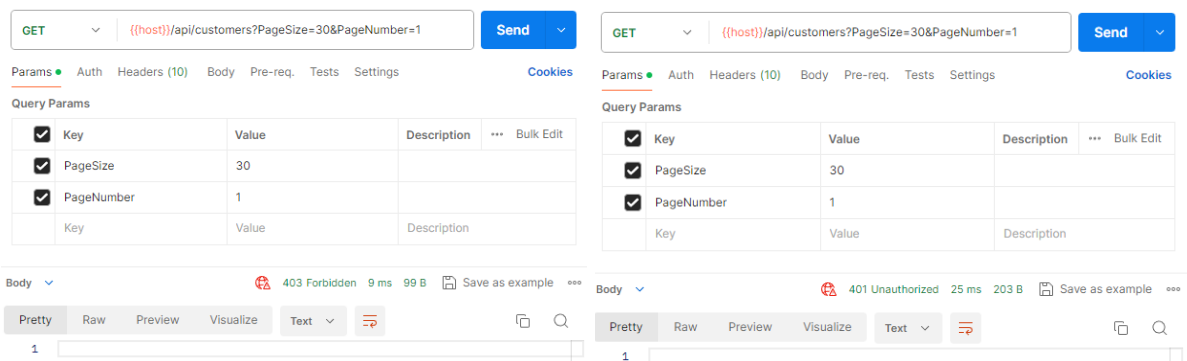


Рисунок 14 – Возможные ошибки при получении списка клиентов

Также необходимо получать не только все записи, но и данные по одной конкретной записи, по указанному уникальному идентификатору. Пример получения данных по клиенту и ошибки показаны на рисунках 15 и 16 соответственно.

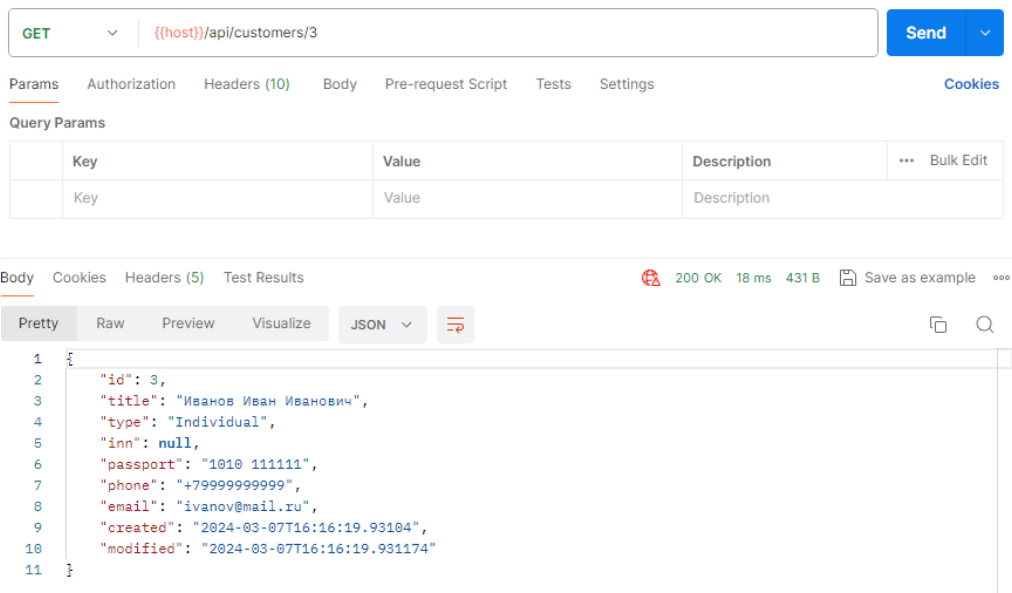


Рисунок 15 – Успешное получение данных клиента

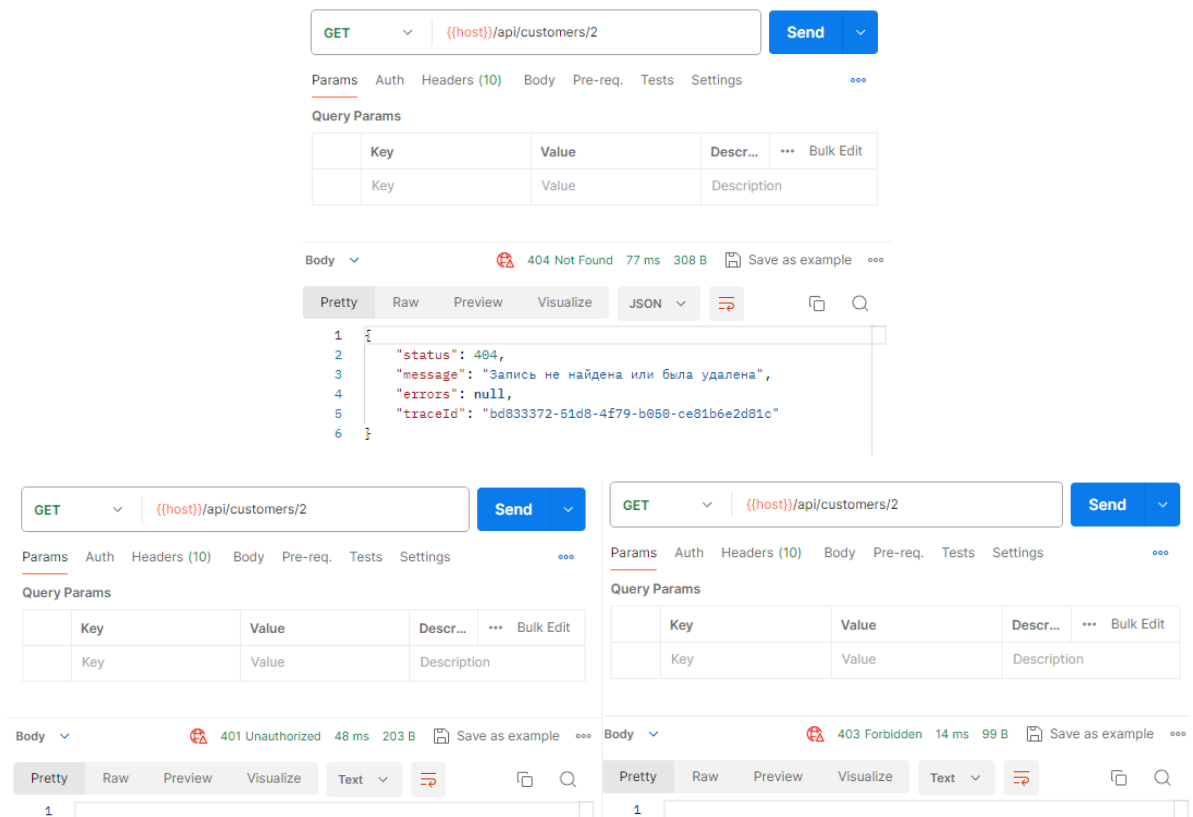


Рисунок 16 – Ошибки при получении данных клиента

Необходимо и создавать записи, успешное выполнение добавление записи и возможные ошибки продемонстрированы на рисунках 17 и 18 соответственно.

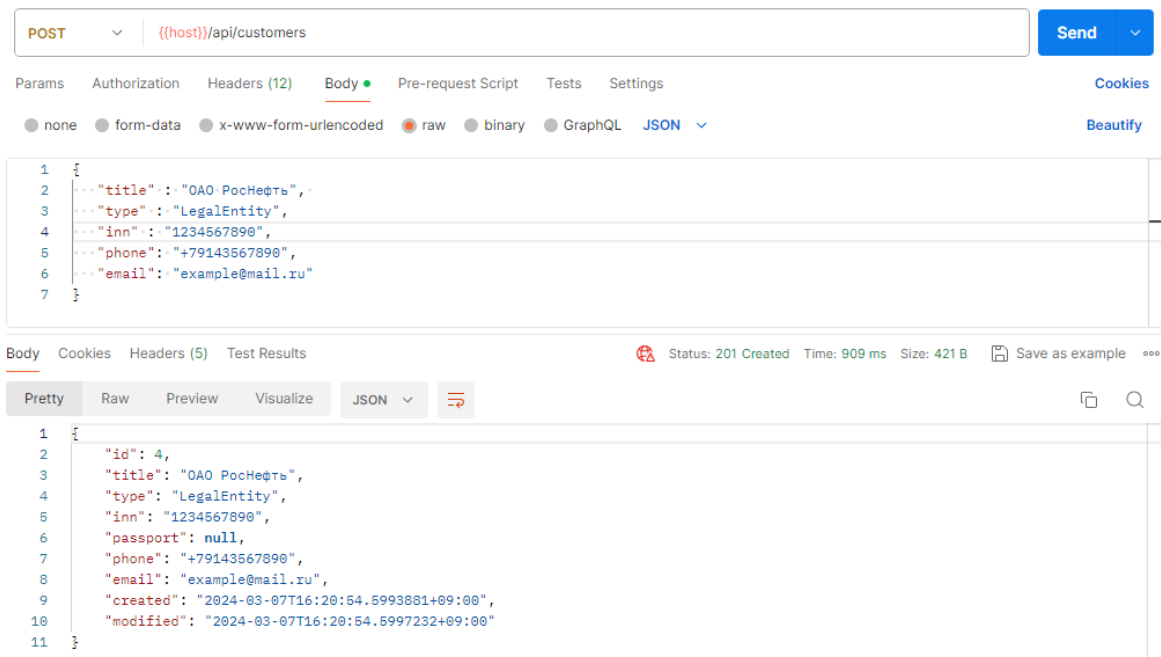


Рисунок 17 – Успешное создание данных о клиенте

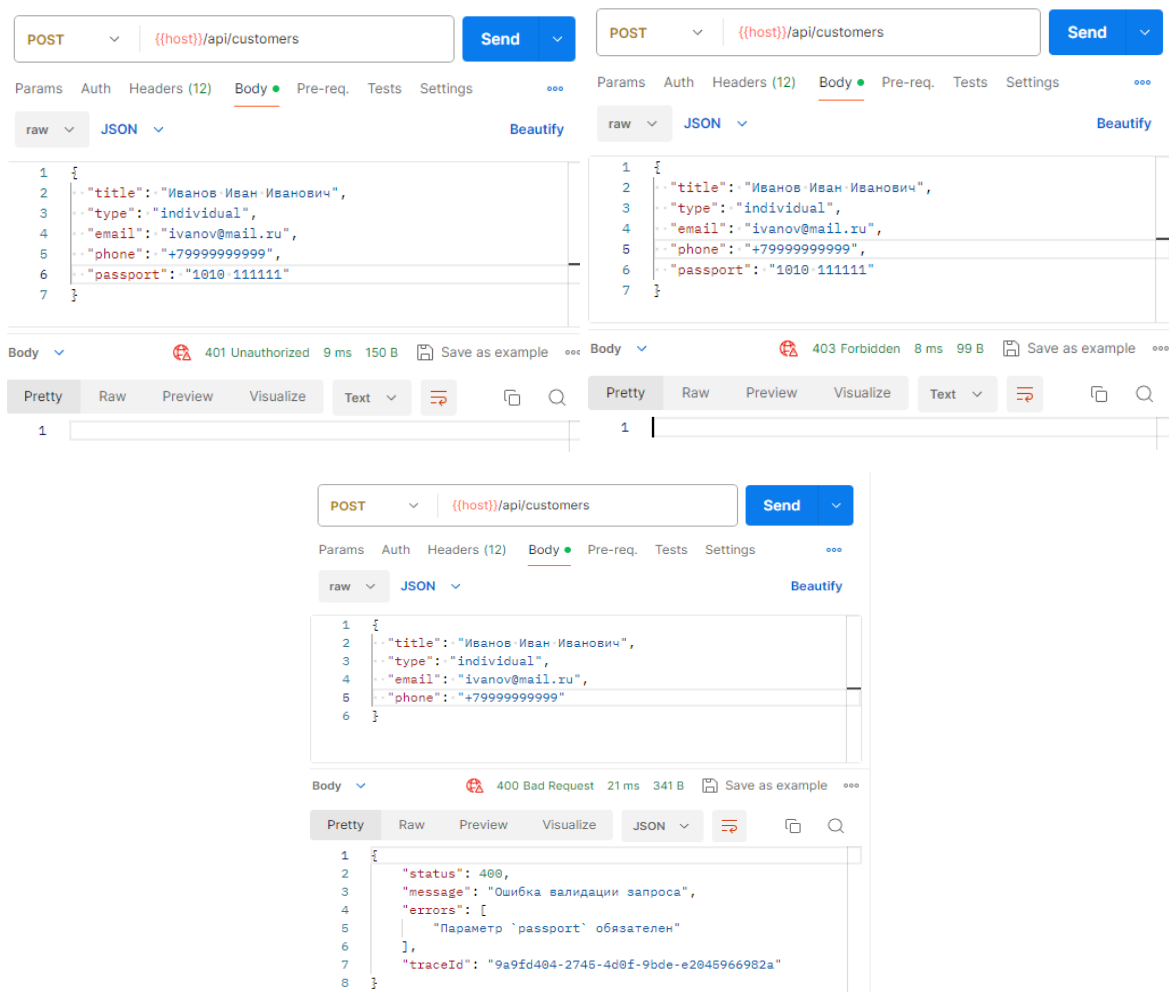


Рисунок 18 – Возможные ошибки при создании записи

Любую созданную запись необходимо редактировать, для этого используется метод для редактирования существующей записи. Пример успешного изменения данных по клиенту и возможные ошибки приведены на рисунках 19 и 20 соответственно.

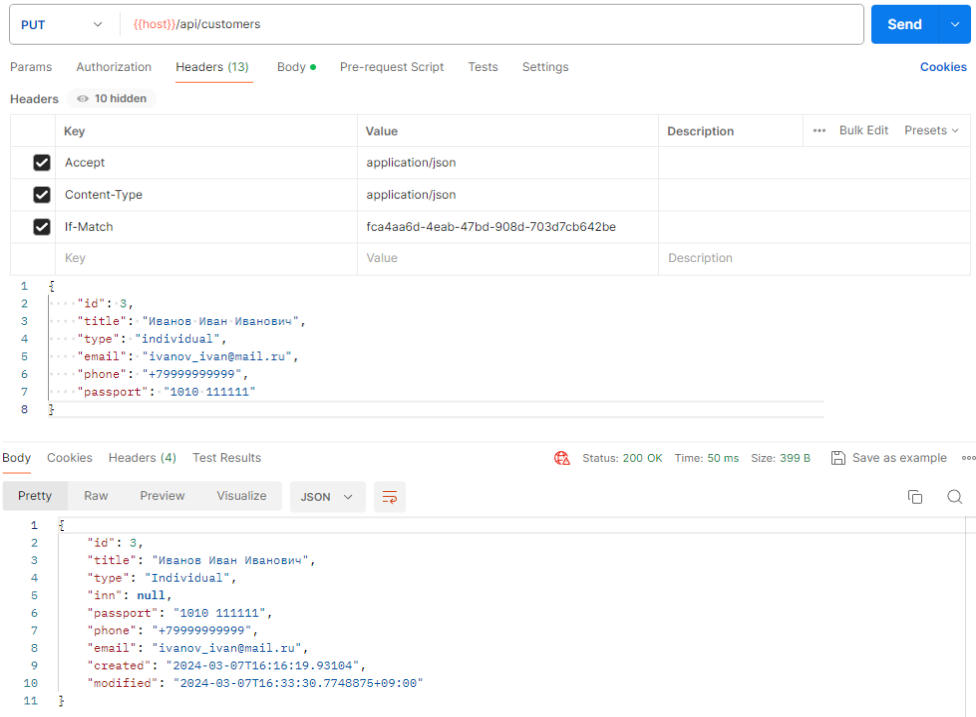


Рисунок 19 – Успешное изменение данных о клиенте

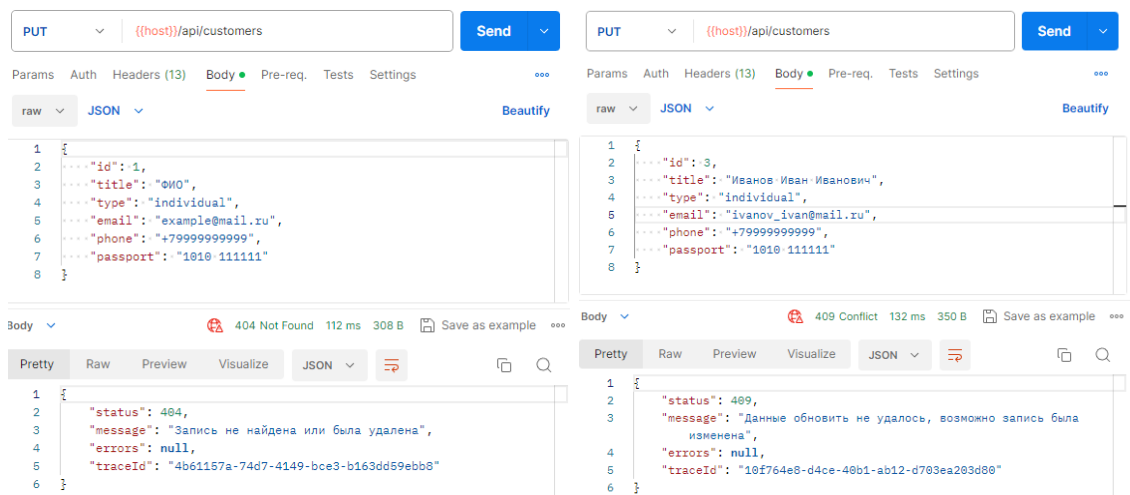


Рисунок 20 – Возможные ошибки при изменении записи

И последнее действие над данными, это удаление. Успешное выполнение удаления записи клиента и возможные ошибки показаны на рисунках 21 и 22 соответственно.

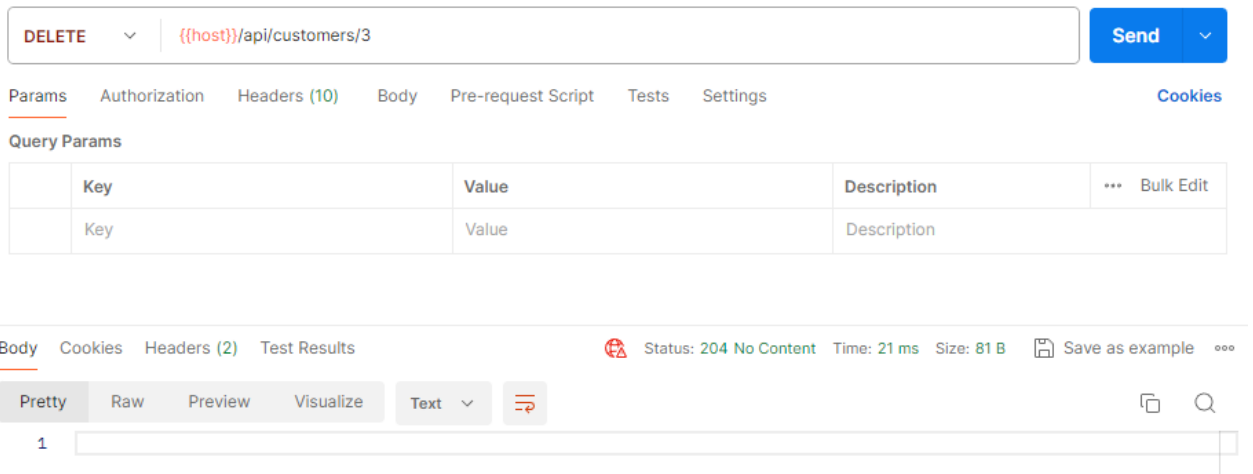


Рисунок 21 – Успешное удаление данных клиента

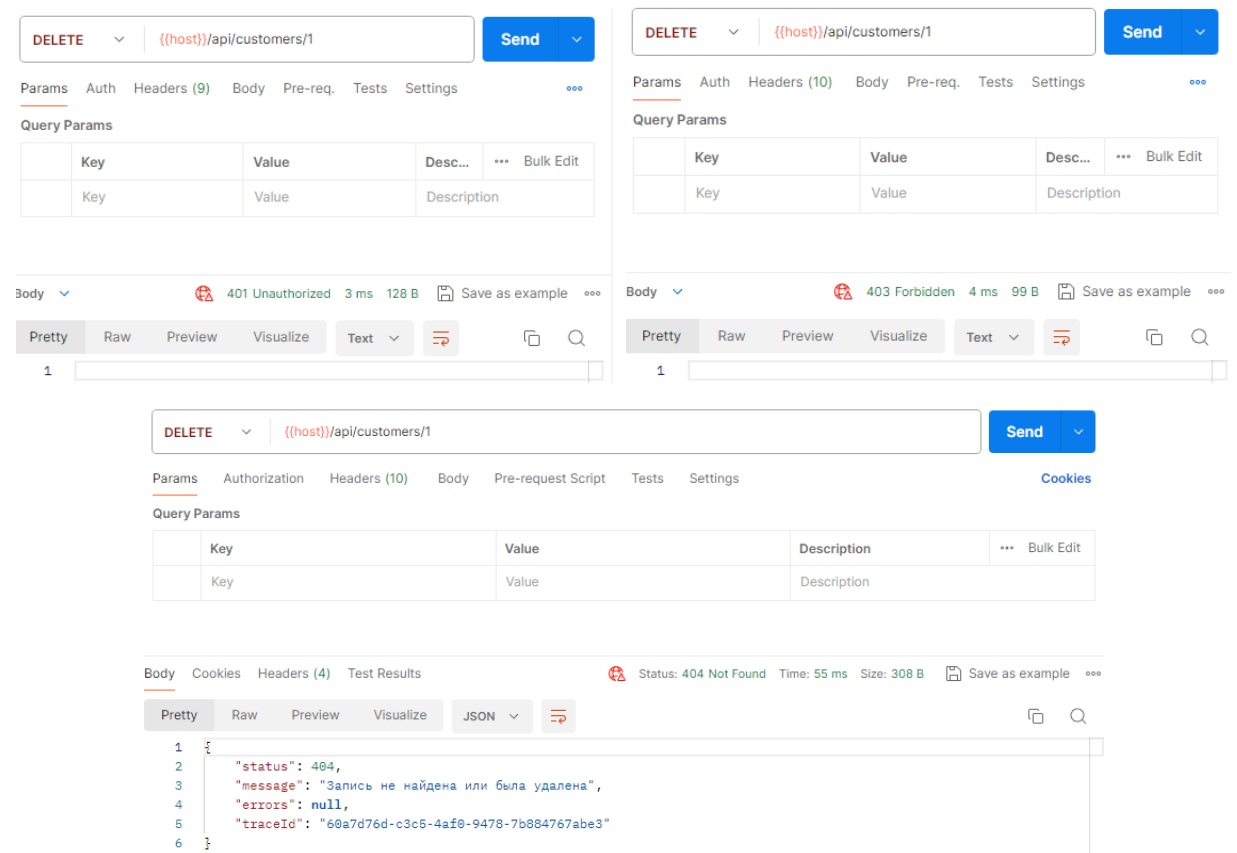


Рисунок 22 – Возможные ошибки при попытке удалить запись

3.5 Разработка клиентского программного обеспечения

3.5.1 Описание взаимодействия

С клиентским ПО могут взаимодействовать пользователи со следующими ролями доступа: администраторы, операторы и водители. Для каждой из ролей будут доступны уникальные доступы на просмотр/редактирование и

удаление данных, а также кнопки взаимодействия. Перечень основных окон взаимодействия.

Окно авторизации. При первом запуске приложения, пользователя «встречает» окно авторизации в системе, где выводится форма для заполнения логина/почты, пароля и кнопка для входа.

Окно с заказами (главное окно). После успешной аутентификации в системе, будет открыто главное окно. Для пользователя с ролью «Администратор» будут выведены все заказы без исключений с возможностью создания, просмотра и изменение статуса заказа, а также дополнительные пункты меню: данные, профиль и настройки.

Для роли «Оператор» будут выведены все незавершенные заказы (т.е. статус заказа не «Выполнен» и не «Отменен»), где указан в качестве оператора данный пользователь с возможностью создания, просмотра и смены статуса заказа. И доступность кнопок переходов в профиль и настройки.

Для «Водитель» будут выведены все заказы со статусом «Доставляется» и в качестве водителя указан текущий пользователь, с возможностью перехода к карточке заказа, а также кнопки перехода к профилю и настройкам.

Также для ролей «Администратор» и «Оператор» доступны кнопки взаимодействия для формирования отчетов с последующим сохранением excel файла на ПК.

Окно карточки заказа. В данном окне отображается вся информация по заказу, с вкладками переключениями: «Заказ», «Счета», «Расходы» и «История», а также кнопки взаимодействия «Изменить», «Сменить статус», «Добавить счет», «Добавить расход» и «Завершить доставку».

Кнопки «Изменить», «Изменить статус» и «Добавить счет» доступны для администратора и оператора. Действие «Добавить расход» доступно для администратора и водителя. Оставшиеся кнопки действия «Добавить расход» и «Завершить доставку» доступны для водителей.

На вкладке «Заказ» (данная вкладка открывается по умолчанию при переходе в карточку) выводится информация по заказу, клиенту и путевому листу. При переключении на вкладку «Счета» отобразится список всех сформированных счетов по данному заказу со своими кнопками управления счетами: «Подтвердить оплату», «Отменить оплату» и «Удалить». На вкладке «Расходы» выводится список расходов, которые были сформированы в период доставки заказа с возможностью удаления выбранной записи. При переходе на вкладку «История» будет выведен список смены статусов по данному заказу.

Окно профиля. Данное окно идентично для любой из ролей, здесь выводится личная информация авторизованного пользователя в системе, форма для смены пароля и кнопки взаимодействия «Справка» и «Выход». При нажатии на кнопку «Выход», выполнится удаление сессии в системе и будет перенаправление на окно авторизации.

3.5.2 Интерфейс пользователя

Окно авторизации. Данное окно выглядит идентично как для ПК, так и для телефонной версии. Внешний вид представлен на рисунке 23.

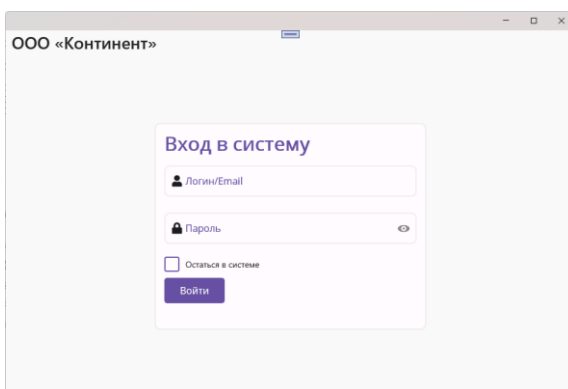


Рисунок 23 – Внешний вид окна авторизации

Окно с заказами. Интерфейс окна для пользователя с ролью «Администратор» продемонстрирован на рисунке 24, для роли «Оператор» на рисунке 25 и для «Водитель» на рисунке 26.

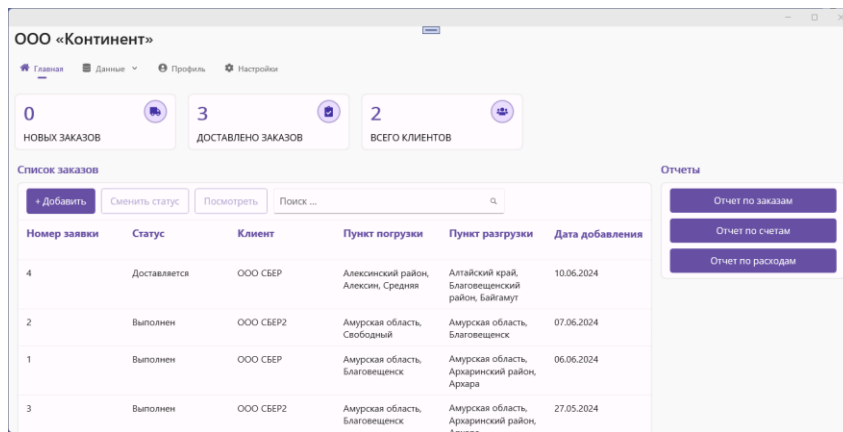


Рисунок 24 – Внешний вид главного окна программы для администратора

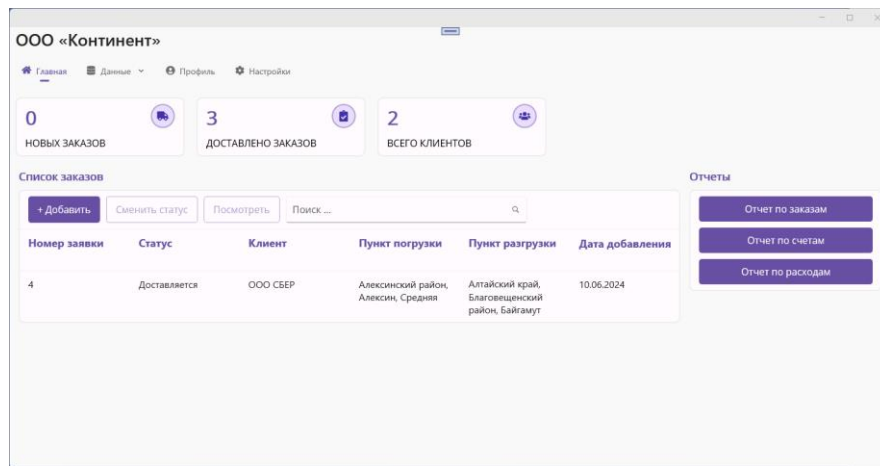
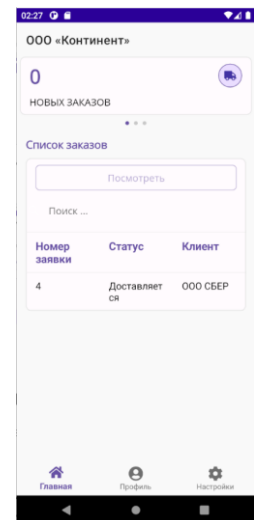
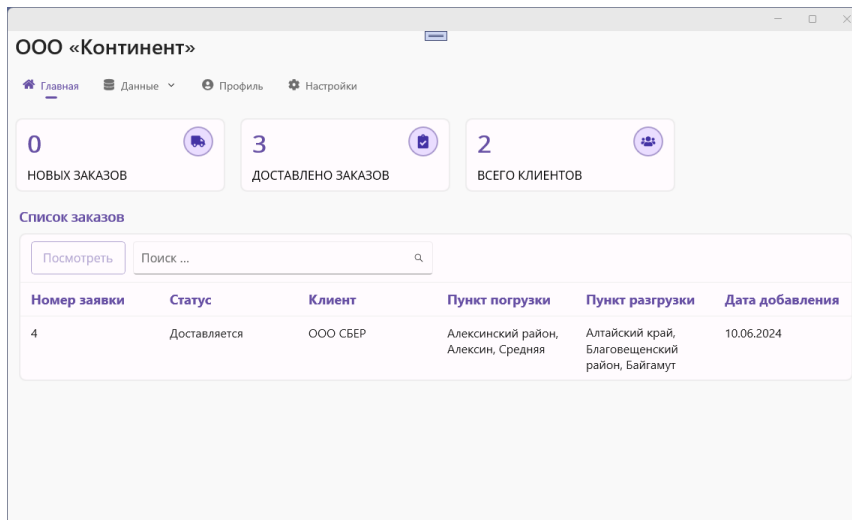


Рисунок 25 – Внешний вид главного окна программы для оператора



а)

б)

Рисунок 26 – Внешний вид главного окна программы для водителя
а) ПК версия; б) мобильная версия

Окно профиля. Внешний вид данного окна продемонстрирован на рисунке 27.

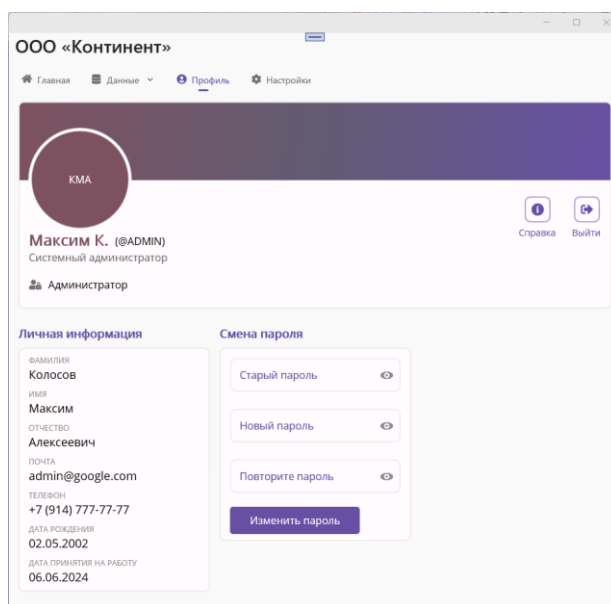


Рисунок 27 – Внешний вид окна профиля

Окно карточки заказа. Интерфейс окна для пользователя с ролью «Администратор» и «Оператор» продемонстрирован на рисунке 28 и для «Водитель» на рисунке 29.

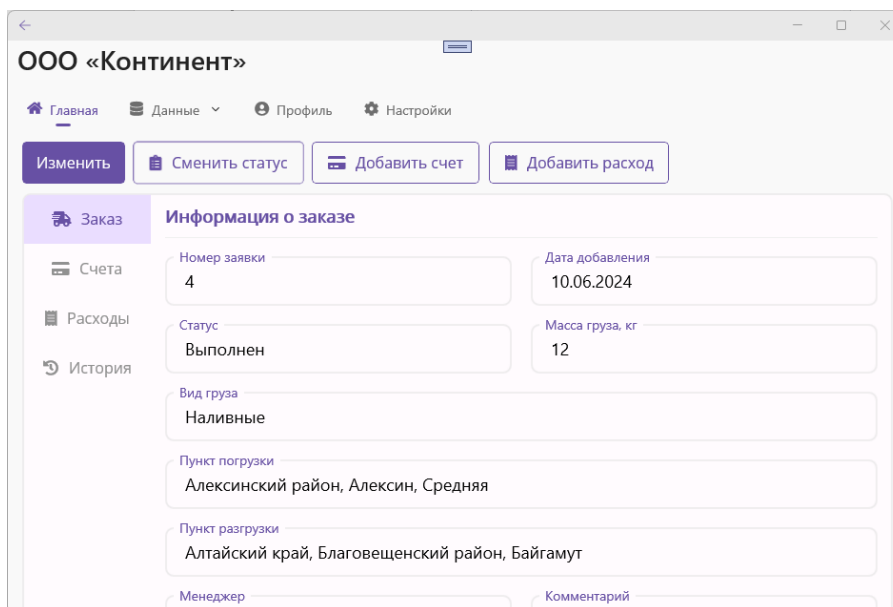
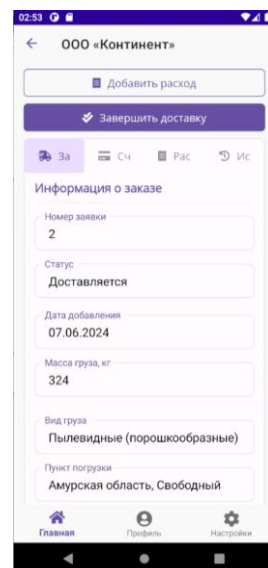
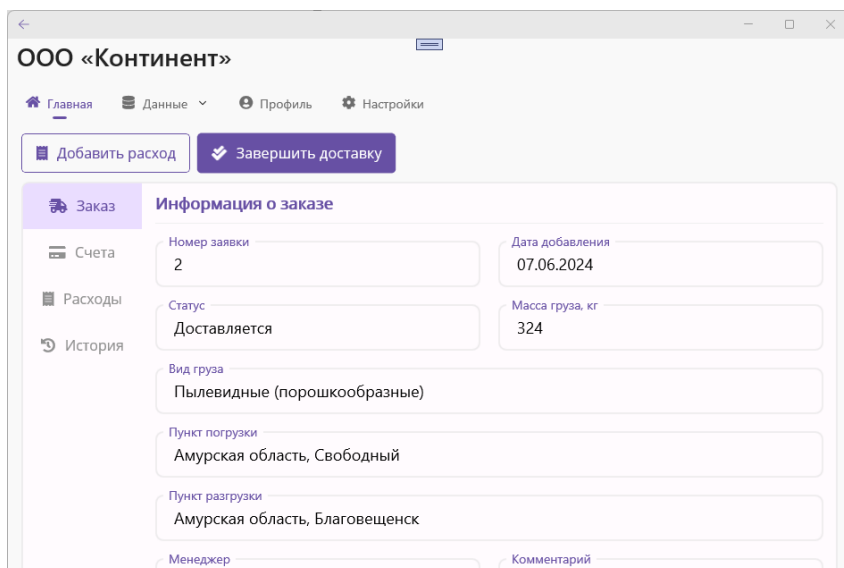


Рисунок 28 – Внешний вид карточки заказа



а)

б)

Рисунок 29 – Внешний вид карточки заказа
а) ПК версия; б) мобильная версия

Внешний вид окна для активных вкладок «Счета» и «Расходы» продемонстрированы на рисунках 30 и 31 соответственно.

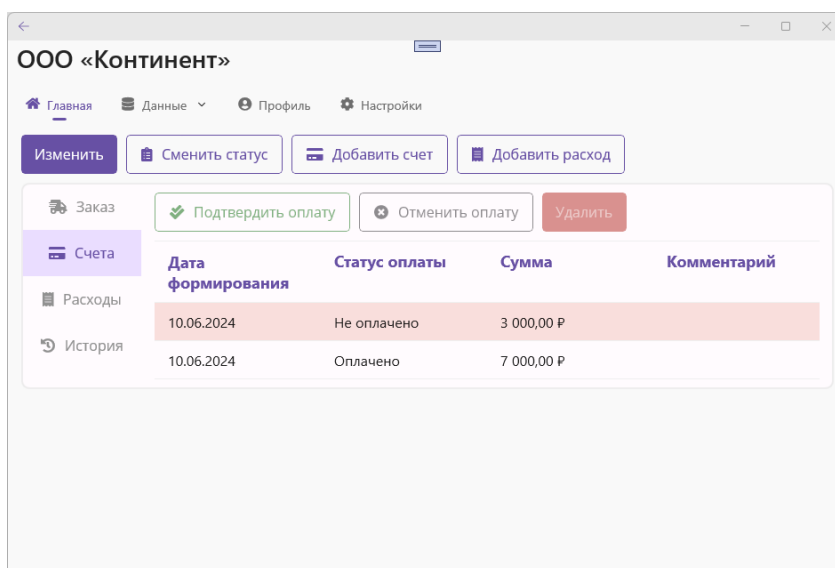


Рисунок 30 – Внешний вид карточки заказа (активная панель «Счета»)

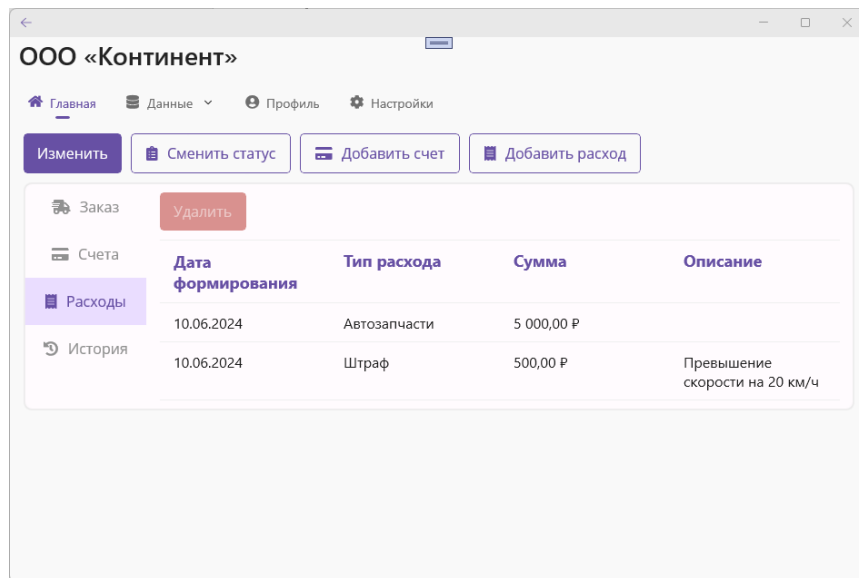


Рисунок 31 – Внешний вид карточки заказа (активная панель «Расходы»)

Интерфейс внешнего вида при активной вкладке «История» показана на рисунке 32. На данной вкладке выводится история «перемещения» заказа по статусам.

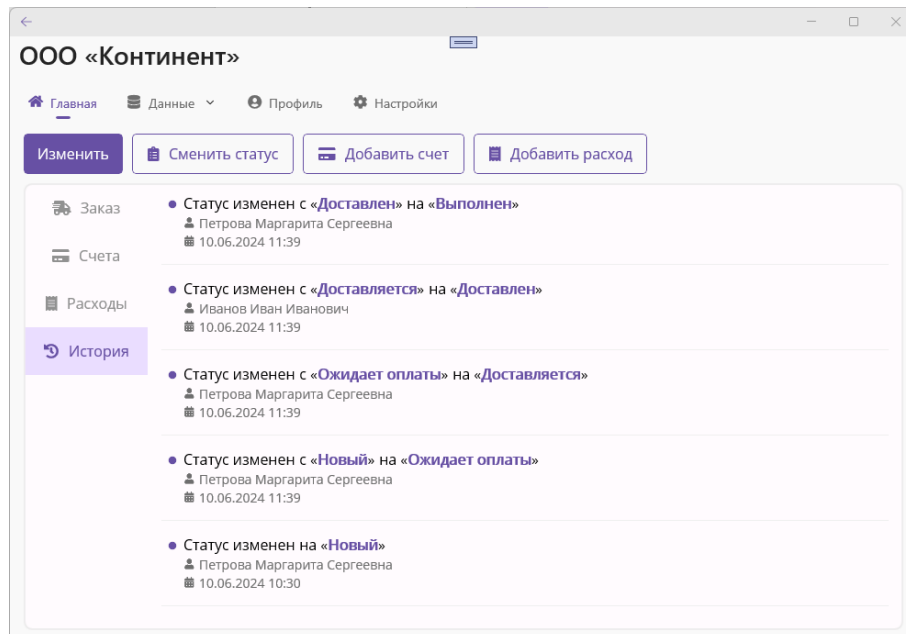


Рисунок 32 – Внешний вид карточки заказа (активная панель «История»)

3.5.2 Тестирование взаимодействия

При первом запуске ПО попадаем на окно авторизации, протестируем форму аутентификации в системе. Для этого указываем неверные данные и пытаемся выполнить вход в систему. Получили ошибку, показанную на рисунке 33.

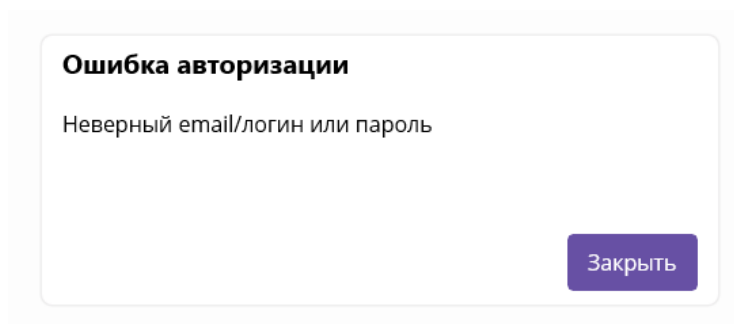


Рисунок 33 – Ошибка авторизации в системе

После успешной авторизации в системе для роли «Администратор» и «Оператор» попадаем на главное окно, со списком заказов. Выполним добавление нового заказа в систему. Для этого нажимаем на активную кнопку «Добавить» и открывается форма с полями для заполнения, продемонстрированным на рисунке 34.

Рисунок 34 – Форма добавления нового заказа для администратора

Для пользователя с ролью «Оператор» поля «Дата добавления» и «Менеджер» доступны только на чтение, рисунок 35.

Добавление заказа

Дата добавления
10.06.2024

Статус заказа
Новый

Клиент

Тип груза

Вес груза, кг

Пункт отправки

Пункт доставки

Менеджер
Петрова Маргарита Сергеевна

Комментарий

Сохранить Закреть

Рисунок 35 – Форма добавления нового заказа для оператора

При попытке сохранить запись с указанными невалидными данными, то будут отображены конкретные ошибки под каждым таким полем, представленным на рисунке 36.

Добавление заказа

Дата добавления
10.06.2024

Статус заказа
Новый

Клиент

Поле обязательно к заполнению

Тип груза

Поле обязательно к заполнению

Вес груза, кг

Поле обязательно к заполнению

Пункт отправки

Поле обязательно к заполнению

Пункт доставки

Поле обязательно к заполнению

Менеджер
Колосов Максим Алексеевич

Комментарий

Сохранить Закреть

Рисунок 36 – Ошибка добавления заказа

После успешного добавления заказа, будет выведено соответствующее сообщение, рисунок 37, и выполнится перенаправление на окно карточки заказа.

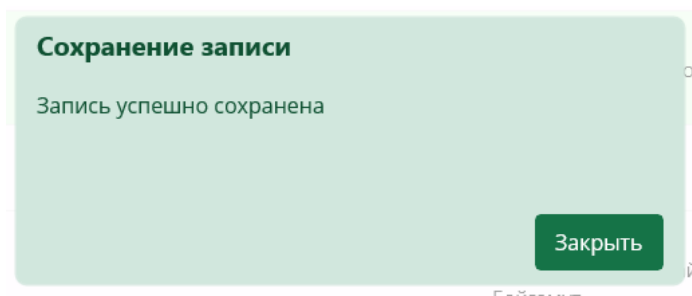


Рисунок 37 – Сообщение об успешном добавлении заказа

Тестирование работы кнопок взаимодействия в окне карточки заказа. Проверим кнопки, доступные для администратора и оператора: «Изменить», «Сменить статус» и «Добавить счет».

При взаимодействии с кнопкой «Изменить» откроется форма, представленной на рисунке 38, изменения текущего заказа, а именно заполнение информации по заказу и путевого листа.

A form titled "Изменение заказа" (Order Change). It has two columns of input fields. The first column contains: "Дата добавления" (Date added) with value "10.06.2024", "Клиент" (Client) with value "ООО СБЕР", "Водитель" (Driver) with value "Иванов Иван Иванович", "Пункт отправки" (Point of origin) with value "Белгородская область, Корочанский район, Коломыцево", "Расстояние, км" (Distance, km) with value "2312", and "Комментарий" (Comment). The second column contains: "Тип груза" (Type of cargo) with value "Наливные", "Вес груза, кг" (Weight of cargo, kg) with value "1234", "Менеджер" (Manager) with value "Петрова Маргарита Сергеевна", "Пункт доставки" (Point of delivery) with value "Алтайский край, Туюнецевский район, Урывка", and "Автомобиль" (Vehicle) with value "Volvo FH 460". At the bottom right, there are two buttons: "Сохранить" (Save) and "Закрыть" (Close).

Рисунок 38 – Форма изменения заказа для администратора

Для пользователя с ролью «Оператор», также поля «Дата добавления» и «Менеджер» доступны только на чтение, рисунок 39.

Изменение заказа

Дата добавления: 10.06.2024

Тип груза: Наливные

Клиент: ООО СБЕР

Вес груза, кг: 1234

Водитель: Иванов Иван Иванович

Менеджер: Петрова Маргарита Сергеевна

Пункт отправки: Белгородская область, Корочанский район, Коломыцево

Пункт доставки: Алтайский край, Тюменцевский район, Урывка

Расстояние, км: 2312

Автомобиль: Volvo FH 460

Комментарий

Сохранить Закрыть

Рисунок 39 – Форма изменения заказа для оператора

При нажатии на кнопку «Добавить счет» откроется форма на создание счета для данного заказа, рисунок 40.

Создание записи

Дата формирования: 10.06.2024

Статус оплаты: Не оплачено

Сумма, Р

Комментарий

Сохранить Закрыть

Рисунок 40 – Форма добавления счета для заказа

После успешного добавление счета появится соответствующее сообщение о том, что запись была успешно создана и на вкладке «Счета» отобразится в списке данная запись.

Проверим кнопку «Добавить расход», доступную для администратора, оператора и водителя. После нажатие на кнопку отобразится форма добавления нового расхода для текущего заказа, представленной на рисунке 41.

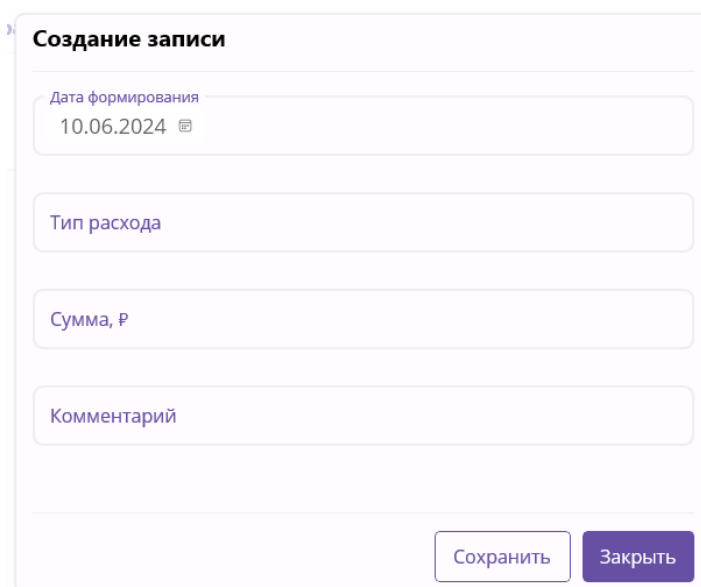


Рисунок 41 – Форма добавления расхода для заказа

После успешного добавление расхода появится соответствующее сообщение о том, что запись была успешно создана и на вкладке «Расходы» отобразится в списке данная запись, с последующей возможностью удаления.

Протестируем доступное действие «Завершить доставку» для пользователей с ролью «Водитель». После нажатия на кнопку появится предупреждение, рисунок 42, которое можно принять или отменить.

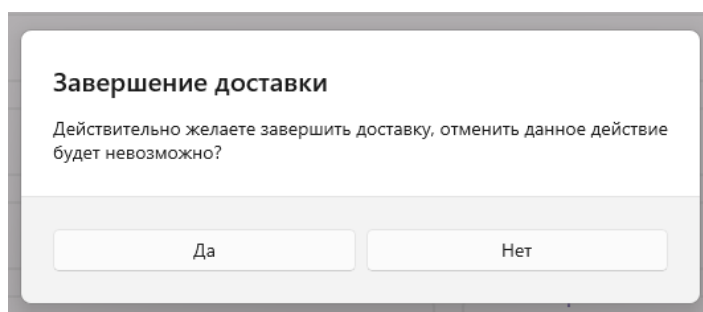


Рисунок 42 – Модальное окно подтверждения завершения доставки

Если нажать «Да», то статус заказа изменится на «Доставлено» и появится соответствующее сообщение о том, что статус был успешно изменен, рисунок 43.

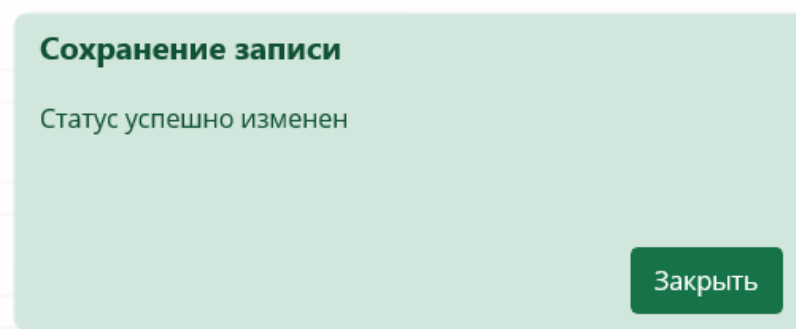


Рисунок 43 – Успешное выполнение завершения доставки

Тестирование взаимодействия в окне профиля, на данной странице имеется форма смены пароля, рисунок 44, куда необходимо указать корректный текущий пароль и задать новый пароль, после чего возможна смена пароля.

Рисунок 44 – Форма смены пароля

Если указать некорректный текущий пароль, то отобразится соответствующее диалоговое сообщение, продемонстрированное на рисунке 45.

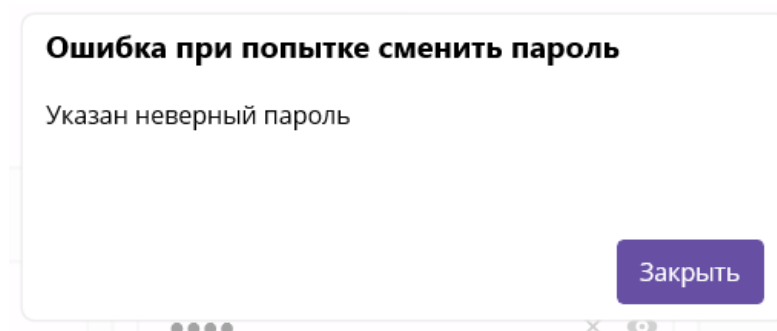


Рисунок 45 – Ошибка при попытке установить новый пароль

Если указать корректно текущий пароль и задать новый пароль, соответствующий требованиям политике безопасности пароля, то отобразится сообщение об успешной смене пароля, рисунок 46.

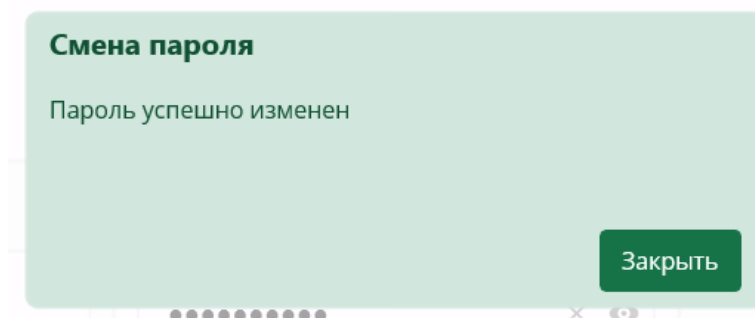


Рисунок 46 – Успешная смена пароля

Кнопка выхода из системы «Выход», после нажатия на данную кнопку появится диалоговое окно для подтверждения, во избежания случайного нажатия, рисунок 47.

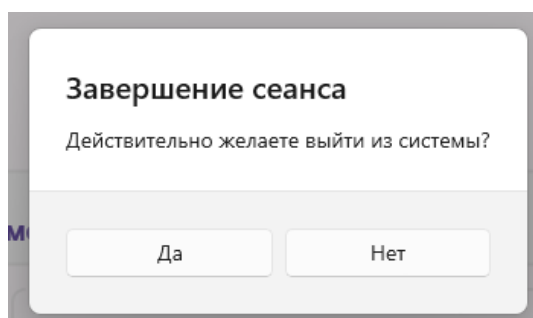


Рисунок 47 – Диалоговое окно подтверждения выхода из системы

Тестирование формирования отчетов, доступны на формирование три вида отчетов: «Отчет по заказам», «Отчет по счетам» и «Отчет по расходам». Формировать данные отчеты могут как администраторы, так и водители. При нажатии на любую из кнопок появится диалоговое окно с выбором границ даты для формирования данных в отчете, рисунок 48.

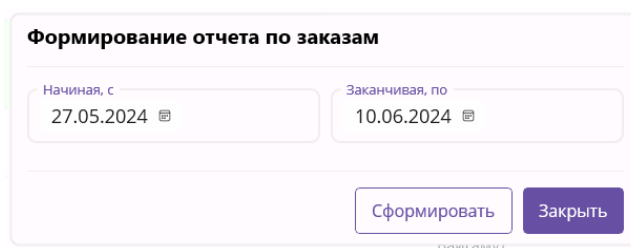


Рисунок 48 – Диалоговое окно выбора дат формирования отчетов

После нажатия на кнопку сформировать, запустится процесс формирования отчета, которое занимает определенное количество времени, и необходимо подождать, после того как отчет будет сформирован, появится диалоговое окно для сохранения файла, представленное на рисунке 49.

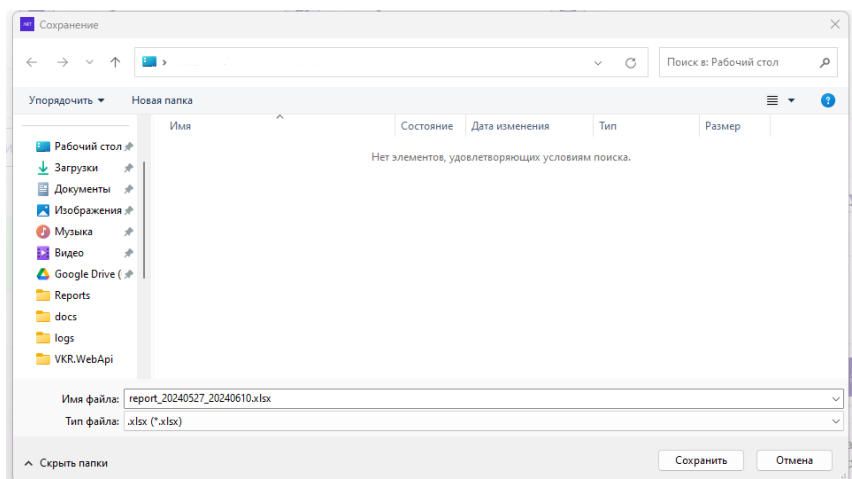


Рисунок 49 – Диалоговое окно сохранения файла

4 ЗАЩИТА ДАННЫХ И НАДЕЖНОСТЬ ИНФОРМАЦИОННОЙ СИСТЕМЫ

4.1 Организация защиты данных

Организация защиты данных в информационной системе (ИС) является важной частью разработки ИС.

Организация защиты данных в информационной системе может быть достигнута путем принятия следующих мер:

- установка и настройка антивирусного программного обеспечения: антивирусное ПО должно быть установлено и настроено на компьютерах, серверах и сетевых устройствах, чтобы обеспечить защиту от вирусов, троянов, шпионских программ и других вредоносных программ;

- установка и настройка фаервола: фаерволы помогают блокировать несанкционированный доступ к ИС из внешней сети. Фаерволы должны быть установлены и настроены на всех устройствах, связанных с ИС;

- резервное копирование данных: важно регулярно создавать резервные копии данных, чтобы восстановить их в случае их потери или повреждения. Резервные копии данных должны храниться в безопасном месте и должны быть защищены от несанкционированного доступа;

- обновление ОС и ПО: для реализации этого пункта нужно обновлять операционную систему, приложения и другое программное обеспечение, используемое в ИС. Это поможет устранить уязвимости, которые могут использоваться злоумышленниками для атаки на ИС;

- контроль доступа: нужно настроить контроль доступа для всех пользователей ИС. Убедиться, что каждый пользователь имеет только те права доступа, которые ему необходимы для выполнения своей работы;

- обучение пользователей: нужно регулярно проводить обучение пользователей ИС о безопасном использовании компьютеров и Интернета, чтобы они могли предотвратить атаки на ИС. Обучение пользователей должно включать

в себя информацию о сильных паролях, защите от вирусов и мошенничества в Интернете.

Выполнив все эти пункты, была достигнута полноценная защита данных в разрабатываемой ИС.

4.2 Надежность ИС

Надежность информационной системы (ИС) – это способность системы выполнять свои функции безотказно и в соответствии с требованиями пользователей в течение всего срока ее использования. Для обеспечения надежности ИС необходимо принимать меры по ее защите от различных угроз, таких как хакерские атаки, вирусы, ошибки человека и т.д.

Одной из важных мер для обеспечения надежности ИС необходимо обеспечивать надежность оборудования, установив качественные компоненты и применяя средства мониторинга и диагностики для раннего выявления проблем.

Кроме того, для обеспечения надежности входящего в информационную систему ПО, были проанализированы возможные аварийные ситуации. Было сделано все возможное, чтобы их предотвратить.

Исходя из выше сказанного, в целом, надежность данной ИС можно оценить на отлично.

5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

Полная работа с информационной системой и ее технической поддержкой подразумевает наличие рабочих мест, а это, в свою очередь, наличие помещений, в которых они находятся. Поэтому необходимо организовать эти места в соответствии с правилами и стандартами (СанПиН 2.2.4.3359-16 "Санитарно-эпидемиологические требования к физическим факторам на рабочих местах", утв. постановлением Главного государственного санитарного врача РФ от 21 июня 2016 г. № 81). и позаботиться о здоровье сотрудников при работе на ПК, разработав рекомендации.

Изучение и решение проблем, связанных с обеспечением здоровой и безопасной рабочей среды, является одной из наиболее важных задач в разработке новых технологий и производственных систем. Изучение и выявление возможных причин несчастных случаев на производстве, профессиональных заболеваний, несчастных случаев, взрывов и пожаров, а также разработка мер и требований по устранению этих причин обеспечивают безопасные и благоприятные условия для человеческого труда. Комфортные и безопасные условия труда являются одним из основных факторов, влияющих на производительность труда сотрудников, поддерживающих работу информационных систем. Работа сотрудников напрямую связана с компьютером и, следовательно, с вредным дополнительным воздействием целой группы факторов, которые значительно снижают производительность их работы.

5.1 Безопасность

Безопасность жизнедеятельности при работе с персональной электронно-вычислительной машиной (ПЭВМ) имеет особое значение для обеспечения защиты информации, сохранности данных и предотвращения нежелательных ситуаций. В данном разделе рассмотрим важные аспекты безопасности, связанные с работой на ПЭВМ.

5.1.1 Организация рабочего места

Базируясь на документе «Приказ МИНИСТЕРСТВО ТРУДА И СОЦИАЛЬНОЙ ЗАЩИТЫ РОССИЙСКОЙ ФЕДЕРАЦИИ от 29 октября 2021 года N 774н «Об утверждении общих требований к организации безопасного рабочего места»», можно предложить следующие требования и рекомендации по организации рабочего места для любого сотрудника, работающего в компании ООО "Континент".

Возможность смены рабочей позы. При организации рабочего места сотрудница должна быть обеспечена возможность смены рабочей позы. Это может быть особенно важно в случае длительной работы, когда сотруднику может потребоваться перемещаться или менять положение. Например, стол и стул должны быть выбраны таким образом, чтобы сотрудник мог комфортно переключаться между сидячим и стоячим положениями.

Во многих случаях, рабочая поза сотрудника в положении «сидя» будет более удобной, чем положение «стоя». Отсюда следует, что при организации рабочего места особое внимание следует уделить удобству сидячего места. Важно выбрать стул, который поддерживает правильное положение спины и обеспечивает комфорт в течение длительного времени.

Если возможно, рабочее место должно быть регулируемым. Это может включать в себя возможность регулировки высоты стола и стула, размеров рабочей поверхности, а также высоты и угла наклона подставки для ног. Регулируемое рабочее место может помочь в обеспечении комфорта и профилактике заболеваний опорно-двигательного аппарата, которые могут возникнуть из-за длительного сидения.

Если регулирование рабочего места невозможно, то можно использовать рабочее место с нерегулируемыми параметрами. В этом случае высота рабочей поверхности должна быть установлена в соответствии с государственными требованиями охраны труда. Это означает, что выбор стола, стула и другого

оборудования должен быть сделан в соответствии с соответствующими требованиями и стандартами.

Организация рабочего пространства на ПЭВМ должна предусматривать рабочий стол, способный оптимально разместить необходимое оборудование, учитывая его количество и конструктивные характеристики, например, размеры компьютера, клавиатуры и других элементов, а также специфику выполняемой работы. Важно использовать столы разных дизайнов, которые отвечают современным стандартам эргономики.

Стул или кресло для работы на ПЭВМ должно иметь конструкцию, которая позволяет поддерживать эффективную рабочую позу и предотвратить утомление, уменьшая статическое напряжение мышц шейно-плечевой зоны и спины. Выбор типа стула или кресла зависит от характера и продолжительности работы на ПЭВМ, а также от роста пользователя.

Рабочий стул или кресло должно быть оборудовано системой подъема и поворота, а также регулировкой высоты, угла наклона сиденья и спинки и расстояния между спинкой и передним краем сиденья. Регулировка каждого из этих параметров должна быть простой и надежно фиксироваться.

Сиденье, спинка и другие элементы стула или кресла должны быть полумягкими, иметь нескользящее, не накапливающее статическое электричество и воздухопроницаемое покрытие, которое легко очищается от загрязнений.

Видеомонитор должен быть расположен на расстоянии от 600 до 700 мм от глаз пользователя, но не ближе 500 мм, учитывая размеры алфавитно-цифровых символов на экране.

Помещения, в которых установлены ПЭВМ, требуют ежедневной влажной уборки. В этих помещениях должны быть аптечки первой помощи и углекислотные огнетушители.

Все вышеуказанные требования применимы к организации и оборудованию рабочих мест на ПЭВМ для учащихся средних и высших учебных заведений.

На рисунке 50 изображено рекомендуемое размещение пользователя ПЭВМ.

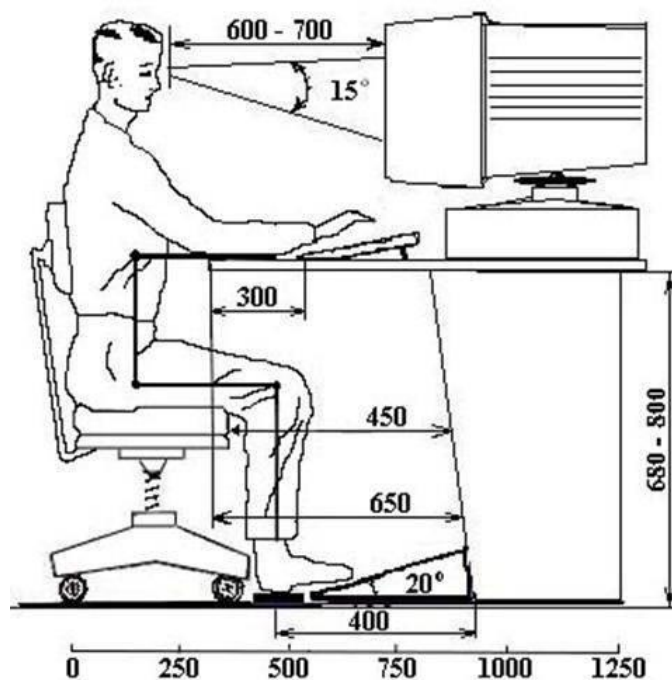


Рисунок 50 – рекомендуемое размещение пользователя ПЭВМ

5.1.2 Освещение

Освещение в кабинетах, где используется персональный компьютер, играет важную роль в обеспечении комфортных и безопасных условий труда. Не только для персонала, но и для клиентов, поскольку правильное освещение может способствовать расслаблению и улучшению настроения.

Согласно Постановлению 28 Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», освещенность на рабочем месте с ПЭВМ должна составлять 300-500 люкс. Это обеспечивает достаточное количество света для выполнения задач, связанных с использова-

нием ПЭВМ, и в то же время предотвращает излишнее напряжение глаз. Однако освещение должно быть регулируемым, чтобы можно было создать более уютную атмосферу при необходимости.

Освещение должно быть равномерным, чтобы избежать резких контрастов между освещенными и теневыми зонами.

Прямой и отраженный блеск на экране ПЭВМ следует избегать, так как это может вызвать блики и утомление глаз.

Естественное освещение предпочтительно, но, если это невозможно, следует использовать искусственное освещение. Искусственное освещение должно быть выполнено в виде комбинированного, то есть общее освещение дополняется местным.

Лампы с дневным светом обычно предпочтительнее, поскольку они меньше утомляют глаза и обеспечивают более естественное восприятие цветов. В кабинете это может помочь создать более приятную и расслабляющую атмосферу.

Важно регулярно проверять и обслуживать систему освещения, чтобы обеспечить ее надежную работу и долговечность. Это особенно важно в кабинете, где надежность и комфорт являются ключевыми факторами для обеспечения эффективного взаимодействия с клиентами.

В кабинете также следует учесть влияние цвета освещения. Исследования показывают, что теплые тона, такие как желтый или оранжевый, могут способствовать расслаблению и улучшению настроения, в то время как холодные тона, такие как синий или зеленый, могут способствовать концентрации и спокойствию.

Также стоит учесть, что освещение может влиять на восприятие пространства. Например, свет, направленный вверх, может создать ощущение более высокого потолка, в то время как свет, направленный вниз, может создать ощущение более уютного и закрытого пространства.

5.1.3 Микроклимат

Микроклимат в производственных помещениях представляет собой комплекс физических факторов, которые влияют на теплообмен человека и определяют его самочувствие, работоспособность, здоровье и производительность труда. Эти факторы включают температуру воздуха, относительную влажность воздуха, скорость движения воздуха и мощность теплового излучения.

Однако, помимо этих внешних факторов, источником повышенной температуры может быть и сама персональная электронно-вычислительная машина (ПЭВМ). Если температура в помещении превышает допустимые показатели, это может привести к снижению работоспособности пользователей, появлению сонливости и утомляемости, а в крайних случаях - даже к потере сознания или более тяжелым последствиям.

Поддержание микроклимата рабочего места в пределах гигиенических норм является важнейшей задачей охраны труда. Показатели микроклимата должны обеспечивать сохранение теплового баланса человека с окружающей средой и поддержание оптимального или допустимого теплового состояния организма.

Оптимальные микроклиматические условия установлены по критериям оптимального теплового и функционального состояния человека. Они обеспечивают общее и локальное ощущение теплового комфорта в течение 8-часовой рабочей смены при минимальном напряжении механизмов терморегуляции, не вызывают отклонений в состоянии здоровья, создают предпосылки для высокого уровня работоспособности и являются предпочтительными на рабочих местах.

В помещении должны соблюдаться оптимальные величины температуры воздуха 22-24 градуса, при относительной влажности воздуха (40-60) %. Для поддержания микроклимата в помещении используются системы вентиляции, которые представляют собой каналы для воздуха, постоянно выгоняющие воз-

дух и насыщающие помещение чистым и свежим воздухом. Если системы вентиляции недостаточно для требуемого уровня микроклимата, то применяется система кондиционирования или отопления.

Системы кондиционирования используются для поддержания постоянной температуры, влажности и очистки воздуха от вредных веществ. Они позволяют решить проблему с задержанием углекислого газа в помещении.

Поддержание микроклимата рабочего места в пределах гигиенических норм является важнейшей задачей охраны труда. Это обеспечивает сохранение теплового баланса человека с окружающей средой и поддержание оптимального или допустимого теплового состояния организма, что в свою очередь способствует высокому уровню работоспособности и общему здоровью работника.

5.1.4 Графический интерфейс пользователя

Представлено описание графического интерфейса пользователя (GUI) информационной системы учета грузоперевозок с учетом стандарта ГОСТ Р ИСО 9241-161-2016 «Эргономика взаимодействия человек-система. Часть 161. Элементы графического пользовательского интерфейса». Данный стандарт определяет принципы и рекомендации для разработки эргономичных элементов графического интерфейса, обеспечивающих удобство и эффективность взаимодействия пользователя с системой.

Интерфейс обеспечивает четкую структуру и логическую организацию элементов, что позволяет пользователям быстро ориентироваться и находить необходимую информацию. Размещение элементов интерфейса основано на принципе «от простого к сложному», что способствует удобству использования.

Интерфейс пользователя соответствует рекомендациям по визуальному оформлению, установленным стандартом. Это включает использование понятных и согласованных значков, кнопок и элементов управления, а также цвето-

вой схемы и типографики, обеспечивающих хорошую читаемость и восприятие информации. Визуальные элементы также согласованы с предпочтениями пользователей, чтобы создать приятный и комфортный интерфейс. Интерфейс профиля пользователя показан на рисунке 51.

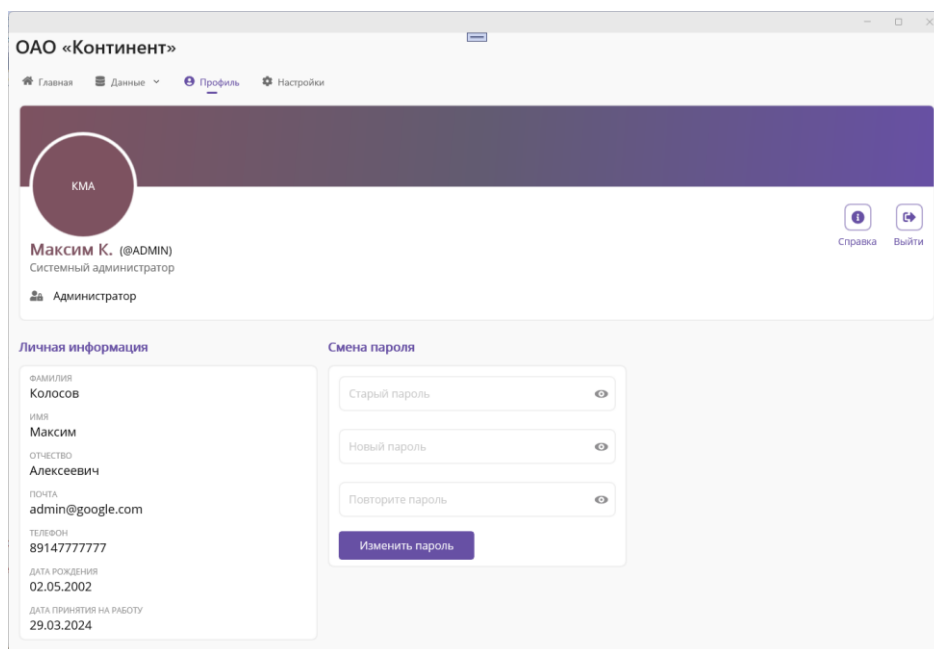


Рисунок 51 – Графический интерфейс пользователя

Интерфейс обладает интерактивностью и интуитивно понятной навигацией, что позволяет пользователям легко освоить систему и выполнять необходимые действия. Это включает использование понятных и информативных меток, контекстных подсказок, возможность отмены и повтора действий, а также удобную систему меню и ссылок для перехода между различными разделами и функциями.

Элементы интерфейса располагаются с учетом принципов стандарта, обеспечивая удобство работы пользователя. Размеры элементов, их отступы и выравнивание оптимизированы для удобного использования на компьютере.

5.2 Экологичность

Действующее законодательство РФ, включая Федеральный закон «Об отходах производства и потребления» от 24.06.1998 N 89-ФЗ, соблюдается в Компании ООО "Континент".

Списываемое офисное оборудование, такое как компьютеры и принтеры, подлежит специализированной утилизации. Техника содержит множество веществ, представляющих потенциальную опасность для окружающей среды. К таким веществам относятся тяжелые металлы (свинец, кадмий, ртуть), полихлорированные бифенилы (ПХБ), поливинилхлорид (ПВХ), присутствующие в некоторых моделях мониторов.

Кроме того, в современных устройствах содержатся также и ценные вещества, включая золото, серебро и палладий, которые могут быть извлечены и повторно использованы. Отсюда идет особенность обращения с офисной техникой - она не только представляет опасность, но и может стать источником ценных ресурсов.

При утилизации такого оборудования важно гарантировать, что все эти вещества обрабатываются должным образом. Такие процедуры требуют специализированного оборудования и квалифицированных специалистов, поэтому списанная оргтехника обычно передается лицензированным организациям, специализирующимся на утилизации электронного оборудования.

По истечении срока службы люминесцентные лампы также требуют особого обращения, поскольку они содержат ртуть. Согласно законодательству, они должны быть переданы в специальные пункты приема для безопасной утилизации, чтобы предотвратить загрязнение окружающей среды.

Когда офисная мебель становится непригодной для использования, с ней обращаются также в соответствии с законом. Некоторые элементы мебели могут быть переработаны или переданы в нужды хозяйственного отдела.

Бумажные отходы, которые неизбежно образуются в ходе работы, собираются отдельно и передаются на переработку. Это способствует не только снижению объема отходов, но и сохранению лесных ресурсов.

5.3 Чрезвычайные ситуации

5.3.1 Меры пожарной безопасности

В случае пожара в кабинете, следует придерживаться следующих инструкций:

- при обнаружении признаков пожара, в первую очередь, следует немедленно активировать систему оповещения о пожаре, если это еще не сделано. Это можно сделать, нажав на пожарную кнопку, обычно расположенную в коридоре рядом с кабинетом;

- сразу после активации сигнализации, необходимо сообщить в пункт охраны, который находится на 1 этаже при главном входе здания, о возникшей ситуации. Если возможно, следует описать место и характер пожара;

- если пожар не распространяется быстро и можно безопасно подойти к огнетушителю, его следует использовать для попытки потушить пожар;

- в любом случае, главный приоритет – это безопасность людей. Нужно немедленно начать эвакуацию. При этом следует помнить о правиле «не паниковать». Паника может вызвать хаос и привести к травмам;

- при планировании пути эвакуации, предпочтение следует отдавать ближайшему запасному выходу. Открывая дверь, нужно сначала приложить к ней руку. Если она горячая, это может означать, что за дверью уже идет пожар. В таком случае, нужно искать другой путь;

- в процессе эвакуации необходимо оставаться как можно ниже относительно пола, поскольку дым и теплые газы поднимаются вверх. Если дыма много, можно использовать влажную ткань для прикрытия дыхательных путей;

- после выхода из здания нужно убедиться, что все присутствующие в кабинете люди покинули здание. Если кто-то пропал из виду, об этом следует немедленно сообщить пожарным;

- после эвакуации ни в коем случае не следует возвращаться в здание, пока пожарные не подтвердят его безопасность;

- важно помнить, что предотвращение пожара – это лучший способ защиты. Поэтому необходимо соблюдать все правила пожарной безопасности и регулярно проверять состояние пожарной сигнализации и огнетушителей;

5.3.2 Рекомендованные действия при угрозе совершения террористического акта

Данная информация взята с официального сайта АмГУ в разделе «Анти-террористическая деятельность».

Цель данных рекомендаций - помочь гражданам правильно ориентироваться и действовать в экстремальных и чрезвычайных ситуациях, а также обеспечить создание условий, способствующих расследованию преступлений. Любой человек должен точно представлять свое поведение и действия в экстремальных ситуациях, психологически быть готовым к самозащите.

Всегда контролируйте ситуацию вокруг себя, особенно когда находитесь на объектах транспорта, культурно-развлекательных, спортивных и торговых центрах.

При обнаружении забытых вещей, не трогая их, сообщите об этом водителю, сотрудникам объекта, службы безопасности, органов милиции. Не пытайтесь заглянуть внутрь подозрительного пакета, коробки, иного предмета.

Не подбирайте бесхозных вещей, как бы привлекательно они не выглядели.

В них могут быть закамуфлированы взрывные устройства (в банках из-под пива, сотовых телефонах и т.п.). Не пинайте на улице предметы, лежащие на земле.

Если вдруг началась активизация сил безопасности и правоохранительных органов, не проявляйте любопытства, идите в другую сторону, но не бегом, чтобы Вас не приняли за противника.

При взрыве или начале стрельбы немедленно падайте на землю, лучше под прикрытие (бордюр, торговую палатку, машину и т.п.). Для большей безопасности накройте голову руками.

Случайно узнав о готовящемся теракте, немедленно сообщите об этом в правоохранительные органы.

В последнее время часто отмечаются случаи обнаружения гражданами подозрительных предметов, которые могут оказаться взрывными устройствами. Подобные предметы обнаруживают в транспорте, на лестничных площадках, около дверей квартир, в учреждениях и общественных местах. Как вести себя при их обнаружении? Какие действия предпринять?

Если обнаруженный предмет не должен, по вашему мнению, находиться в этом месте, не оставляйте этот факт без внимания.

Если вы обнаружили забытую или бесхозную вещь в общественном транспорте, опросите людей, находящихся рядом. Постарайтесь установить, чья она и кто ее мог оставить. Если хозяин не установлен, немедленно сообщите о находке водителю (машинисту).

Если вы обнаружили неизвестный предмет в подъезде своего дома, опросите соседей, возможно, он принадлежит им. Если владелец не установлен - немедленно сообщите о находке в ваше отделение милиции.

Если вы обнаружили неизвестный предмет в учреждении, немедленно сообщите о находке администрации или охране.

Во всех перечисленных случаях:

- не трогайте, не передвигайте, не вскрывайте обнаруженный предмет;
- зафиксируйте время обнаружения предмета;
- постарайтесь сделать все возможное, чтобы люди отошли как можно дальше от находки;
- обязательно дождитесь прибытия оперативно-следственной группы (помните, что вы являетесь очень важным очевидцем).

Помните: внешний вид предмета может скрывать его настоящее назначение. В качестве камуфляжа для взрывных устройств используются самые обычные бытовые предметы: сумки, пакеты, коробки, игрушки и т.п.

Не предпринимайте самостоятельно никаких действий с находками или подозрительными предметами, которые могут оказаться взрывными устройствами – это может привести к их взрыву, многочисленным жертвам и разрушениям.

ЗАКЛЮЧЕНИЕ

В ходе написания выпускной квалификационной работы были выполнены все поставленные в ней задачи. Была подробно исследована структура предприятия ООО «Континент» для понимания процессов документооборота в ней.

Целью данной работы являлось разработка информационной системы учета грузоперевозками.

Для выполнения цели, были реализованы все поставленные задачи: проанализирована предметная область, выполнены все этапы проектирования базы данных, разработана серверная часть для ИС, разработано клиентское программное обеспечение на ПК и телефон, организована защита данных и надежность ИС.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Васильев А. Н. Программирование на С# для начинающих / А. Н. Васильев. - М. : Изд-во ООО «Издательство «Эксмо»», 2023. - 586 с.

2 Гаврилюк В. И. Restful: системные принципы и применение / В. И. Гаврилюк. - М. : Изд-во ООО «Издательство Молодой ученый», 2020. - 46 с.

3 Информационные системы : моногр. / Е. В. Бурцева [и др.]. - Изд-во Тамб. Гос. Техн. Ун-та, 2020. - 121 с.

4 Моргунов Е. П. PostgreSQL. Основы языка SQL / Е. П. Моргунов. - Изд-во БХВ-Петербург, 2020. - 336 с.

5 Методы сервиса авторизации REST API [Электронный ресурс]. М.,2024. URL: <https://docs.wavesenterprise.com/ru/1.5.2/how-to-use/rest-api-auth-methods.html> (дата обращения: 14.01.2024).

6 Официальная документация PostgreSQL [Электронный ресурс]. М.,2024. URL: <https://www.postgresql.org/docs/>

7 Общий порядок действий по созданию систем защиты информации [Электронный ресурс]. М.,2024. URL: <https://trinosoft.com/pages/is/omop.php> (дата обращения: 20.01.2024).

8 Обзор протокола HTTP [Электронный ресурс]. М.,2024. URL: <https://developer.mozilla.org/ru/docs/Web/HTTP/Overview> (дата обращения: 16.01.2024).

9 Принципы построения REST JSON API / Хабр [Электронный ресурс]. М.,2024. URL: <https://habr.com/ru/articles/447322/> (дата обращения: 16.01.2024).

10 Примеры тестовых сценариев Postman [Электронный ресурс]. М.,2024. URL: <https://infostart.ru/1c/articles/1545930/> (дата обращения: 16.01.2024).

11 Руководство по REST архитектуре [Электронный ресурс]. М.,2024. URL: <https://bookflow.ru/rukovodstvo-po-rest-arhitekture/> (дата обращения: 15.01.2024).

12 Хориков В.А. Разработка информационных систем. Учебник / В. А. Хориков. - СПб.: БХВ-Петербург, 2019. - 384 с.

13 Visual Studio: IDE и редактор кода для разработчиков [Электронный ресурс]. М.,2024. URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 15.01.2024г).

ПРИЛОЖЕНИЕ А

Техническое задание

1 Введение

1.1 Наименование программы

«Разработка ИС учета грузоперевозок для компании ООО «Континент»

1.2 Краткая характеристика области применения программы

Информационная система разрабатывается с целью оптимизации использования ресурсов компании: эффективного отслеживания деятельности организации по грузоперевозкам, составления отчетов в логистический отдел компании, которые помогут эффективно распределять по необходимым направлениям (маршрутам) грузы и автотранспорт, в зависимости от сложности маршрута, назначать в рейс водителей с соответствующим опытом и навыками. Для повышения эффективности перевозок в целом и быстрого составления отчетов за выбранный период.

2 Основания для разработки

Основанием для разработки служит задание к выпускной квалификационной работе выданное, кафедрой ИиУС.

3 Назначение разработки

Программное обеспечение предназначено для автоматизации составления отчетов по перевозкам, снижение трудозатрат и количества необходимого персонала.

3.1 Функциональное назначение

Регистрация всех заказов компании по перевозке груза, исполнение заказов и ведение по ним отчетов.

Ограничение доступа к информационной системе с помощью идентификации пользователя (парольная защита) с системой ролей.

Оператор осуществляет ввод, удаление и редактирование той или иной информации, с возможностью автоматизации (подстановок из других таблиц,

Продолжение ПРИЛОЖЕНИЯ А Техническое задание

фиксированных наборов данных и т.д.) и входного контроля от некорректных действий.

Оператор осуществляет поиск по разнообразным признакам (не менее одного в каждой из информационных категорий, например, ФИО водителя и т.д.). С помощью функции поиска можно получить информацию об отдельной единице категории или совокупности объектов, занесенных в информационную систему.

Оператор осуществляет сортировку по различным признакам (не менее одного в каждой из информационных категорий, например, по характеристикам перевозимых грузов и т.д.).

Оператор, заполняя соответствующие поля для запроса в системе, имеет возможность быстро находить соответствующую информацию для формирования отчёта по основным логистическим вопросам компании.

3.2 Эксплуатационное назначение

Программное обеспечение должно эксплуатироваться как сотрудниками в компании, так и для самих водителей.

4 Требования к программе или программному изделию

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

Разрабатываемое программное обеспечение должно обеспечить:

- разграничение прав доступа;
- просмотр информации по заказам;
- просмотр информации по клиентам;
- формирование отчетов.

После запуска программы пользователю отображается форма ввода логина и пароля. В системе существует несколько ролей со своими уникальными

Продолжение ПРИЛОЖЕНИЯ А Техническое задание

правами доступа – администратор, оператор и водитель. Программа проверяет тип пользователя и открывает соответствующий интерфейс.

Для администратора доступны все подсистемы программы.

Оператору доступно добавление и изменение заказов, добавление новых клиентов в базу, а также вывод отчетов по заказам.

Для водителя будут доступны только личные заказы, для которых можно добавлять расходы и завершать доставку.

4.1.2 Требования к организации входных и выходных данных

Входные данные о заказах, пользователях, автомобилях, водителях и клиентах будут отправлены с форм клиентского приложения, заполненных оператором, и сохранены в реляционной базе данных (PostgreSQL). Программное обеспечение обеспечивает разграничение прав доступа к данным.

Выходные данные должны быть организованы в виде отчетов о записях клиентов, заказов (csv), а также отчетов по перевозкам pdf формата. Доступ к данной информации также разделены по ролям (доступно администраторам и операторам).

4.1.3 Требования к временным характеристикам

После изменения данных, находящихся в базе данных, новая информация на клиентах должна обновляться не позднее, чем через 1 минуту.

4.2 Требования к надежности

Вероятность безотказной работы системы должна составлять не менее 90 % при условии исправности сети (связи клиентских приложений с серверной частью).

4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы

Продолжение ПРИЛОЖЕНИЯ А Техническое задание

В связи с тем, что в базе данных хранятся данные о совершенных клиентами покупках (финансовая информация) — базу данных стоит резервировать (резервирование замещением).

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- организацией бесперебойного питания технических средств;
- использованием лицензионного программного обеспечения;
- регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
- регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

4.2.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 10 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

4.2.3 Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует

Продолжение ПРИЛОЖЕНИЯ А Техническое задание

обеспечить работу пользователя без предоставления ему административных привилегий.

4.3 Условия эксплуатации

Программа (клиента) запускается на компьютере или телефоне оператора, и телефоне, доступным для водителей. Серверная часть системы размещается на отдельном выделенном сервере. Должна существовать устойчивая связь по сети между клиентом и сервером.

4.3.1 Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны предоставляться заданные характеристики, обязаны удовлетворять требованиям, предъявляемым к техническим устройствам, на которых запущена программа.

4.3.2 Требования к видам обслуживания

Программа не требует проведения каких-либо видов обслуживания.

4.3.3 Требования к численности и квалификации персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 3 штатных единиц – администратор, оператор и водитель.

При установке и настройке системы необходим системный администратор. В процессе эксплуатации с программой работают оператор и водитель.

Системный администратор должен иметь высшее профильное образование и сертификаты компании-производителя операционной системы. В перечень задач, выполняемых системным администратором, должны входить:

- поддержание работоспособности технических средств;
- установка клиентских приложений;
- настройка сервера;
- настройка СУБД.

Продолжение ПРИЛОЖЕНИЯ А

Продолжение ПРИЛОЖЕНИЯ А Техническое задание

Пользователь программы (оператор и водитель) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

4.4 Требования к составу и параметрам технических средств

Состав технических средств:

- Компьютер оператора, включающий в себя:
 - процессор с тактовой частотой, не менее 2 ГГц;
 - оперативную память объемом, не менее 4 Гб;
 - видеокарту, монитор, мышь, клавиатура.
- Телефон водителя (Android или iOS), включающий в себя:
 - версия Android 7.0 или выше;
 - версия iOS 8.0 или выше.
- Два выделенных сервера (основной и резервный), включающий в себя:
 - процессор с тактовой частотой, не менее 2 ГГц;
 - оперативную память объемом, не менее 8 Гб;
 - видеокарту, монитор, мышь.

4.5 Требования к информационной и программной совместимости

Пользовательский интерфейс должен быть интуитивно понятным и содержать подсказки. Отчеты должны содержать лишь интересующую информацию.

Клиентское приложение обменивается с сервером сообщениями по сети, при этом используется протокол HTTP. Должно быть исключено появление посторонних устройств в сети.

4.6 Требование к маркировке и упаковке

Программное изделие передается по сети Internet в виде архива – загружается с официального сайта производителя. Специальных требований к маркировке

Продолжение ПРИЛОЖЕНИЯ А Техническое задание

не предъявляется. Для проверки подлинности программного обеспечения рекомендуется проверять контрольные суммы загруженных файлов со значениями, указанными на официальном сайте.

4.7 Требования к транспортированию и хранению

Специальных требований не предъявляется.

4.8 Специальные требования

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса, разработанного согласно рекомендациям компании-производителя операционной системы.

5 Требования к программной документации

Предварительный состав программной документации:

- техническое задание (включает описание применения);
- программа и методика испытаний;
- руководство системного программиста;
- руководство оператора;
- руководство программиста;
- ведомость эксплуатационных документов;
- формуляр.

6 Технико-экономические показатели

Эффективность системы определяется удобством использования системы, а также экономической выгодой, полученной от внедрения аппаратно-программного комплекса.

7 Стадии и этапы разработки

Этапы создания приложения, которые необходимо выполнить:

1 этап – разработка технического задания, определение требований к приложению, стадий, этапов и сроков разработки программы, согласование и утверждение технического задания;

Продолжение ПРИЛОЖЕНИЯ А Техническое задание

- 2 этап – анализ процессов деятельности организации;
- 3 этап – анализ предметной области и средств разработки;
- 4 этап – разработка программного продукта;
- 5 этап – тестирование программного продукта;
- 6 этап – доработка программного продукта;
- 7 этап – согласование созданного приложения с требованиями заказчика;
- 8 этап – внедрение и сопровождение.

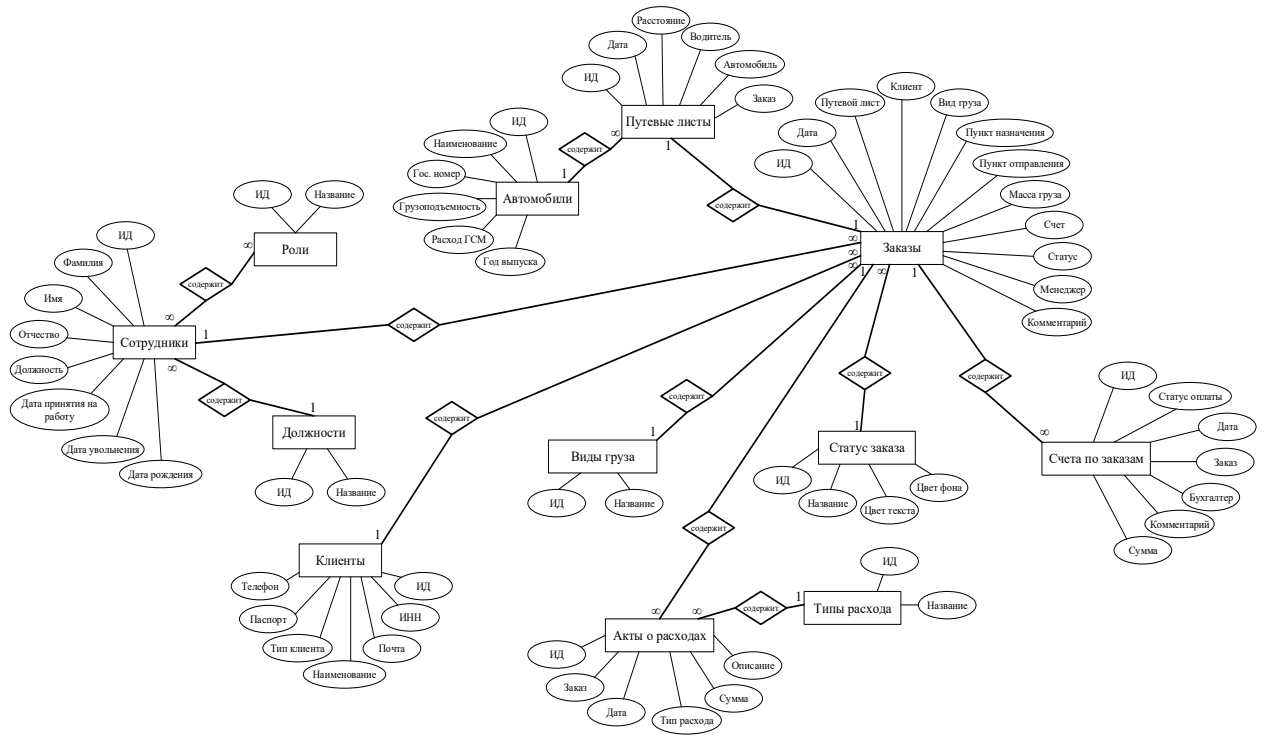
8 Порядок контроля и приемки

Приемосдаточные испытания программы должны проводиться согласно разработанной исполнителем и согласованной заказчиком «Программы и методики испытаний».

Ход проведения приемосдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний. На основании протокола испытаний исполнитель совместно с заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.

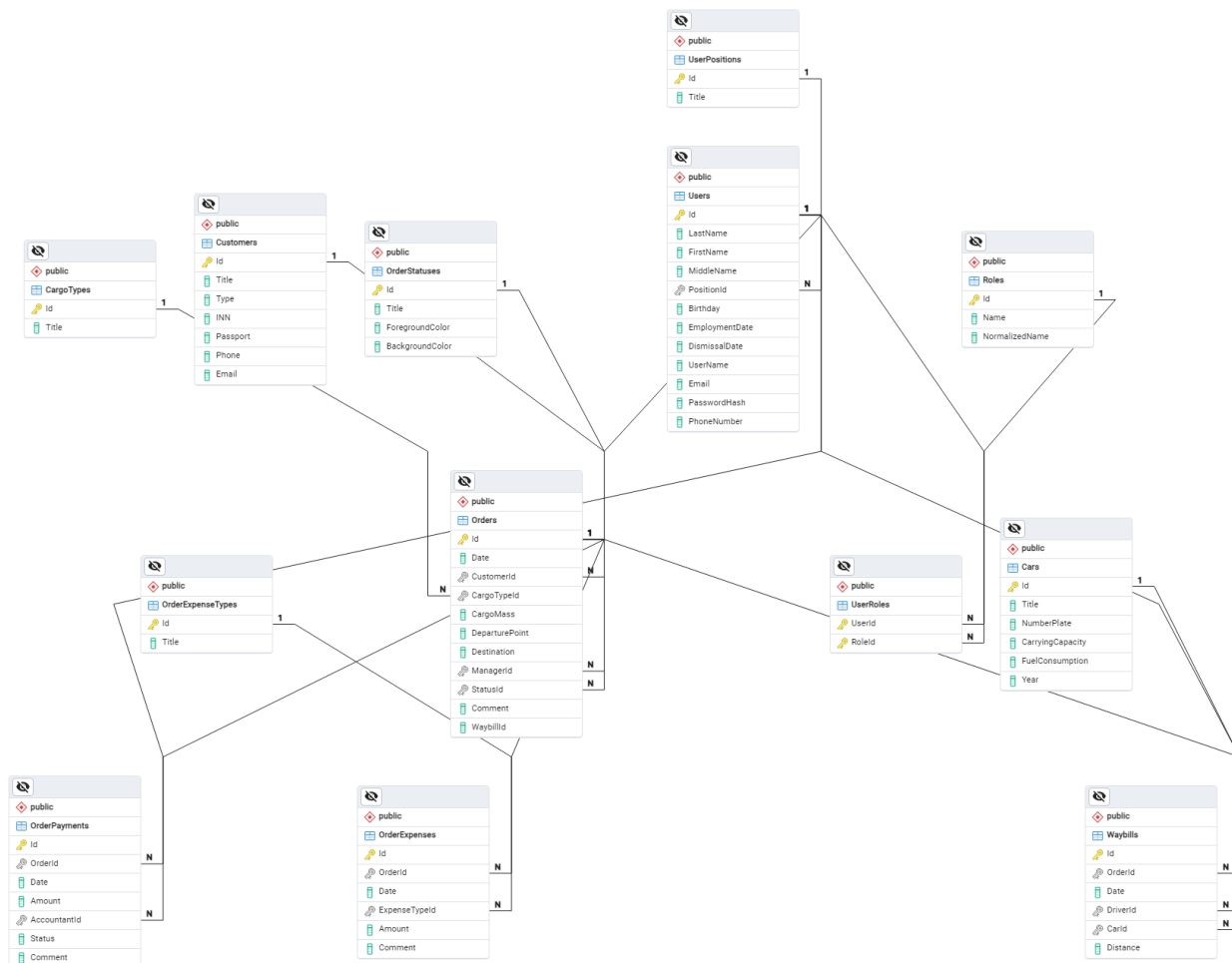
ПРИЛОЖЕНИЕ Б

ER-диаграмма модели в нотации Чена



ПРИЛОЖЕНИЕ В

Логическая схема данных



ПРИЛОЖЕНИЕ Г

Описание методов взаимодействия

Модель ошибки:

```
{
  "status": 400,
  "message": "Текст ошибки",
  "errors": null,
  "traceId": "2816f88c-e2e9-4f46-9247-d70d55b44854"
}
```

Параметр	Тип	Описание
status	Целое число	Статус ответа (>=400)
message	Строка	Текст ошибки
errors	Массив строк	Список ошибок с подробным описанием (при наличии)
traceId	Строка	Уникальный идентификатор взаимодействия

1. *Аутентификация пользователя (получение токена)*

Запрос

POST /api/OAuth/authenticate

```
{
  "userName": "admin@google.com",
  "password": "Pass@word1"
}
```

Параметр	Тип	Описание	Обязательный
userName	Строка	Логин/Email пользователя	Да
password	Строка	Пароль пользователя	Да

Ответ

200 (OK) – Успешное выполнение.

```
{
  "token": "eyJhbGciOiJIUzI1NiIs
  "expiresToken": "2024-02-22T06:50:18.1610979Z",
  "refreshToken": "CfDJ8Oi2cfZdI7NEl7TvopyvsR6qz"
}
```

Параметр	Тип	Описание
token	Строка	Токен взаимодействия, необходим для авторизации в системе
expiresToken	Дата/время	Время жизни токена взаимодействия
refreshToken	Строка	Токен обновления, необходим для генерации нового токена

400 (BadRequest) – Ошибка запроса.

500 (InternalServerError) – Ошибка сервера.

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

2. Обновление токена взаимодействия

Запрос

POST /api/OAuth/refresh-token

```
{  
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9",  
  "refreshToken": "CfDJ8Oi2cfZdI7NEl7TvopyvsR5EvFI8"  
}
```

Параметр	Тип	Описание	Обязательный
token	Строка	Токен взаимодействия	Да
refreshToken	Строка	Токен обновления	Да

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "token": "eyJhbGciOiJIUzI1NiIs",  
  "expiresToken": "2024-02-22T06:50:18.1610979Z",  
  "refreshToken": "CfDJ8Oi2cfZdI7NEl7TvopyvsR6qz"  
}
```

Параметр	Тип	Описание
token	Строка	Токен взаимодействия, необходим для авторизации в системе
expiresToken	Дата/время	Время жизни токена взаимодействия
refreshToken	Строка	Токен обновления, необходим для генерации нового токена

400 (BadRequest) – Ошибка запроса.

500 (InternalServerError) – Ошибка сервера.

3. Получение личной информации

Запрос

POST /api/OAuth/my-info

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 1,  
  "userName": "admin",  
  "normalizedUserName": "ADMIN",  
  "firstName": "Максим",  
  "lastName": "Колосов",  
}
```

Продолжение ПРИЛОЖЕНИЯ Г

Описание методов взаимодействия

```

"middleName": "Алексеевич",
"shortName": "Максим К.",
"fullName": "Колосов Максим Алексеевич",
"email": "admin@google.com",
"normalizedEmail": "ADMIN@GOOGLE.COM",
"phoneNumber": null,
"position": "Системный администратор",
"birthday": null,
"employmentDate": null,
"dismissalDate": null,
"roles": [
  "Администратор"
],
"permissions": []
}

```

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
userName	Строка	Логин
normalizedUserName	Строка	Логин в верхнем регистре
firstName	Строка	Имя
lastName	Строка	Фамилия
middleName	Строка	Отчество
shortName	Строка	Краткое имя (фамилия имя)
fullName	Строка	Полное имя (ФИО)
email	Строка	Почта
normalizedEmail	Строка	Почта в верхнем регистре
phoneNumber	Строка	Номер телефона
position	Строка	Должность
birthday	Дата	Дата рождения
employmentDate	Дата	Дата принятия на работу
dismissalDate	Дата	Дата увольнения
roles	Массив строк	Список ролей
permissions	Массив строк	Список разрешений

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

500 (InternalServerError) – Ошибка сервера.

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

4. Смена пароля

Запрос

POST /api/account/change-password

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

```
{  
  "oldPassword": "Pass@word1",  
  "newPassword": "NewPass@word1"  
}
```

Параметр	Тип	Описание	Обязательный
oldPassword	Строка	Текущий пароль	Да
newPassword	Строка	Новый пароль	Да

Ответ

204 (NoContent) – Успешное выполнение.

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

500 (InternalServerError) – Ошибка сервера.

5. Сотрудники

5.1. Получить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/users/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

ETag: 64976d00-32e7-4522-a130-f6d60463ca18

```
{  
  "id": 1,  
  "userName": "admin",  
  "normalizedUserName": "ADMIN",  
  "email": "admin@google.com",  
  "normalizedEmail": "ADMIN@GOOGLE.COM",  
  "emailConfirmed": true,  
  "phoneNumber": null,  
  "lastName": "Колосов",  
}
```


Продолжение ПРИЛОЖЕНИЯ Г

Описание методов взаимодействия

```

"firstName": "Максим",
"middleName": "Алексеевич",
"positionId": 1,
"position": {
  "id": 1,
  "title": "Системный администратор"
},
"birthday": null,
"employmentDate": null,
"dismissalDate": null,
"created": "2024-03-02T22:37:50.067416",
"modified": "2024-03-03T20:39:17.285271",
"roles": [
  "Администратор"
],
"permissions": []
}

```

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
userName	Строка	Логин
normalizedUserName	Строка	Логин в верхнем регистре
email	Строка	Почта
normalizedEmail	Строка	Почта в верхнем регистре
emailConfirmed	Да/Нет	Почта подтверждена
phoneNumber	Строка	Номер телефона
firstName	Строка	Имя
lastName	Строка	Фамилия
middleName	Строка	Отчество
positionId	Целое число	Идентификатор должности
position.id		
position.title	Строка	Название должности
birthday	Дата	Дата рождения
employmentDate	Дата	Дата принятия на работу
dismissalDate	Дата	Дата увольнения
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения
roles	Массив строк	Список ролей
permissions	Массив строк	Список разрешений

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 500 (InternalServerError) – Ошибка сервера.

5.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

DELETE /api/users/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

- 204 (NoContent) – Успешное выполнение.
- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 500 (InternalServerError) – Ошибка сервера.

5.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/users

```
{  
  "email": "username@mail.ru",  
  "userName": "username",  
  "password": "password",  
  "phoneNumber": "79999999999",  
  "lastName": "Иванов",  
  "firstName": "Иван",  
  "middleName": "Иванович",  
  "positionId": 1,  
  "birthday": "2000-01-24",  
  "employmentDate": "2022-10-23",  
  "dismissalDate": null,  
  "roles": [  
    "Оператор"  
  ],  
}
```

Продолжение ПРИЛОЖЕНИЯ Г

Описание методов взаимодействия

```
"permissions": [
  "user:create"
]
}
```

Параметр	Тип	Описание
email	Строка	Почта
userName	Строка	Логин
password	Строка	Пароль
phoneNumber	Строка	Номер телефона
firstName	Строка	Имя
lastName	Строка	Фамилия
middleName	Строка	Отчество
positionId	Целое число	Идентификатор должности
birthday	Дата	Дата рождения
employmentDate	Дата	Дата принятия на работу
dismissalDate	Дата	Дата увольнения
roles	Массив строк	Список ролей
permissions	Массив строк	Список разрешений

Ответ

200 (OK) – Успешное выполнение.

```
{
  "id": 2,
  "userName": "username",
  "normalizedUserName": "USERNAME",
  "email": "username@mail.ru",
  "normalizedEmail": "USERNAME@MAIL.RU",
  "emailConfirmed": true,
  "phoneNumber": 79999999999,
  "lastName": "Иванов",
  "firstName": "Иван",
  "middleName": "Иванович",
  "positionId": 1,
  "position": {
    "id": 1,
    "title": "Системный администратор"
  },
  "birthday": "2000-01-24",
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"employmentDate": "2022-10-23",
  "dismissalDate": null,
  "created": "2024-03-02T22:37:50.067416",
  "modified": "2024-03-03T20:39:17.285271",
  "roles": [
    "Оператор"
  ],
  "permissions": [
    "user:create"
  ]
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

5.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/users

```
{
  "id": 2,
  "phoneNumber": "79999999999",
  "lastName": "Иванов",
  "firstName": "Иван",
  "middleName": "Иванович",
  "positionId": 2,
  "birthday": "2000-01-24",
  "employmentDate": "2022-10-23",
  "dismissalDate": null,
  "roles": [
    "Оператор"
  ],
  "permissions": [
    "user:create"
  ]
}
```

Продолжение ПРИЛОЖЕНИЯ Г
Описание методов взаимодействия

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
phoneNumber	Строка	Номер телефона	Нет
firstName	Строка	Имя	Да
lastName	Строка	Фамилия	Да
middleName	Строка	Отчество	Нет
positionId	Целое число	Идентификатор должности	Нет
birthday	Дата	Дата рождения	Нет
employmentDate	Дата	Дата принятия на работу	Нет
dismissalDate	Дата	Дата увольнения	Нет
roles	Массив строк	Список ролей	Нет
permissions	Массив строк	Список разрешений	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
  "id": 2,
  "userName": "username",
  "normalizedUserName": "USERNAME",
  "email": "username@mail.ru",
  "normalizedEmail": "USERNAME@MAIL.RU",
  "emailConfirmed": true,
  "phoneNumber": 7999999999,
  "lastName": "Иванов",
  "firstName": "Иван",
  "middleName": "Иванович",
  "positionId": 2,
  "position": {
    "id": 2,
    "title": "Старший оператор"
  },
  "birthday": "2000-01-24",
  "employmentDate": "2022-10-23",
  "dismissalDate": null,
  "created": "2024-03-02T22:37:50.067416",
  "modified": "2024-03-03T20:39:17.285271",
  "roles": [
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"Оператор"  
],  
"permissions": [  
  "user:create"  
]  
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

5.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/users?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "items": [  
    {  
      "id": 1,  
      "userName": "admin",  
      "normalizedUserName": "ADMIN",  
      "email": "admin@google.com",  
      "normalizedEmail": "ADMIN@GOOGLE.COM",  
      "emailConfirmed": true,  
      "phoneNumber": null,  
      "lastName": "Колосов",  
      "firstName": "Максим",  
      "middleName": "Алексеевич",  
      "positionId": 1,  
      "position": {
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```

    "id": 1,
      "title": "Системный администратор"
    },
    "birthday": null,
    "employmentDate": null,
    "dismissalDate": null,
    "created": "2024-03-02T22:37:50.067416",
    "modified": "2024-03-03T19:30:16.852886",
    "roles": [
      "Администратор"
    ],
    "permissions": []
  }
],
"totalCount": 1,
"nextLink": null
}

```

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

5.6. Установить пароль

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/users/set-password

```

{
  "id": 2,
  "password": "password"
}

```

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
password	Строка	Новый пароль	Да

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

Ответ

204 (NoContent) – Успешное выполнение.

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

6. Роли

6.1. Получить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/roles/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

ETag: 64976d00-32e7-4522-a130-f6d60463ca18

```
{  
  "id": 1,  
  "name": "Администратор",  
  "normalizedName": "АДМИНИСТРАТОР",  
  "created": "2024-03-02T22:37:50.067416",  
  "modified": "2024-03-03T20:39:17.285271",  
  "permissions": []  
}
```

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
name	Строка	Название
normalizedName	Строка	Название в верхнем регистре
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения
permissions	Массив строк	Список разрешений

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

6.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

DELETE /api/roles/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

204 (NoContent) – Успешное выполнение.

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

6.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/roles

```
{  
  "name": "Оператор",  
  "permissions": [  
    "role:create"  
  ]  
}
```

Параметр	Тип	Описание	Обязательный
name	Строка	Название	Да
permissions	Массив строк	Список разрешений	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 2,  
  "name": "Оператор",  
  "normalizedName": "ОПЕРАТОР",  
  "created": "2024-03-02T22:37:50.067416",  
}
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"modified": "2024-03-03T20:39:17.285271",  
  "permissions": [  
    "role:create"  
  ]  
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

6.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/roles

```
{  
  "id": 2,  
  "name": "Оператор",  
  "permissions": [  
    "role:create",  
    "role:update"  
  ]  
}
```

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
name	Строка	Название	Да
permissions	Массив строк	Список разрешений	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 2,  
  "name": "Оператор",  
  "normalizedName": "ОПЕРАТОР",  
  "created": "2024-03-02T22:37:50.067416",  
  "modified": "2024-03-03T20:39:17.285271",  
  "permissions": [  
    "role:create",
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"role:update"
]
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

6.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/roles?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
  "items": [
    {
      "id": 1,
      "name": "Администратор",
      "normalizedName": "АДМИНИСТРАТОР",
      "created": "2024-03-02T22:37:50.067416",
      "modified": "2024-03-03T19:30:16.852886",
      "permissions": []
    }
  ],
  "totalCount": 1,
  "nextLink": null
}
```

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 500 (InternalServerError) – Ошибка сервера.

7. Должности

7.1. Получить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/user-positions/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

ETag: 64976d00-32e7-4522-a130-f6d60463ca18

```
{  
  "id": 1,  
  "title": "Системный администратор",  
  "created": "2024-03-02T22:37:50.067416",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
title	Строка	Название
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения

- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 500 (InternalServerError) – Ошибка сервера.

7.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

DELETE /api/user-positions/{id}

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

204 (NoContent) – Успешное выполнение.

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

7.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/user-positions

```
{  
  "title": "Старший операционист"  
}
```

Параметр	Тип	Описание	Обязательный
title	Строка	Название	Да

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 2,  
  "title": "Старший операционист",  
  "created": "2024-03-02T22:37:50.067416",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

7.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/user-positions

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
{
  "id": 2,
  "name": "Ведущий операционист"
}
```

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
title	Строка	Название	Да

Ответ

200 (OK) – Успешное выполнение.

```
{
  "id": 2,
  "title": "Ведущий операционист",
  "created": "2024-03-02T22:37:50.067416",
  "modified": "2024-03-03T20:39:17.285271"
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

7.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/user-positions?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
  "items": [
    {
      "id": 1,
      "title": "Системный администратор",
    }
  ]
}
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```

    "created": "2024-03-02T22:37:50.067416",
    "modified": "2024-03-03T19:30:16.852886"
  }
],
"totalCount": 1,
"nextLink": null
}

```

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 500 (InternalServerError) – Ошибка сервера.

8. Виды груза

8.1. Получить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/order-cargo-types/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

ETag: 64976d00-32e7-4522-a130-f6d60463ca18

```

{
  "id": 1,
  "title": "Опасный груз",
  "created": "2024-03-02T22:37:50.067416",
  "modified": "2024-03-03T20:39:17.285271"
}

```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
title	Строка	Название
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

8.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

DELETE /api/order-cargo-types/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

204 (NoContent) – Успешное выполнение.

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

8.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/order-cargo-types

```
{
  "title": "Негабаритный груз"
}
```

Параметр	Тип	Описание	Обязательный
title	Строка	Название	Да

Ответ

200 (OK) – Успешное выполнение.

```
{
```


Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"id": 2,  
"title": "Негабаритный груз",  
"created": "2024-03-02T22:37:50.067416",  
"modified": "2024-03-03T20:39:17.285271"  
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

8.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/order-cargo-types

```
{  
  "id": 2,  
  "name": "Габаритный груз"  
}
```

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
title	Строка	Название	Да

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 2,  
  "title": "Габаритный груз",  
  "created": "2024-03-02T22:37:50.067416",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

8.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/order-cargo-types?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
  "items": [
    {
      "id": 1,
      "title": "Опасный груз",
      "created": "2024-03-02T22:37:50.067416",
      "modified": "2024-03-03T19:30:16.852886"
    }
  ],
  "totalCount": 1,
  "nextLink": null
}
```

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

9. Типы расхода

9.1. Получить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/order-expense-types/{id}

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

ETag: 64976d00-32e7-4522-a130-f6d60463ca18

```
{
  "id": 1,
  "title": "Заправка автомобиля",
  "created": "2024-03-02T22:37:50.067416",
  "modified": "2024-03-03T20:39:17.285271"
}
```

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
title	Строка	Название
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

9.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

DELETE /api/order-expense-types/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

204 (NoContent) – Успешное выполнение.

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

9.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/order-expense-types

```
{
  "title": "Прочее"
}
```

Параметр	Тип	Описание	Обязательный
title	Строка	Название	Да

Ответ

200 (OK) – Успешное выполнение.

```
{
  "id": 2,
  "title": "Прочее",
  "created": "2024-03-02T22:37:50.067416",
  "modified": "2024-03-03T20:39:17.285271"
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

9.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/order-expense-types

```
{
  "id": 2,
  "name": " Прочие расходы"
}
```

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
title	Строка	Название	Да

Ответ

200 (OK) – Успешное выполнение.

Продолжение ПРИЛОЖЕНИЯ Г

Описание методов взаимодействия

```
{
  "id": 2,
  "title": "Прочие расходы",
  "created": "2024-03-02T22:37:50.067416",
  "modified": "2024-03-03T20:39:17.285271"
}
```

- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 404 (NotFound) – Не найдено
- 409 (Conflict) – Конфликт сохранения.
- 500 (InternalServerError) – Ошибка сервера.

9.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/order-expense-types?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
  "items": [
    {
      "id": 1,
      "title": "Заправка автомобиля",
      "created": "2024-03-02T22:37:50.067416",
      "modified": "2024-03-03T19:30:16.852886"
    }
  ],
  "totalCount": 1,
  "nextLink": null
}
```

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 500 (InternalServerError) – Ошибка сервера.

10. Статусы заказа

10.1. *Получить запись*

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/order-statuses/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

ETag: 64976d00-32e7-4522-a130-f6d60463ca18

```
{  
  "id": 1,  
  "title": "Новый",  
  "foregroundColor": null,  
  "backgroundColor": null,  
  "created": "2024-03-02T22:37:50.067416",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
title	Строка	Название
foregroundColor	Строка	Цвет текста
backgroundColor	Строка	Фон текста
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения

- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 500 (InternalServerError) – Ошибка сервера.

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

10.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

DELETE /api/order-statuses/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

204 (NoContent) – Успешное выполнение.

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

10.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/order-statuses

```
{  
  "title": "Выполнен",  
  "foregroundColor": "White",  
  "backgroundColor": "Green"  
}
```

Параметр	Тип	Описание	Обязательный
title	Строка	Название	Да
foregroundColor	Строка	Цвет текста	Нет
backgroundColor	Строка	Фон текста	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 2,  
  "title": "Выполнен",  
  "foregroundColor": "White",  
  "backgroundColor": "Green",  
  "created": "2024-03-02T22:37:50.067416",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

10.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/order-statuses

```
{
  "id": 2,
  "name": "Выполнен",
  "backgroundColor": "DarkGreen"
}
```

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
title	Строка	Название	Да
foregroundColor	Строка	Цвет текста	Нет
backgroundColor	Строка	Фон текста	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
  "id": 2,
  "title": "Выполнен",
  "foregroundColor": "White",
  "backgroundColor": "Green",
  "created": "2024-03-02T22:37:50.067416",
  "modified": "2024-03-03T20:39:17.285271"
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

10.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/order-statuses?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
  "items": [
    {
      "id": 1,
      "title": "Новый",
      "foregroundColor": null,
      "backgroundColor": null,
      "created": "2024-03-02T22:37:50.067416",
      "modified": "2024-03-03T19:30:16.852886"
    }
  ],
  "totalCount": 1,
  "nextLink": null
}
```

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

11. Клиенты

11.1. Получить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

GET /api/customers/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

Etag: 64976d00-32e7-4522-a130-f6d60463ca18

```
{  
  "id": 1,  
  "title": "Иванов Иван Иванович",  
  "type": "Individual",  
  "inn": null,  
  "passport": "1010 111111",  
  "phone": "+79999999999",  
  "email": "example@mail.ru",  
  "created": "2024-03-01T14:43:46.597469",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
title	Строка	Название
type	Строка	Тип клиента
inn	Строка	ИНН
passport	Строка	Паспорт (серия и номер)
phone	Строка	Номер телефона
email	Строка	Почта
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

11.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

DELETE /api/customers/{id}

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

204 (NoContent) – Успешное выполнение.

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

11.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/customers

```
{
  "title": "ОАО Клиент",
  "type": "LegalEntity",
  "inn": "12345678901",
  "phone": "+79999999999",
  "email": "example@mail.ru",
  "created": "2024-03-01T14:43:46.597469",
  "modified": "2024-03-03T20:39:17.285271"
}
```

Параметр	Тип	Описание	Обязательный
title	Строка	Название	Да
type	Строка	Тип клиента (LegalEntity или Individual)	Да
inn	Строка	ИНН	Да (если type = LegalEntity)
passport	Строка	Паспорт (серия и номер)	Да (если type = Individual)
phone	Строка	Номер телефона	Да
email	Строка	Почта	Да

Ответ

200 (OK) – Успешное выполнение.

```
{
  "id": 2,
  "title": "ОАО Клиент",
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"type": "LegalEntity",  
  "inn": "12345678901",  
  "passport": null,  
  "phone": "+79999999999",  
  "email": "example@mail.ru",  
  "created": "2024-03-01T14:43:46.597469",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

11.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/customers

```
{  
  "id": 2,  
  "title": "АО Клиент",  
  "type": "LegalEntity",  
  "inn": "12345678901",  
  "passport": null,  
  "phone": "+79999999999",  
  "email": "example@mail.ru"  
}
```

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
title	Строка	Название	Да
foregroundColor	Строка	Цвет текста	Нет
backgroundColor	Строка	Фон текста	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 2,  
  "title": "АО Клиент",
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"type": "LegalEntity",  
  "inn": "12345678901",  
  "passport": null,  
  "phone": "+79999999999",  
  "email": "example@mail.ru",  
  "created": "2024-03-02T22:37:50.067416",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

11.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/customers?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "items": [  
    {  
      "id": 1,  
      "title": "Иванов Иван Иванович",  
      "type": "Individual",  
      "inn": null,  
      "passport": "1010 111111",  
      "phone": "+79999999999",  
      "email": "example@mail.ru",  
      "created": "2024-03-01T14:43:46.597469",  
      "modified": "2024-03-01T14:43:46.597666"  
    }  
  ]  
}
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
],  
  "totalCount": 1,  
  "nextLink": null  
}
```

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

12. Автомобили

12.1. *Получить запись*

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/cars/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

Etag: 64976d00-32e7-4522-a130-f6d60463ca18

```
{  
  "id": 1,  
  "title": "Mercedes-Benz Arocs",  
  "year": 2022,  
  "numberPlate": "O345AO 28PUC",  
  "fuelConsumption": 10,  
  "carryingCapacity": 41  
  "created": "2024-03-01T14:43:46.597469",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
title	Строка	Название
year	Целое число	Год выпуска
numberPlate	Строка	Государственный номер
fuelConsumption	Дробное число	Расход ГСМ
carryingCapacity	Дробное число	Грузоподъемность
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

12.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

DELETE /api/cars/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

204 (NoContent) – Успешное выполнение.

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

12.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/cars

```
{
  "title": "Mercedes-Benz Arocs",
  "year": 2022,
  "numberPlate": "O345AO 28PUC",
  "fuelConsumption": 10,
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"carryingCapacity": 41
}
```

Параметр	Тип	Описание	Обязательный
title	Строка	Название	Да
year	Целое число	Год выпуска	Да
numberPlate	Строка	Государственный номер	Да
fuelConsumption	Дробное число	Расход ГСМ	Да
carryingCapacity	Дробное число	Грузоподъемность	Да

Ответ

200 (OK) – Успешное выполнение.

```
{
  "id": 2,
  "title": "Mercedes-Benz Arocs",
  "numberPlate": "O345AO 28PUC",
  "carryingCapacity": 41,
  "fuelConsumption": 10,
  "year": 2022,
  "created": "2024-03-01T14:43:46.597469",
  "modified": "2024-03-03T20:39:17.285271"
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

12.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/cars

```
{
  "id": 1,
  "title": "Mercedes-Benz Arocs",
  "numberPlate": "O345AO 125PUC",
  "carryingCapacity": 40,
  "fuelConsumption": 10,
  "year": 2022
}
```


Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

}

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
title	Строка	Название	Да
foregroundColor	Строка	Цвет текста	Нет
backgroundColor	Строка	Фон текста	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
  "id": 1,
  "title": "Mercedes-Benz Arocs",
  "numberPlate": "O345AO 125PUC",
  "carryingCapacity": 41,
  "fuelConsumption": 10,
  "year": 2022,
  "created": "2024-03-02T22:37:50.067416",
  "modified": "2024-03-03T20:39:17.285271"
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

12.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/cars?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
  "items": [
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```

{
  "id": 1,
  "title": "Mercedes-Benz Arocs",
  "numberPlate": "O345AO 28PUC",
  "carryingCapacity": 41,
  "fuelConsumption": 10,
  "year": 2022,
  "created": "2024-03-01T14:43:46.597469",
  "modified": "2024-03-01T14:43:46.597666"
}
],
"totalCount": 1,
"nextLink": null
}

```

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

13. Акты о расходах

13.1. *Получить запись*

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/order-expenses/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

ETag: 64976d00-32e7-4522-a130-f6d60463ca18

```

{
  "id": 1,
  "orderId": 1,

```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"date": "2024-03-01T10:34:23.345634",
"expenseTypeId": 1,
"expenseType": {
  "id": 1,
  "title": "Заправка автомобиля"
},
"amount": 3500,
"comment": "",
"created": "2024-03-01T14:43:46.597469",
"modified": "2024-03-03T20:39:17.285271"
}
```

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
orderId	Целое число	Идентификатор заказа
date	Дата/Время	Дата совершения акта
expenseTypeId	Целое число	Идентификатор типа расхода
expenseType.Id		
expenseType.Title	Строка	Название типа расхода
amount	Дробное число	Сумма
comment	Текст	Комментарий
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения

- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 500 (InternalServerError) – Ошибка сервера.

13.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
DELETE /api/order-expenses/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

- 204 (NoContent) – Успешное выполнение.
- 400 (BadRequest) – Ошибка запроса.

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

13.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/order-expenses

```
{  
  "date": "2024-02-23T02:45:00Z",  
  "amount": 23456.34,  
  "orderId": 3,  
  "expenseTypeId": 1,  
  "comment": ""  
}
```

Параметр	Тип	Описание	Обязательный
date	Дата/Время	Дата совершения акта	Да
amount	Дробное число	Сумма	Да
orderId	Целое число	Идентификатор заказа	Да
expenseTypeId	Целое число	Идентификатор типа расхода	Да
comment	Текст	Комментарий	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 1,  
  "date": "2024-02-23T02:45:00Z",  
  "amount": 23456.34,  
  "orderId": 3,  
  "expenseTypeId": 1,  
  "expenseType": {  
    "id": 1,  
    "title": "Заправка автомобиля"  
  },  
  "comment": ""  
  "created": "2024-03-01T14:43:46.597469",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

13.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/order-expenses

```
{  
  "id": 1,  
  "date": "2024-02-23T02:45:00Z",  
  "amount": 23456.34,  
  "orderId": 3,  
  "expenseTypeId": 1,  
  "comment": "Комментарий"  
}
```

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
title	Строка	Название	Да
foregroundColor	Строка	Цвет текста	Нет
backgroundColor	Строка	Фон текста	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 1,  
  "date": "2024-02-23T02:45:00Z",  
  "amount": 23456.34,  
  "orderId": 3,  
  "expenseTypeId": 1,  
  "expenseType": {  
    "id": 1,  
    "title": "Заправка автомобиля"  
  },  
  "comment": "Комментарий"  
  "created": "2024-03-01T14:43:46.597469",
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"modified": "2024-03-03T20:39:17.285271"
```

```
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

13.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/order-expenses?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
```

```
  "items": [
```

```
    {
```

```
      "id": 1,
```

```
      "orderId": 1,
```

```
      "date": "2024-03-01T10:34:23.345634",
```

```
      "expenseTypeId": 1,
```

```
      "expenseType": {
```

```
        "id": 1,
```

```
        "title": "Заправка автомобиля"
```

```
      },
```

```
      "amount": 3500,
```

```
      "comment": "",
```

```
      "created": "2024-03-01T14:43:46.597469",
```

```
      "modified": "2024-03-03T20:39:17.285271"
```

```
    }
```

```
  ],
```

```
  "totalCount": 1,
```

```
  "nextLink": null
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

}

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

14. Счета по заказам

14.1. *Получить запись*

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/order-payments/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

Etag: 64976d00-32e7-4522-a130-f6d60463ca18

```
{
  "id": 1,
  "orderId": 1,
  "date": "2024-03-01T10:34:23.345634",
  "amount": 3500,
  "accountantId": 1,
  "accountant": {
    "id": 1,
    "title": "Иванов Иван Иванович"
  },
  "status": "NotPaid",
  "comment": "",
  "created": "2024-03-01T14:43:46.597469",
  "modified": "2024-03-03T20:39:17.285271"
}
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
orderId	Целое число	Идентификатор заказа
date	Дата/Время	Дата формирования счета
amount	Дробное число	Сумма
accountantId	Целое число	Идентификатор сотрудника (бухгалтера)
accountant.Id		
accountant.Title	Строка	ФИО сотрудника
status	Строка	Статус оплаты (NotPaid или Paid)
comment	Текст	Комментарий
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

14.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

DELETE /api/order-payments/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

204 (NoContent) – Успешное выполнение.

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

14.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/order-payments

{

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"orderId": 1,  
"date": "2024-03-01T10:34:23.345634",  
"amount": 15000,  
"accountantId": 1,  
"status": "NotPaid",  
"comment": ""  
}
```

Параметр	Тип	Описание	Обязательный
orderId	Целое число	Идентификатор заказ	Да
date	Дата/Время	Дата совершения акта	Да
amount	Дробное число	Сумма	Да
accountantId	Целое число	Идентификатор сотрудника	Да
status	Строка	Статус оплаты (NotPaid или Paid)	Да
comment	Текст	Комментарий	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 1,  
  "orderId": 1,  
  "date": "2024-03-01T10:34:23.345634",  
  "amount": 3500,  
  "accountantId": 1,  
  "accountant": {  
    "id": 1,  
    "title": "Иванов Иван Иванович"  
  },  
  "status": "NotPaid",  
  "comment": "",  
  "created": "2024-03-01T14:43:46.597469",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

14.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/order-payments

```
{
  "id": 1,
  "orderId": 1,
  "date": "2024-03-01T10:34:23.345634",
  "amount": 15000,
  "accountantId": 1,
  "status": "Paid",
  "comment": "Оплачено"
}
```

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
orderId	Целое число	Идентификатор заказа	Да
date	Дата/Время	Дата совершения акта	Да
amount	Дробное число	Сумма	Да
accountantId	Целое число	Идентификатор сотрудника	Да
status	Строка	Статус оплаты (NotPaid или Paid)	Да
comment	Текст	Комментарий	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
  "id": 1,
  "orderId": 1,
  "date": "2024-03-01T10:34:23.345634",
  "amount": 3500,
  "accountantId": 1,
  "accountant": {
    "id": 1,
    "title": "Иванов Иван Иванович"
  },
  "status": "Paid",
  "comment": "Оплачено",
  "created": "2024-03-01T14:43:46.597469",
}
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"modified": "2024-03-03T20:39:17.285271"
```

```
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

14.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/order-payments?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
```

```
  "items": [
```

```
    {
```

```
      "id": 1,
```

```
      "orderId": 1,
```

```
      "date": "2024-03-01T10:34:23.345634",
```

```
      "amount": 3500,
```

```
      "accountantId": 1,
```

```
      "accountant": {
```

```
        "id": 1,
```

```
        "title": "Иванов Иван Иванович"
```

```
      },
```

```
      "status": "Paid",
```

```
      "comment": "Оплачено",
```

```
      "created": "2024-03-01T14:43:46.597469",
```

```
      "modified": "2024-03-03T20:39:17.285271"
```

```
    }
```

```
  ],
```

```
  "totalCount": 1,
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"nextLink": null
}
```

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 500 (InternalServerError) – Ошибка сервера.

15. Путевые листы

15.1. *Получить запись*

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/waybills/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

ETag: 64976d00-32e7-4522-a130-f6d60463ca18

```
{
  "id": 1,
  "orderId": 1,
  "date": "2024-03-01T10:34:23.345634",
  "driverId": 3,
  "driver": {
    "id": 3,
    "title": "Иванов Иван Иванович"
  },
  "carId": 1,
  "car": {
    "id": 1,
    "title": "Mercedes-Benz Arocs"
  },
  "distance": 5100,
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"created": "2024-03-01T14:43:46.597469",
"modified": "2024-03-03T20:39:17.285271"
}
```

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
orderId	Целое число	Идентификатор заказа
date	Дата/Время	Дата формирования заказа
distance	Дробное число	Расстояние, км
driverId	Целое число	Идентификатор сотрудника (водителя)
driver.Id		
driver.Title	Строка	ФИО сотрудника
carId	Целое число	Идентификатор автомобиля
car.Id		
car.Title	Строка	Название автомобиля
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения

- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 500 (InternalServerError) – Ошибка сервера.

15.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
DELETE /api/waybills/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

- 204 (NoContent) – Успешное выполнение.
- 400 (BadRequest) – Ошибка запроса.
- 401 (Unauthorized) – Не авторизован.
- 403 (Forbidden) – Нет доступа.
- 500 (InternalServerError) – Ошибка сервера.

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

15.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/waybills

```
{  
  "orderId": 1,  
  "date": "2024-03-01T10:34:23.345634",  
  "distance": 5100,  
  "carId": 1,  
  "driverId": 3  
}
```

Параметр	Тип	Описание	Обязательный
orderId	Целое число	Идентификатор заказа	Да
date	Дата/Время	Дата совершения акта	Да
distance	Дробное число	Расстояние, км	Да
carId	Целое число	Идентификатор автомобиля	Да
driverId	Целое число	Идентификатор сотрудника (водителя)	Да

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 1,  
  "orderId": 1,  
  "date": "2024-03-01T10:34:23.345634",  
  "driverId": 3,  
  "driver": {  
    "id": 3,  
    "title": "Иванов Иван Иванович"  
  },  
  "carId": 1,  
  "car": {  
    "id": 1,  
    "title": "Mercedes-Benz Arocs"  
  },  
  "distance": 5100,  
  "created": "2024-03-01T14:43:46.597469",  
  "modified": "2024-03-03T20:39:17.285271"  
}
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

15.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/waybills

```
{  
  "id": 1,  
  "orderId": 1,  
  "date": "2024-03-01T10:34:23.345634",  
  "distance": 5000,  
  "carId": 1,  
  "driverId": 3  
}
```

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
orderId	Целое число	Идентификатор заказ	Да
date	Дата/Время	Дата совершения акта	Да
amount	Дробное число	Сумма	Да
accountantId	Целое число	Идентификатор сотрудника	Да
status	Строка	Статус оплаты (NotPaid или Paid)	Да
comment	Текст	Комментарий	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 1,  
  "orderId": 1,  
  "date": "2024-03-01T10:34:23.345634",  
  "driverId": 3,  
  "driver": {  
    "id": 3,  
    "title": "Иванов Иван Иванович"  
  },  
}
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"carId": 1,  
"car": {  
  "id": 1,  
  "title": "Mercedes-Benz Arocs"  
},  
"distance": 5000,  
"created": "2024-03-01T14:43:46.597469",  
"modified": "2024-03-03T20:39:17.285271"  
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

15.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/waybills?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "items": [  
    {  
      "id": 1,  
      "orderId": 1,  
      "date": "2024-03-01T10:34:23.345634",  
      "driverId": 3,  
      "driver": {  
        "id": 3,  
        "title": "Иванов Иван Иванович"  
      },  
      "carId": 1,  
    },  
  ],  
}
```


Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```

"car": {
  "id": 1,
  "title": "Mercedes-Benz Arocs"
},
"distance": 5000,
"created": "2024-03-01T14:43:46.597469",
"modified": "2024-03-03T20:39:17.285271"
}
],
"totalCount": 1,
"nextLink": null
}

```

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

16. Заказы

16.1. *Получить запись*

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/orders/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

200 (OK) – Успешное выполнение.

ETag: 64976d00-32e7-4522-a130-f6d60463ca18

```

{
  "id": 1,
  "date": "2024-03-01T10:34:23.345634",
  "customerId": 2,
  "customer": {

```

Продолжение ПРИЛОЖЕНИЯ Г
Описание методов взаимодействия

```
"id": 2,  
  "title": "Петров Петр Петрович"  
},  
"cargoTypeId": 1,  
"cargoType": {  
  "id": 1,  
  "title": "Mercedes-Benz Arocs"  
},  
"cargoMass": 200,  
"departurePoint": "Москва, 117209",  
"destination": "Благовещенск, 675000",  
"managerId": 1,  
"manager": {  
  "id": 1,  
  "title": "Иванов Иван Иванович"  
},  
"statusId": 3,  
"status": {  
  "id": 3,  
  "title": "Новый"  
},  
"comment": "",  
"waybillId": 1,  
"created": "2024-03-01T14:43:46.597469",  
"modified": "2024-03-03T20:39:17.285271"  
}
```

Продолжение ПРИЛОЖЕНИЯ Г
Описание методов взаимодействия

Параметр	Тип	Описание
id	Целое число	Уникальный идентификатор
date	Дата/Время	Дата формирования заказа
customerId	Целое число	Идентификатор клиента
customer.Id		
customer.Title	Строка	ФИО клиента
cargoTypeId	Целое число	Идентификатор типа груза
cargoType.Id		
cargoType.Title	Строка	Название типа груза
cargoMass	Дробное число	Вес груза, кг
departurePoint	Строка	Пункт отправления
destination	Строка	Пункт назначения
managerId	Целое число	Идентификатор сотрудника (менеджера)
manager.Id		
manager.Title	Строка	ФИО сотрудника
statusId	Целое число	Идентификатор статуса
status.Id		
status.Title	Строка	Название статуса
comment	Текст	Комментарий
waybillId	Целое число	Идентификатор путевого листа
created	Дата/Время	Дата создания
modified	Дата/Время	Дата изменения

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

16.2. Удалить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

DELETE /api/orders/{id}

Параметр	Тип	Описание	Обязательный
id	Целое число	Идентификатор записи	Да

Ответ

204 (NoContent) – Успешное выполнение.

400 (BadRequest) – Ошибка запроса.

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

16.3. Создать запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

POST /api/orders

```
{  
  "date": "2024-03-01T10:34:23.345634",  
  "customerId": 2,  
  "cargoTypeId": 1,  
  "cargoMass": 200,  
  "departurePoint": "Москва, 117209",  
  "destination": "Благовещенск, 675000",  
  "managerId": 1,  
  "statusId": 3,  
  "waybillId": 1,  
  "comment": ""  
}
```

Параметр	Тип	Описание	Обязательный
date	Дата/Время	Дата формирования заказа	Да
customerId	Целое число	Идентификатор клиента	Да
cargoTypeId	Целое число	Идентификатор типа груза	Да
cargoMass	Дробное число	Вес груза, кг	Да
departurePoint	Строка	Пункт отправления	Да
destination	Строка	Пункт назначения	Да
managerId	Целое число	Идентификатор сотрудника (менеджера)	Да
statusId	Целое число	Идентификатор статуса	Да
comment	Текст	Комментарий	Нет
waybillId	Целое число	Идентификатор путевого листа	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{  
  "id": 1,  
  "date": "2024-03-01T10:34:23.345634",  
  "customerId": 2,
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```
"customer": {
  "id": 2,
  "title": "Петров Петр Петрович"
},
"cargoTypeId": 1,
"cargoType": {
  "id": 1,
  "title": "Mercedes-Benz Arocs"
},
"cargoMass": 200,
"departurePoint": "Москва, 117209",
"destination": "Благовещенск, 675000",
"managerId": 1,
"manager": {
  "id": 1,
  "title": "Иванов Иван Иванович"
},
"statusId": 3,
"status": {
  "id": 3,
  "title": "Новый"
},
"comment": "",
"waybillId": 1,
"created": "2024-03-01T14:43:46.597469",
"modified": "2024-03-03T20:39:17.285271"
}
```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.

16.4. Изменить запись

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

If-Match: 64976d00-32e7-4522-a130-f6d60463ca18

PUT /api/orders

```
{
  "id": 1,
```

Продолжение ПРИЛОЖЕНИЯ Г
Описание методов взаимодействия

```
"date": "2024-03-01T10:34:23.345634",
"customerId": 2,
"cargoTypeId": 1,
"cargoMass": 200,
"departurePoint": "Москва, 117209",
"destination": "Благовещенск, 675000",
"managerId": 1,
"statusId": 2,
"waybillId": 1,
"comment": ""
}
```

Параметр	Тип	Описание	Обязательный
id	Целое число	Уникальный идентификатор	Да
date	Дата/Время	Дата формирования заказа	Да
customerId	Целое число	Идентификатор клиента	Да
cargoTypeId	Целое число	Идентификатор типа груза	Да
cargoMass	Дробное число	Вес груза, кг	Да
departurePoint	Строка	Пункт отправления	Да
destination	Строка	Пункт назначения	Да
managerId	Целое число	Идентификатор сотрудника (менеджера)	Да
statusId	Целое число	Идентификатор статуса	Да
comment	Текст	Комментарий	Нет
waybillId	Целое число	Идентификатор путевого листа	Нет

Ответ

200 (OK) – Успешное выполнение.

```
{
  "id": 1,
  "date": "2024-03-01T10:34:23.345634",
  "customerId": 2,
  "customer": {
    "id": 2,
    "title": "Петров Петр Петрович"
  },
  "cargoTypeId": 1,
  "cargoType": {
    "id": 1,
    "title": "Mercedes-Benz Arocs"
  }
}
```

Продолжение ПРИЛОЖЕНИЯ Г Описание методов взаимодействия

```

},
  "cargoMass": 200,
  "departurePoint": "Москва, 117209",
  "destination": "Благовещенск, 675000",
  "managerId": 1,
  "manager": {
    "id": 1,
    "title": "Иванов Иван Иванович"
  },
  "statusId": 2,
  "status": {
    "id": 2,
    "title": "Доставляется"
  },
  "comment": "",
  "waybillId": 1,
  "created": "2024-03-01T14:43:46.597469",
  "modified": "2024-03-03T20:39:17.285271"
}

```

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

404 (NotFound) – Не найдено

409 (Conflict) – Конфликт сохранения.

500 (InternalServerError) – Ошибка сервера.

16.5. Получить список записей

Запрос

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

GET /api/orders?PageSize={pageSize}&PageNumber={pageNumber}

Параметр	Тип	Описание	Обязательный
pageSize	Целое число	Количество записей в ответе	Нет
pageNumber	Целое число	Номер страницы	Нет

Ответ

200 (OK) – Успешное выполнение.

```

{
  "items": [

```

Продолжение ПРИЛОЖЕНИЯ Г
Описание методов взаимодействия

```
{
  "id": 1,
  "date": "2024-03-01T10:34:23.345634",
  "customerId": 2,
  "customer": {
    "id": 2,
    "title": "Петров Петр Петрович"
  },
  "cargoTypeId": 1,
  "cargoType": {
    "id": 1,
    "title": "Mercedes-Benz Arocs"
  },
  "cargoMass": 200,
  "departurePoint": "Москва, 117209",
  "destination": "Благовещенск, 675000",
  "managerId": 1,
  "manager": {
    "id": 1,
    "title": "Иванов Иван Иванович"
  },
  "statusId": 2,
  "status": {
    "id": 2,
    "title": "Доставляется"
  },
  "comment": "",
  "waybillId": 1,
  "created": "2024-03-01T14:43:46.597469",
  "modified": "2024-03-03T20:39:17.285271"
}
],
"totalCount": 1,
"nextLink": null
}
```


Продолжение ПРИЛОЖЕНИЯ Г
Описание методов взаимодействия

Параметр	Тип	Описание
items	Массив	Массив с информацией полученных записей
totalCount	Целое число	Всего записей
nextLink	Строка	Ссылка на следующую «пачку» элементов

400 (BadRequest) – Ошибка запроса.

401 (Unauthorized) – Не авторизован.

403 (Forbidden) – Нет доступа.

500 (InternalServerError) – Ошибка сервера.